deviceTRUST 21.1

Contents

Welcome

September 6, 2025

The deviceTRUST® documentation provides information about the installation, setup and a product reference for deviceTRUST.

Quick setup

To get started with deviceTRUST, choose your Architecture and then take a look at the Getting Started guide.

More information

- Architecture describes the different deployment scenarios.
- Getting Started provides the essential steps for a successful deviceTRUST installation.
- Download provides links that can be used to download the latest deviceTRUST Software.
- Installation details some important usage scenarios, plus how to install the deviceTRUST Console, Agent and Client Extension.
- Templates details the templates which can be used to quickly implement a use case.
- Reporting contains information about analyzing and reporting contextual information of your users and devices.
- Reference provides a set of reference material describing the more advanced features of deviceTRUST.
- Troubleshooting important steps are described to help troubleshoot the deviceTRUST installation and configuration.
- Releases contains the release notes detailing new features and bug fixes within the released deviceTRUST products.

Architecture

June 20, 2025

TABLE OF CONTENTS

Local Scenario

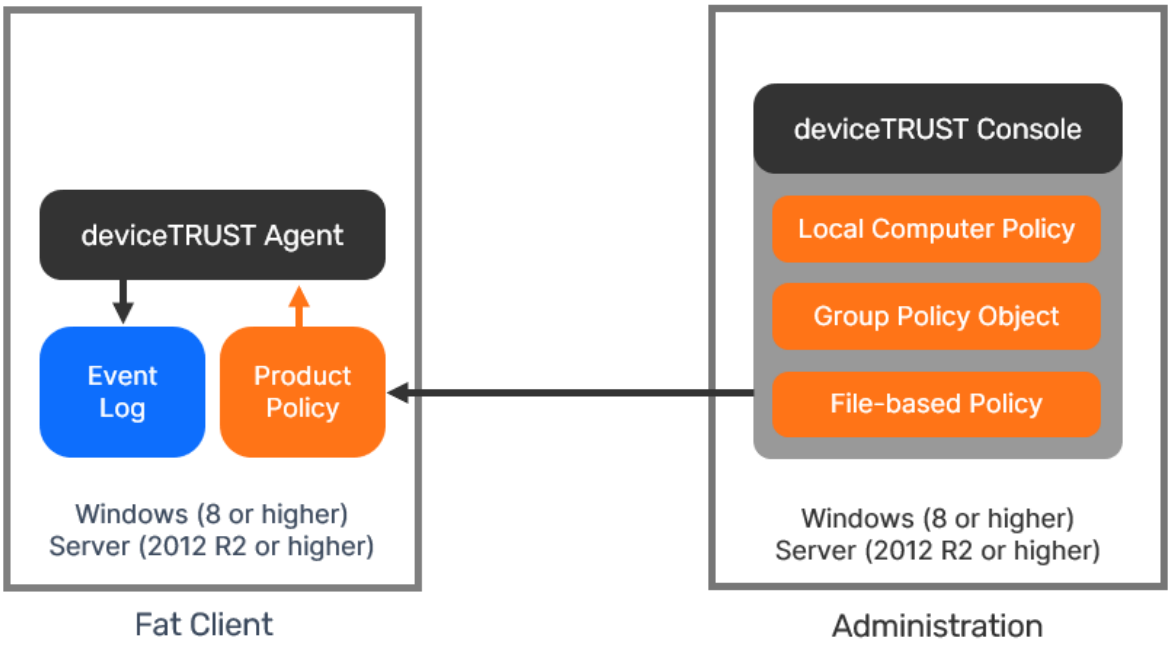
deviceTRUST requires only one main component when installing on local devices, the deviceTRUST Agent. The deviceTRUST component can be installed and configured within minutes and can be fully integrated with existing deployment processes and management tools. No additional infrastructure (e.g. a database or a web server) is required for deviceTRUST to be installed in your environment.

deviceTRUST® Agent

This component needs to be installed on the local device. The Property Reference describes which properties of the local agent are available within the users'session.

Architecture - Microsoft Windows local devices

The following diagram details the deviceTRUST architecture when the agent is installed on a Windows OS, with deviceTRUST making the user and device context information available within the local desktop session. Policy is made available to the deviceTRUST Agent using existing Microsoft Active Directory Group Policy Management or file-based. All operations performed by the deviceTRUST Agent are written to the Microsoft Windows Event Log.



Remote Scenario

deviceTRUST consists of two main components when installing in remote environments, the deviceTRUST Agent and the deviceTRUST Client Extension. Both deviceTRUST components can be installed and configured within minutes and can be fully integrated with existing deployment processes and management tools. No additional infrastructure (e.g. a database or a web server) is required for deviceTRUST to be installed in your environment.

Solutions are provided by deviceTRUST® for both traditional and modern Operating Systems (OS). On a traditional OS such as Microsoft Windows, an extensibility framework is available that enables deviceTRUST to send user and device context within the communication channel between the clients and the Remote Desktop Services host. deviceTRUST also provides a solution for more modern OS's, such as Apple iOS, which offer no extensibility framework.

deviceTRUST Agent

This component needs to be installed on the remoting host that delivers the remote session to the users. The following technologies are supported by deviceTRUST: Amazon WorkSpaces, Citrix Virtual Apps and Desktops™ (CVAD), Microsoft Azure Virtual Desktop (AVD), Microsoft Remote Desktop Session Host (RDSH) or VMware Horizon View.

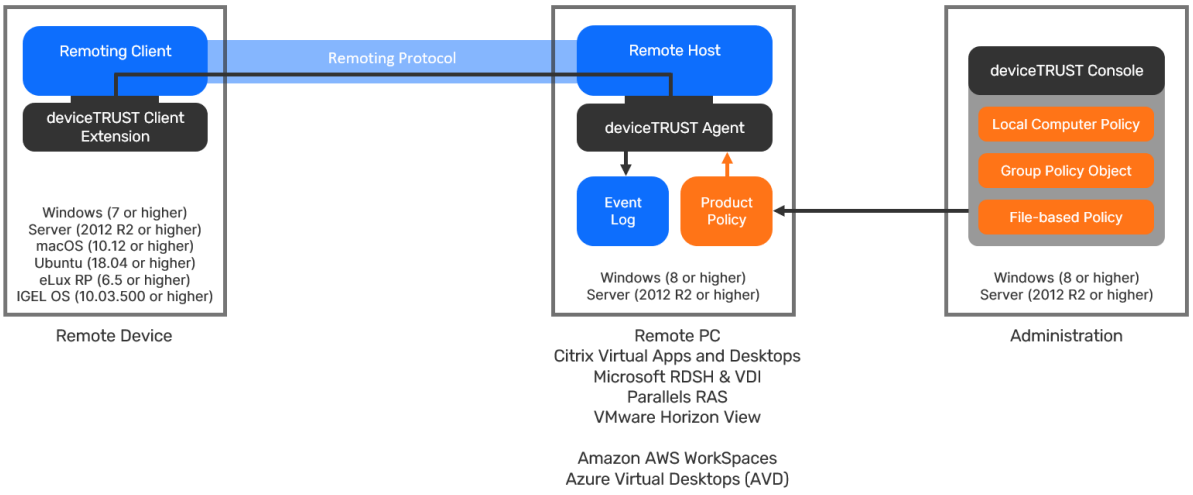
deviceTRUST Client Extension

This component needs to be installed on the remote device which will be used to connect to the remote host delivering the published applications and desktops. It is not required to have deviceTRUST Client Extension installed onto all of your remote devices but recommended to get the full range of context information about the remote device and its user into the users'virtual session.

In the absence of the deviceTRUST Client Extension on the remote device, deviceTRUST delivers the LOCAL_* properties into the users'remote session. The Property Reference describes which properties of the local agent and remote device are available within the users'session.

Architecture - Windows, macOS, Ubuntu, eLux® RP or IGEL OS device

The following diagram details the deviceTRUST architecture when the remote client is installed on a Windows, macOS, Ubuntu, eLux RP or IGEL OS device, with deviceTRUST sending the user and device context information within the communication channel offered by the remoting protocol. Policy is made available to the deviceTRUST Agent using existing Microsoft Active Directory Group Policy Management or file-based. All operations performed by the deviceTRUST Agent are written to the Microsoft Windows Event Log.



Operating System Compatibility

- Apple iOS and iPadOS
- Apple macOS
- IGEL OS
- Microsoft Windows
- Ubuntu Desktop

- Unicon eLux

Apple iOS and iPadOS

Compatible operating systems (deviceTRUST® Client Extension):

- Apple iOS 11.x and later
- Apple iPadOS 11.x and later

Compatible technologies:

- Citrix Virtual Apps and Desktops™
- Citrix Cloud™
- Microsoft Remote Desktop Services
- VMware Horizon View (Blast)
- VMware Horizon View (PCoIP)
- VMware Horizon View (RDP)

Apple macOS

Compatible operating systems (deviceTRUST Client Extension):

- Apple macOS 10.13 and later

Compatible technologies:

- Amazon WorkSpaces (PCoIP)
- Citrix Virtual Apps and Desktops
- Citrix Cloud
- Microsoft Remote Desktop Services (via FreeRDP 2)
- VMware Horizon View (Blast)
- VMware Horizon View (PCoIP)
- VMware Horizon View (RDP)

IGEL OS

Compatible technologies (deviceTRUST Client Extension):

- Amazon WorkSpaces (PCoIP) natively integrated in IGEL OS 11.08.200 and later - 1
- Citrix Virtual Apps and Desktops natively integrated in IGEL OS 10.03.500 and later
- Citrix Cloud natively integrated in IGEL OS 10.03.500 and later
- Microsoft Azure Virtual Desktop (AVD) natively integrated in IGEL OS 11.08.200 and later - 1

- Microsoft Remote Desktop natively integrated in IGEL OS 10.03.500 and later
- VMware Horizon View (Blast) natively integrated in IGEL OS 10.08.230 and later - 1
- VMware Horizon View (PCoIP) natively integrated in IGEL OS 10.08.230 and later - 1
- VMware Horizon View (RDP) natively integrated in IGEL OS 10.08.230 and later - 1

> **Note**
>
> Compatibility with previous IGEL OS releases is available by contacting deviceTRUST Support.

Microsoft Windows

Compatible operating systems (deviceTRUST Agent and Console):

- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later

Compatible operating systems (deviceTRUST Client Extension):

- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later

Compatible technologies:

- Amazon WorkSpaces (PCoIP)
- Azure Virtual Desktop
- Azure Active Directory
- Citrix Virtual Apps and Desktops
- Citrix Cloud
- Microsoft Remote Desktop Services
- Parallels Remote Application Server
- VMware Horizon View (Blast)
- VMware Horizon View (PCoIP)
- VMware Horizon View (RDP)

Ubuntu Desktop

Compatible operating systems (deviceTRUST Client Extension):

- Ubuntu Desktop 18.04 LTS and later

Compatible technologies:

- Amazon WorkSpaces (PCoIP)
- Citrix Virtual Apps and Desktops

- Citrix Cloud
- Microsoft Remote Desktop Services (via FreeRDP 2)
- VMware Horizon View (Blast)
- VMware Horizon View (PCoIP)
- VMware Horizon View (RDP)

Unicon™ eLux

Compatible technologies (deviceTRUST Client Extension):

- Citrix Virtual Apps and Desktops natively integrated in eLux RP 6.5 and later
- Citrix Cloud natively integrated in eLux RP 6.5 and later
- Microsoft Remote Desktop natively integrated in eLux RP 6.5 and later

Platform Compatibility

- Amazon Web Services
- Citrix Systems
- Microsoft
- Parallels
- VMware

Amazon Web Services

Compatible technologies:

- Amazon WorkSpaces (PCoIP)

Compatible operating systems:

- Apple macOS 10.13 and later
- IGEL OS 11.08.200 or later - 1
- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later
- Ubuntu Desktop 18.04 LTS

> **Note**
>
> 1 Compatibility with previous IGEL OS releases is available by contacting deviceTRUST® Support.

Citrix Systems

Compatible technologies:

- Citrix Virtual Apps and Desktops™
- Citrix Cloud™

Compatible operating systems:

- Apple iOS 11.x and later
- Apple iPadOS 11.x and later
- Apple macOS 10.13 and later
- IGEL OS 10.03.500 or later
- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later
- Ubuntu Desktop 18.04 LTS and later
- Unicon eLux RP 6.5 and later

Microsoft

Compatible technologies:

- Microsoft Remote Desktop Services
- Azure Virtual Desktop - 1
- Azure Active Directory - 2

Compatible operating systems:

- Apple iOS 11.x and later - 1
- Apple iPadOS 11.x and later - 1
- Apple macOS 10.13 and later (via FreeRDP 2)
- IGEL OS 11.08.200 or later - 3(via RDP Session)
- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later
- Ubuntu Desktop 18.04 LTS and later (via FreeRDP 2)

> **Note**
>
> - 1 Compatibility with Azure Virtual Desktop is available on Microsoft Windows and IGEL OS only.
> - 2 ompatibility with Azure Active Directory provided by deviceTRUST Agent on Microsoft Windows only.

> • 4 Compatibility with previous IGEL OS releases is available by contacting deviceTRUST Support.

Parallels

Compatible technologies:

- Parallels Remote Application Server

Compatible operating systems:

- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later

VMware

Compatible technologies:

- VMware Horizon View (Blast)
- VMware Horizon View (PCoIP)
- VMware Horizon View (RDP)

Compatible operating systems:

- Apple iOS 11.x and later
- Apple iPadOS 11.x and later
- Apple macOS 10.13 and later
- IGEL OS 10.08.230 or later - 1
- Microsoft Windows 10 and later
- Microsoft Windows Server 2012 and later
- Ubuntu Desktop 18.04 LTS and later

> **Note**
>
> 1 Compatibility with previous IGEL OS releases is available by contacting deviceTRUST Support.

Getting Started

deviceTRUST® requires some simple but essential configuration steps to be performed to enable deviceTRUST functionality for your remoting environments or for your local devices.

Scenario: Remote

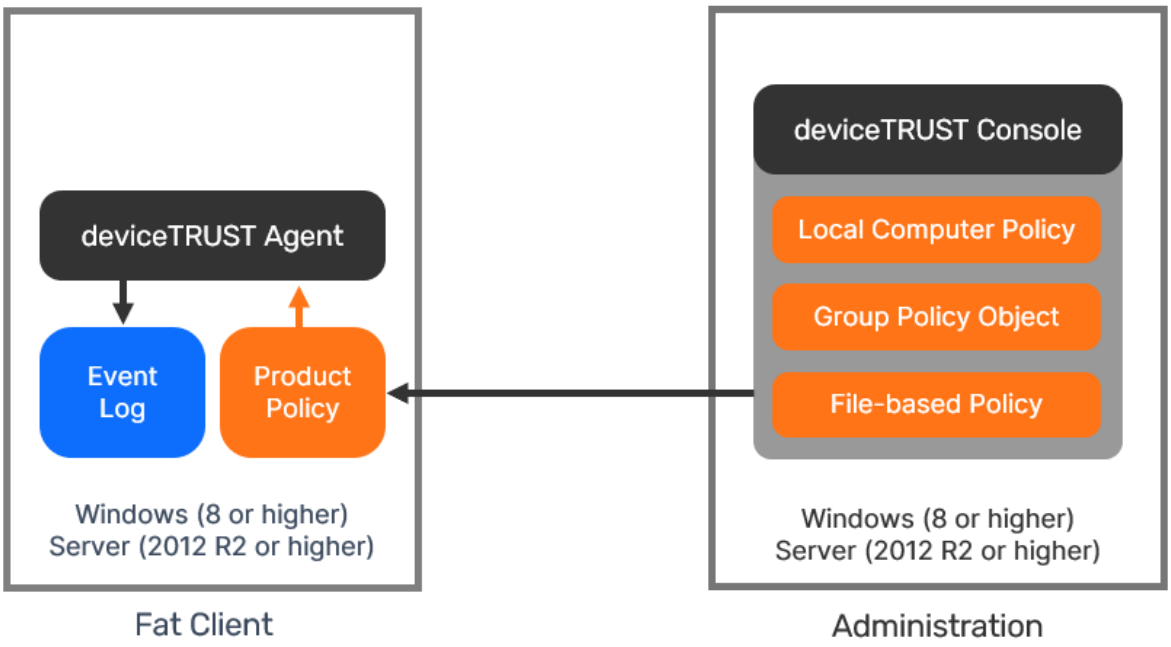
In remote scenarios, deviceTRUST transports the context information from the users remote device into the virtual session where the configuration is enforced. Please visit Getting Started for Remote Devices to begin the guide.

Scenario: Local

In local scenarios, deviceTRUST collects context information and executes actions locally. Please visit Getting Started for Local Devices to begin the guide.

Getting Started for Local

deviceTRUST® requires some simple but essential configuration steps to be performed to enable deviceTRUST functionality for your local devices. We will guide you step-by-step through simple deviceTRUST installation and configuration steps to enable deviceTRUST with an unauthorized USB drives use case for your local devices.



We will perform the following steps:

- Step 1: Download the deviceTRUST setup binaries
- Step 2: Install the deviceTRUST Agent
- Step 3: Install the deviceTRUST Console
- Step 4: Enter your deviceTRUST License

Step 1: Download the deviceTRUST setup binaries

The latest deviceTRUST software can be found on our Download page and your personalized license can be found within your product license certificate.

Step 2: Install the deviceTRUST Agent

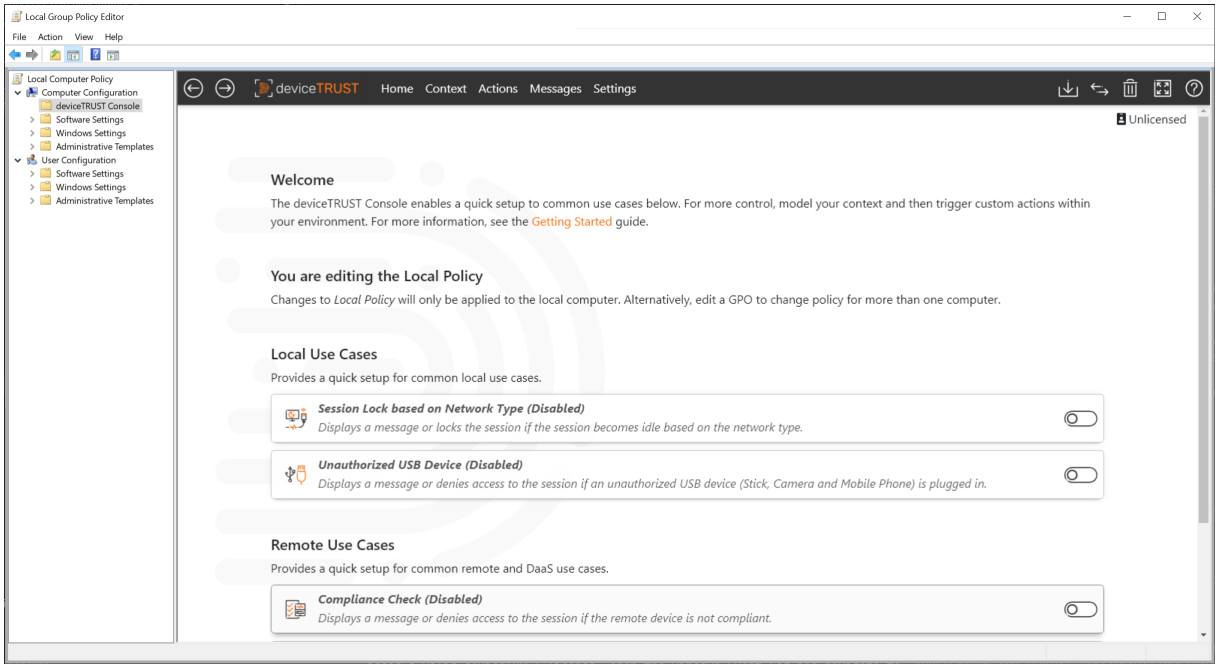
Start the installation of the deviceTRUST Agent on your local device. Follow the steps in the section Installing the Agent to complete the installation.

Step 3: Install the deviceTRUST Console

To configure and to apply contextual security policies to the deviceTRUST Agent you need to use the deviceTRUST Console. The deviceTRUST Console supports various ways to provide the contextual security policies to the deviceTRUST Agent. Those options are using the Local Policy Editor, a Group Policy Object (GPO) or file-based.

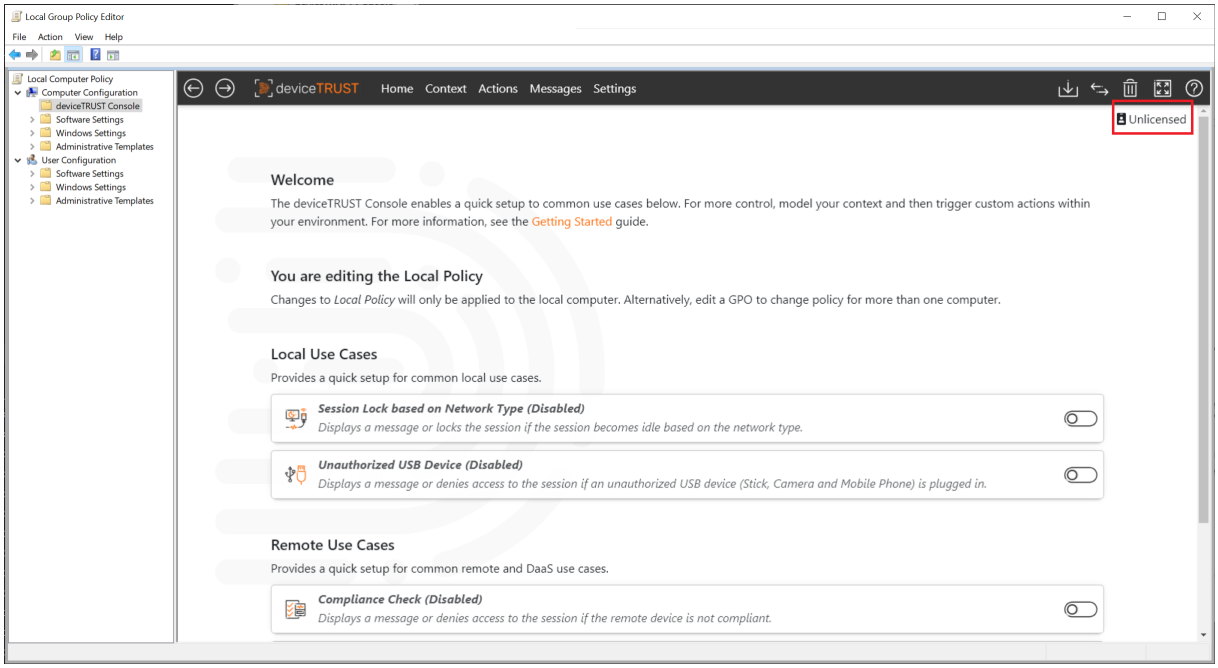
Within the Getting Started Guide, for simplicity, we use the Local Policy Editor to quickly and efficiently create, edit, and use contextual security policies. Follow the steps in the section Installing the Console to complete the installation.

The deviceTRUST Console includes a node within the Local Policy Editor `COMPUTER CONFIGURATION \DEVICETRUST CONSOLE` which can be used to model the context of a user, and then act on changes to that context by triggering custom actions within your environment.

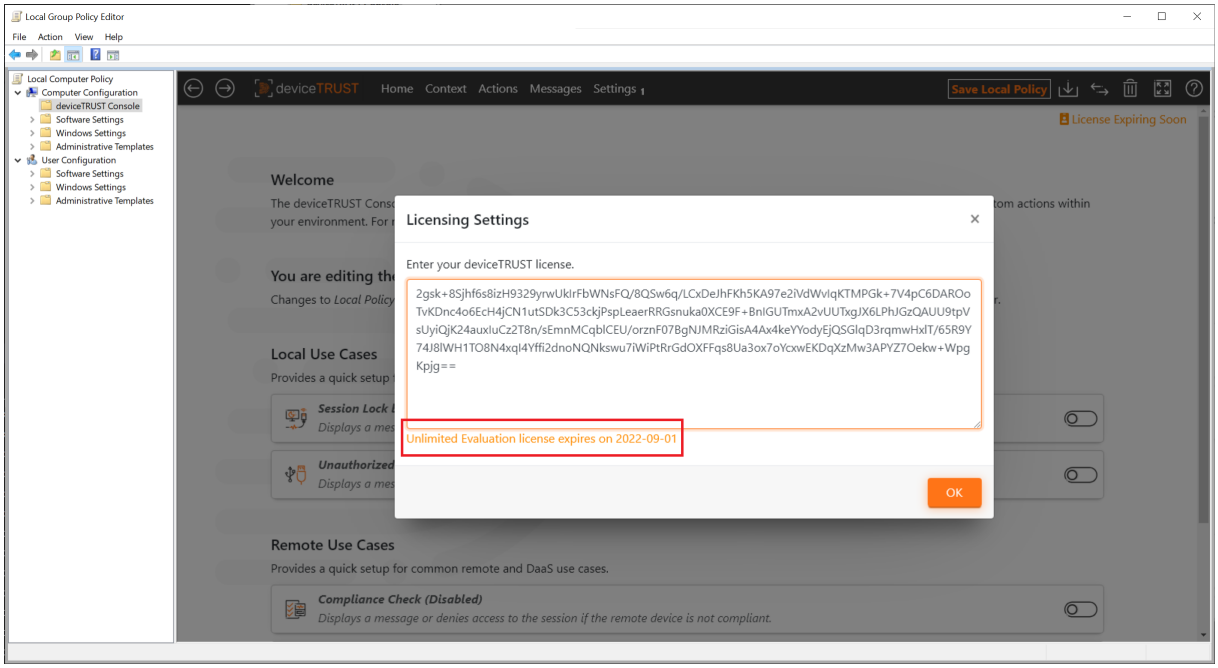


Step 4: Enter your deviceTRUST License

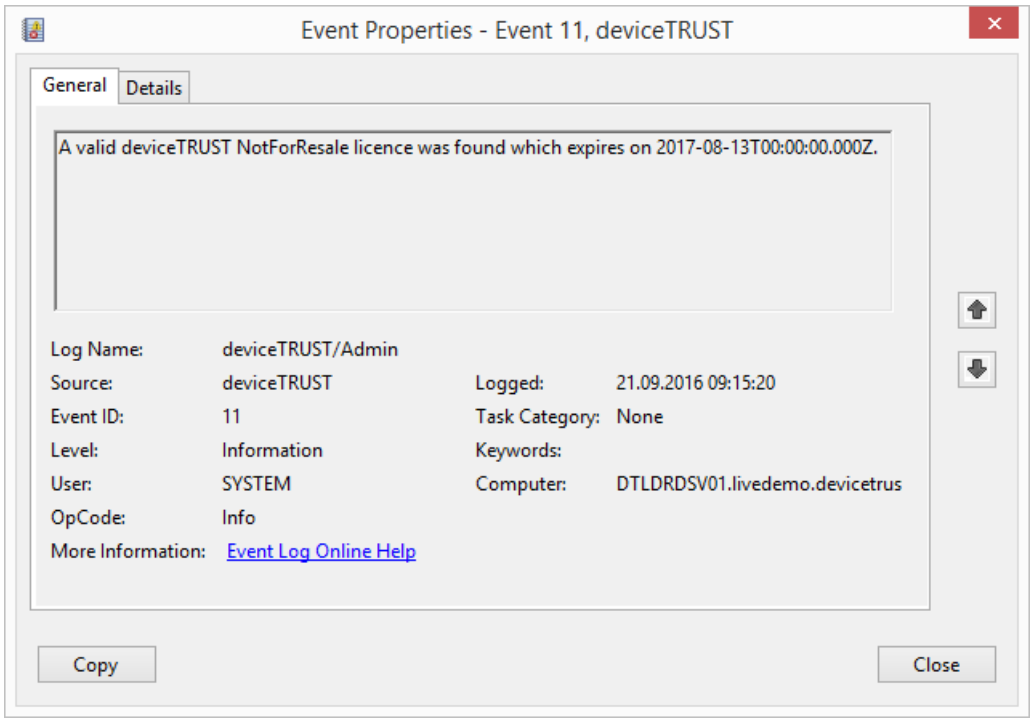
To add the license into the deviceTRUST contextual security policy open the Local Policy Editor and navigate to `DEVICETRUST CONSOLE` and click on the `UNLICENSED` link on the homepage.



Enter your deviceTRUST license and make sure it is valid. Close the license editor with `OK` and click on `SAVE TO LOCAL COMPUTER POLICY` in the top right toolbar.

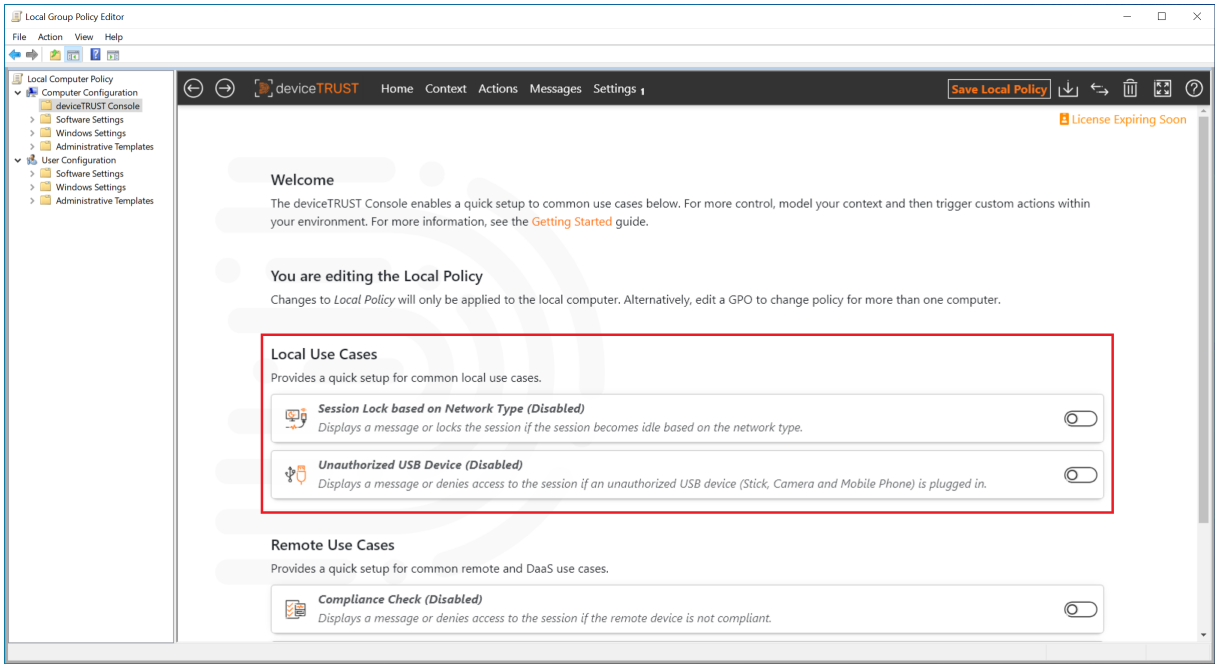


deviceTRUST is now enabled and will work for all users except local administrators connecting to that remoting or DaaS host system with deviceTRUST Agent installed. To check if you have added a valid deviceTRUST license, open the Windows Event Log and navigate to `APPLICATION AND SERVICE LOGS\DEVICETRUST\ADMIN` and check for the existence of event ID 11 which states that your deviceTRUST license is valid.

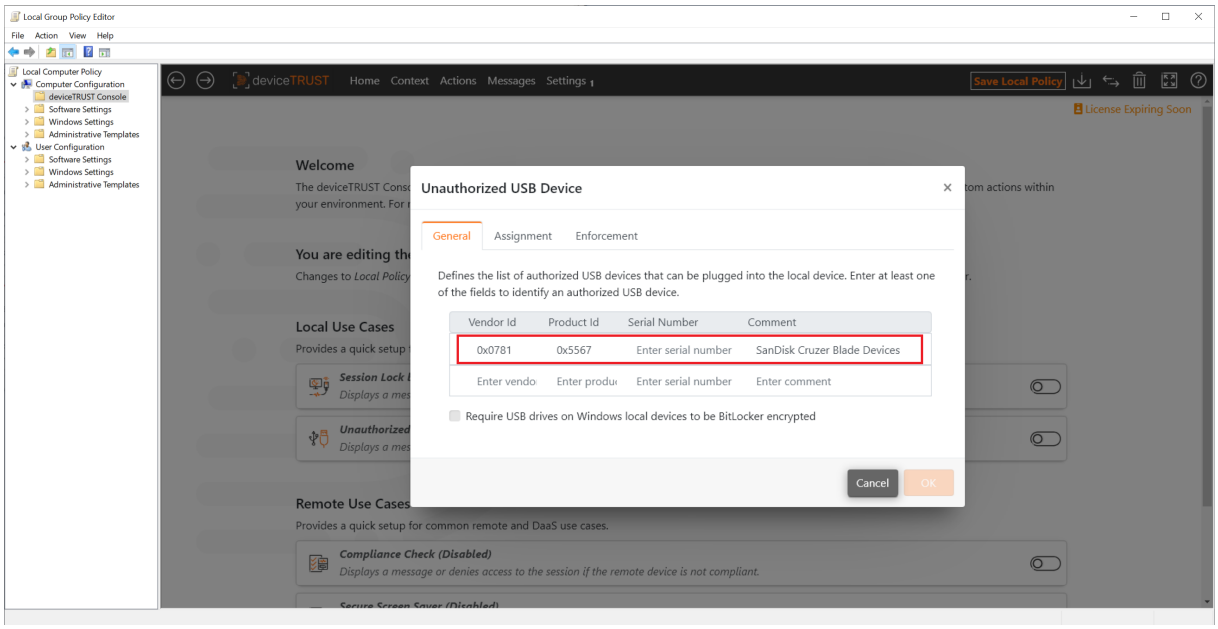


Step 5: Enable the Unauthorized USB Drive use case

We will use the deviceTRUST Console to create a contextual security policy that makes access to the session dependent on whether the USB device being used has been authorized. The deviceTRUST Console includes a set of use cases which can be used to quickly implement a use case. Launch the deviceTRUST Console and navigate to LOCAL USE CASES on the homepage.

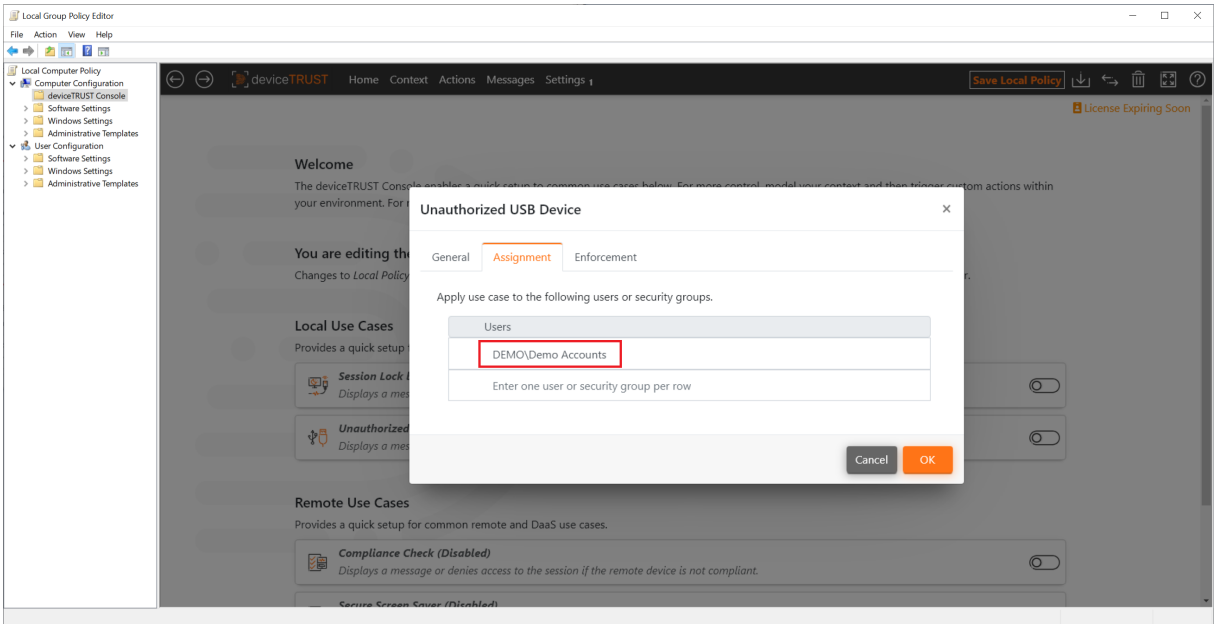


Select the UNAUTHORIZED USB DEVICE use case and add authorized USB devices on the GENERAL tab into the list of authorized USB devices.

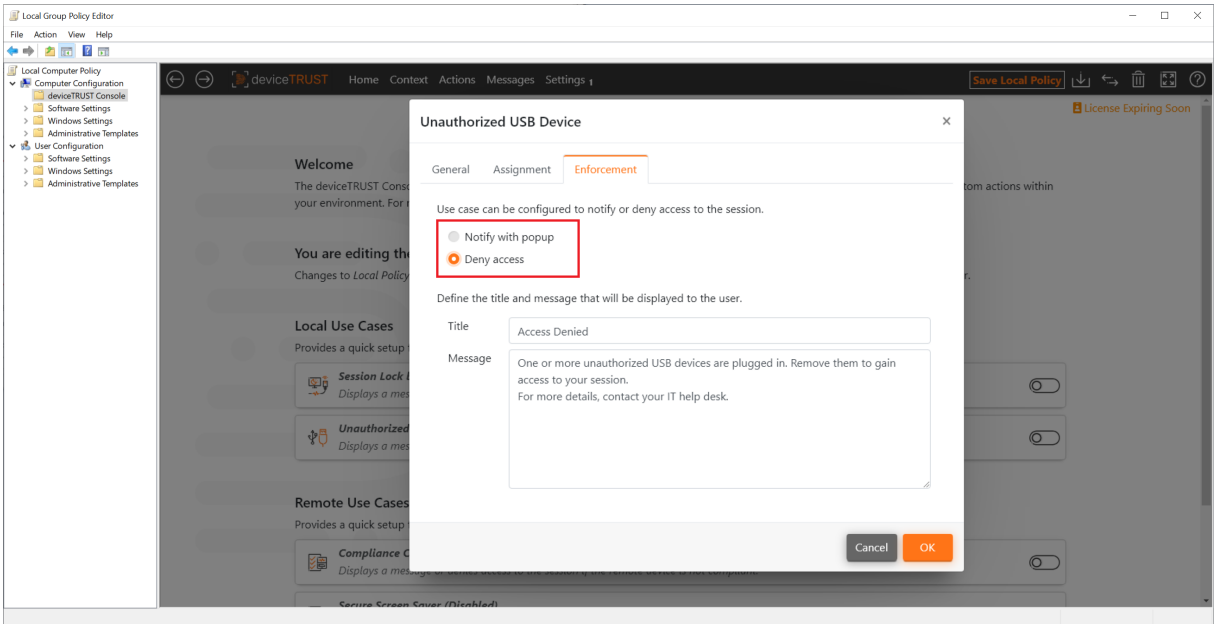


Click on the ASSIGNMENT configuration tab and add USERS and / or SECURITY GROUPS to apply

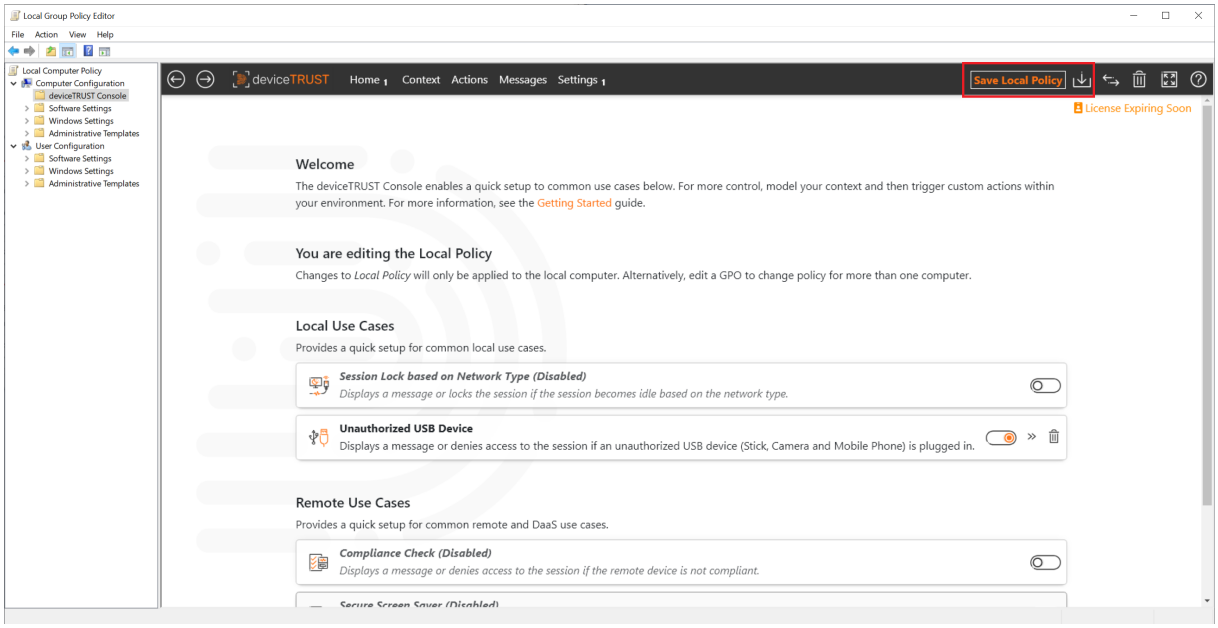
the use case for.



Click on the ENFORCEMENT configuration tab and select DENY ACCESS.

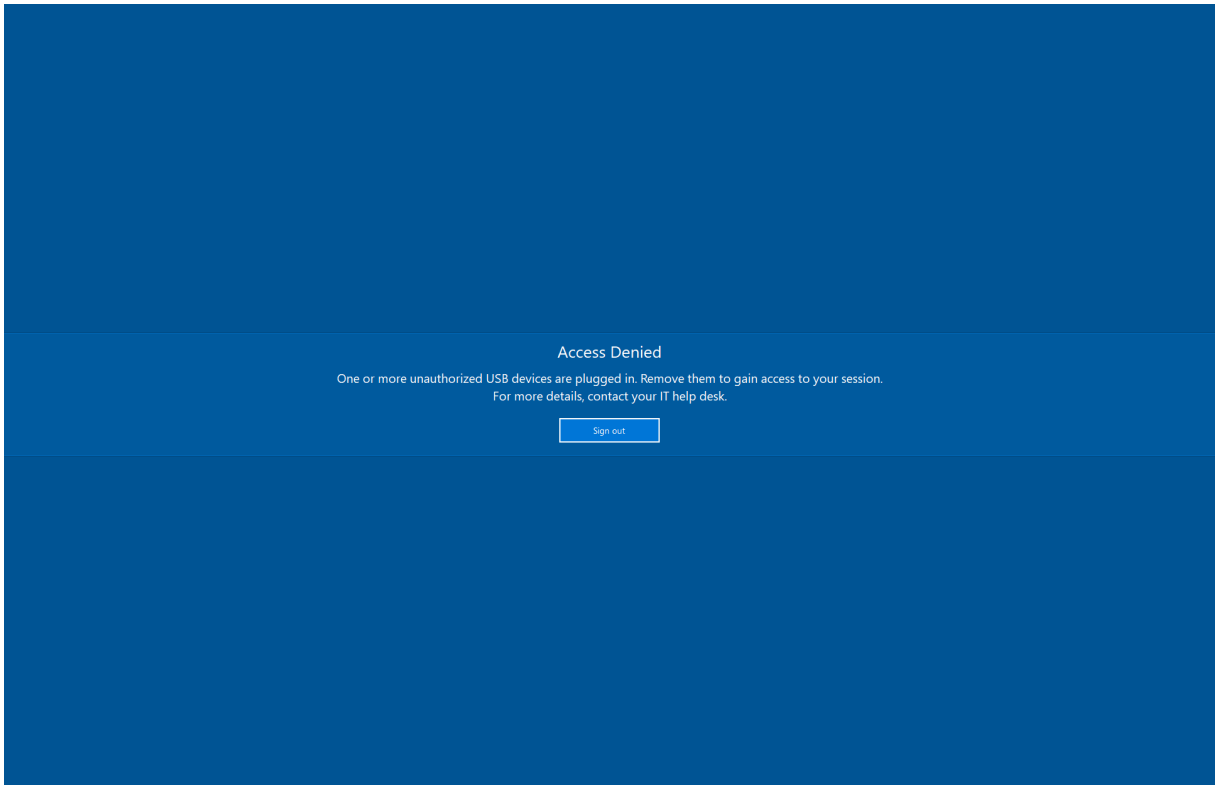


Click on SAVE TO LOCAL COMPUTER POLICY in the top right toolbar to save the unauthorized USB device use case to the local computer policy.



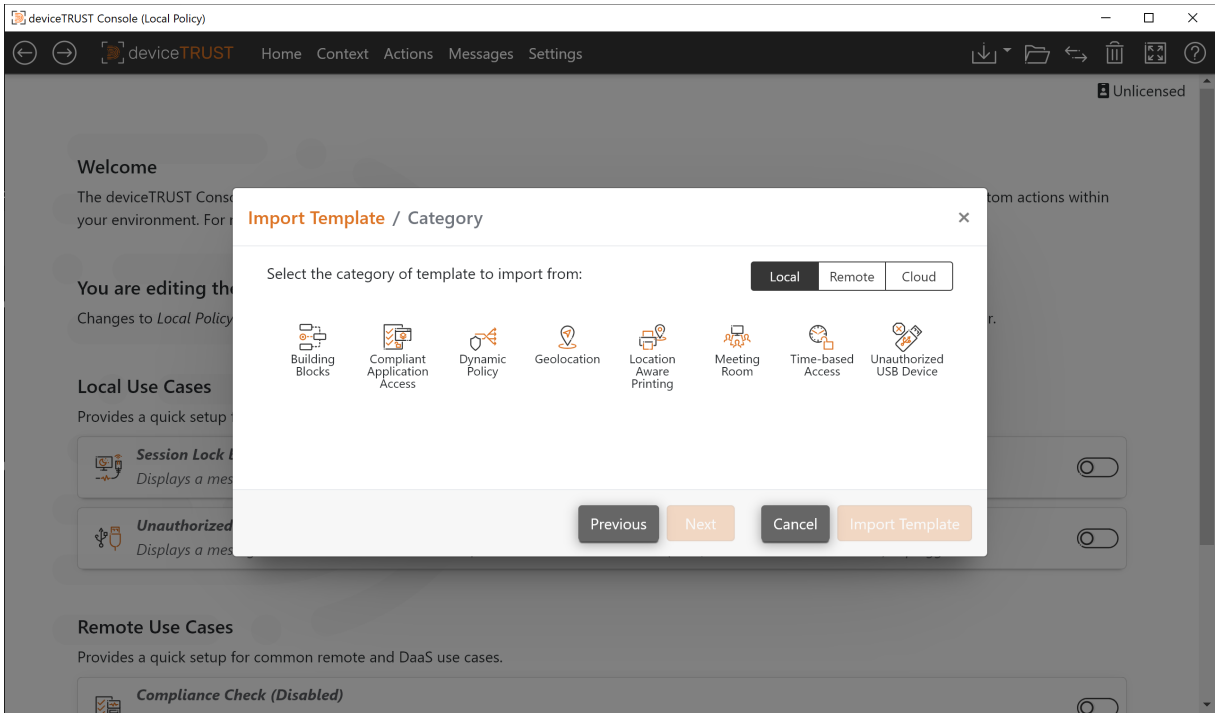
Step 6: Test the Unauthorized USB Device use case

Sign in with a non-administrative user account to the local device and then plug in an authorized USB device at runtime. The authorized USB device is displayed in Windows Explorer and can be used. Now plug in an unauthorized USB device in addition or exclusively to see how deviceTRUST can easily and dynamically control access to the session depending on the USB device in use.



Next steps

You have now successfully implemented your first use case with deviceTRUST for your local devices. Feel free to check out our additional use cases provided on the deviceTRUST Console homepage under LOCAL USE CASES. In addition, the deviceTRUST Console gives you access to many more configuration Templates for a wide variety of use cases.

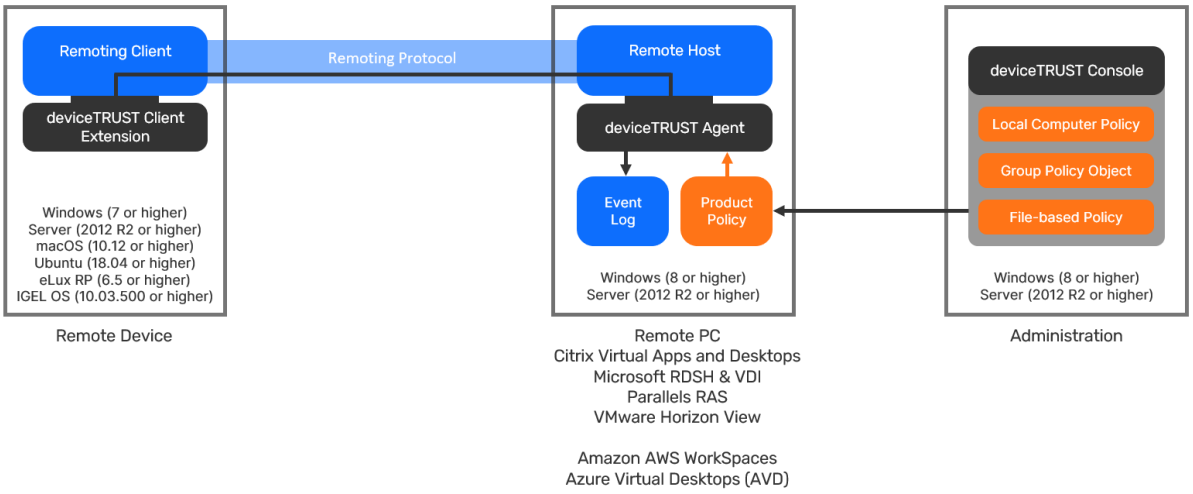


Troubleshooting

If your deviceTRUST installation or configuration does not work as expected, you can use the Troubleshooting guide to start troubleshooting.

Getting Started for Remote

deviceTRUST® requires some simple but essential configuration steps to be performed to enable deviceTRUST functionality for your remote environments. We will guide you step-by-step through simple deviceTRUST installation and configuration steps to enable deviceTRUST with a compliance check use case within your remote environment.



We will perform the following steps:

- Step 1: Download the deviceTRUST setup binaries
- Step 2: Install the deviceTRUST Agent
- Step 3: Install the deviceTRUST Console
- Step 4: Enter your deviceTRUST License
- Step 5: Install the deviceTRUST Client Extension on a Microsoft Windows device
- Step 6: Enable the Compliance Check use case
- Step 7: Check that access is denied when the deviceTRUST Client Extension is not installed
- Step 8: Test the Compliance Check use case from a Microsoft Windows device

Step 1: Download the deviceTRUST setup binaries

The latest deviceTRUST software can be found on our Download page and your personalized license can be found within your product license certificate.

Step 2: Install the deviceTRUST Agent

Start the installation of the deviceTRUST Agent on your remoting or DaaS host system, which can be Amazon WorkSpaces, Citrix Virtual Apps and Desktops (CVAD), Microsoft Azure Virtual Desktop (AVD), Microsoft Remote Desktop Session Host (RDSH) or VMware Horizon View. Follow the steps in the section Installing the Agent to complete the installation.

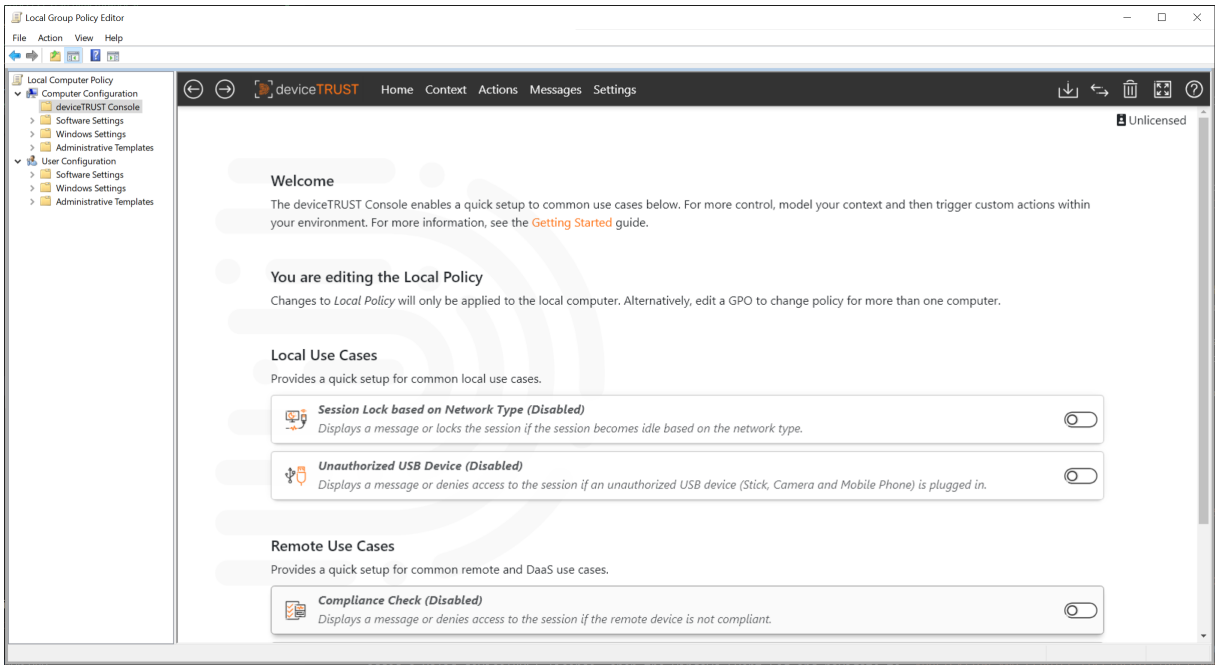
Step 3: Install the deviceTRUST Console

To configure and to apply contextual security policies to the deviceTRUST Agent you need to use the deviceTRUST Console. The deviceTRUST Console supports various ways to provide the contextual

security policies to the deviceTRUST Agent. Those options are using the Local Policy Editor, a Group Policy Object (GPO) or file-based.

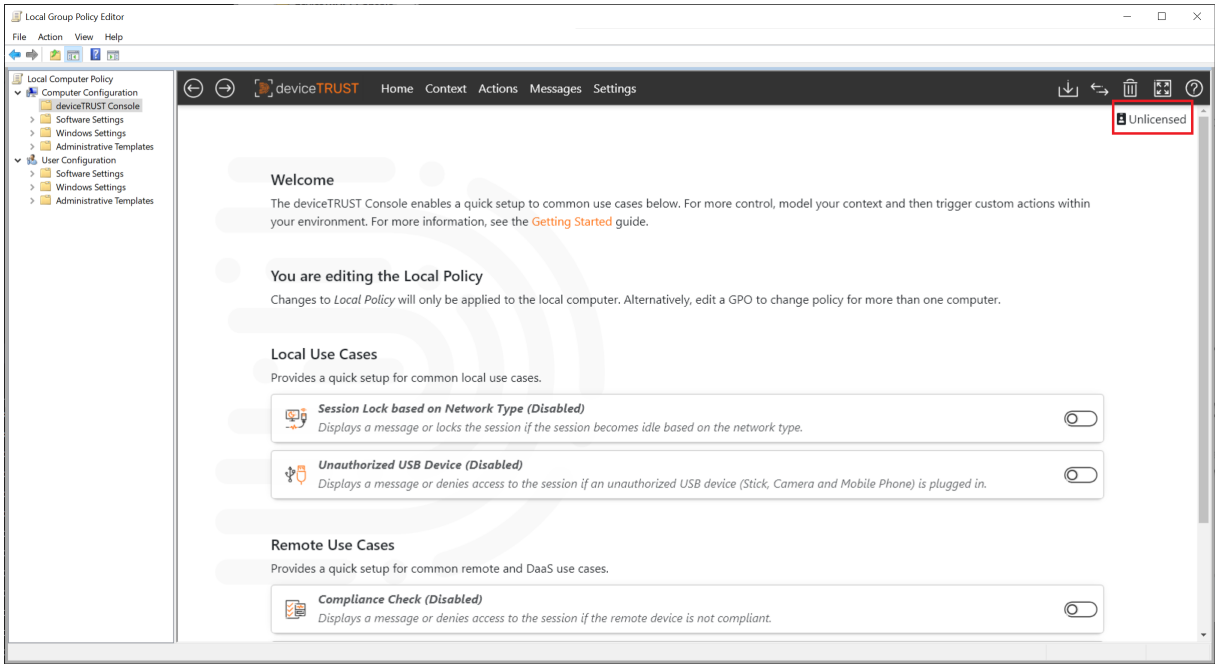
Within the Getting Started Guide, for simplicity, we use the Local Policy Editor to quickly and efficiently create, edit, and use contextual security policies. Follow the steps in the section Installing the Console to complete the installation.

The deviceTRUST Console includes a node within the Local Policy Editor `COMPUTER CONFIGURATION \DEVICETRUST CONSOLE` which can be used to model the context of a user, and then act on changes to that context by triggering custom actions within your environment.

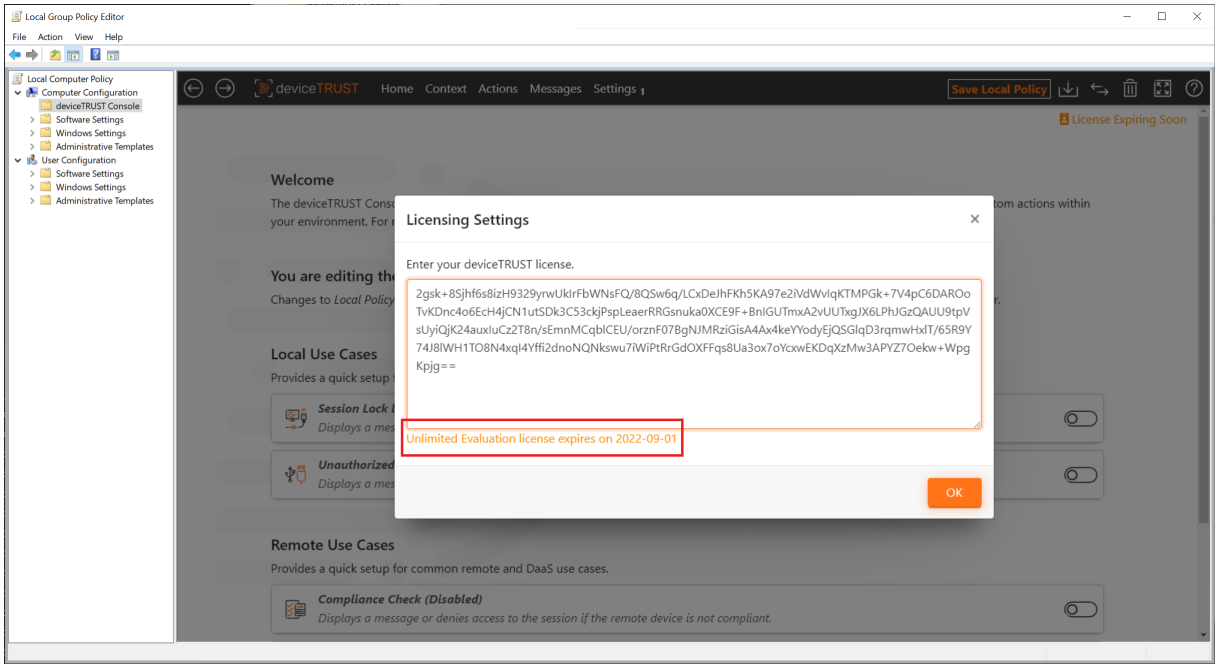


Step 4: Enter your deviceTRUST License

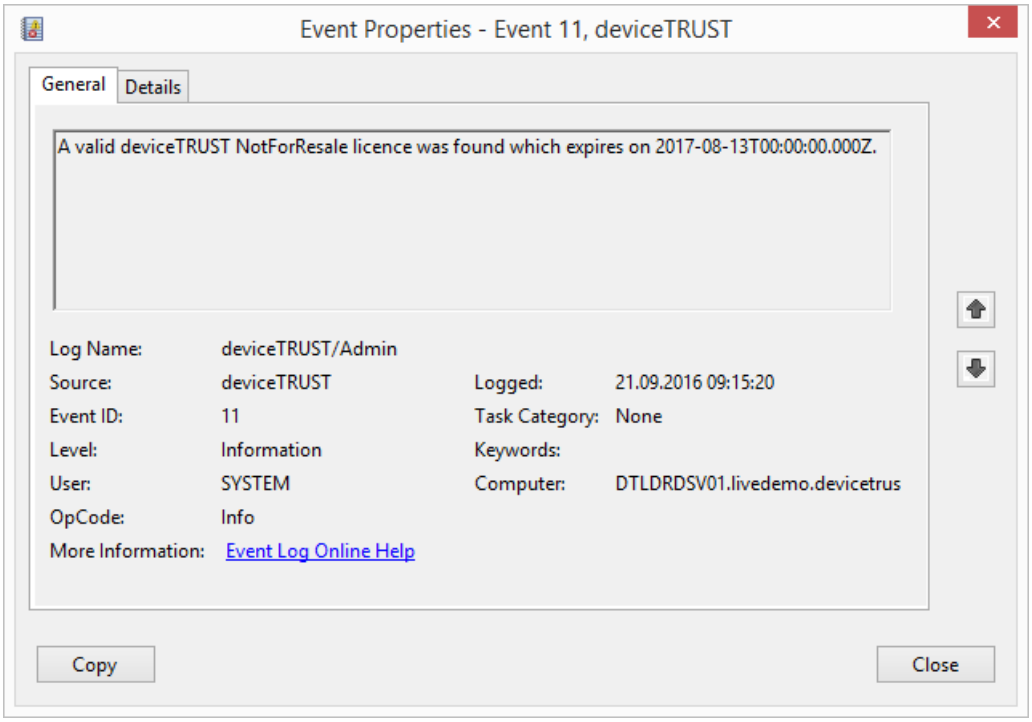
To add the license into the deviceTRUST contextual security policy open the Local Policy Editor and navigate to `DEVICETRUST CONSOLE` and click on the `UNLICENSED` link on the homepage.



Enter your deviceTRUST license and make sure it is valid. Close the license editor with `OK` and click on `SAVE TO LOCAL COMPUTER POLICY` in the top right toolbar.



deviceTRUST is now enabled and will work for all users except local administrators connecting to that remoting or DaaS host system with deviceTRUST Agent installed. To check if you have added a valid deviceTRUST license, open the Windows Event Log and navigate to `APPLICATION AND SERVICE LOGS\DEVICETRUST\ADMIN` and check for the existence of event ID 11 which states that your deviceTRUST license is valid.

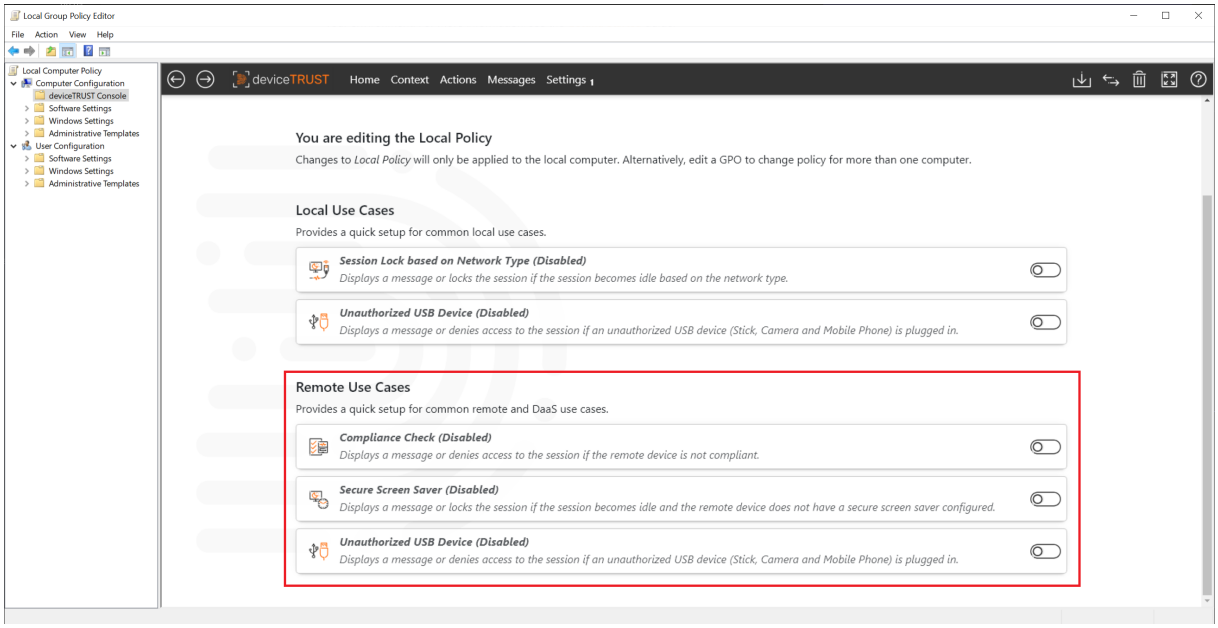


Step 5: Install the deviceTRUST Client Extension on a Microsoft Windows device

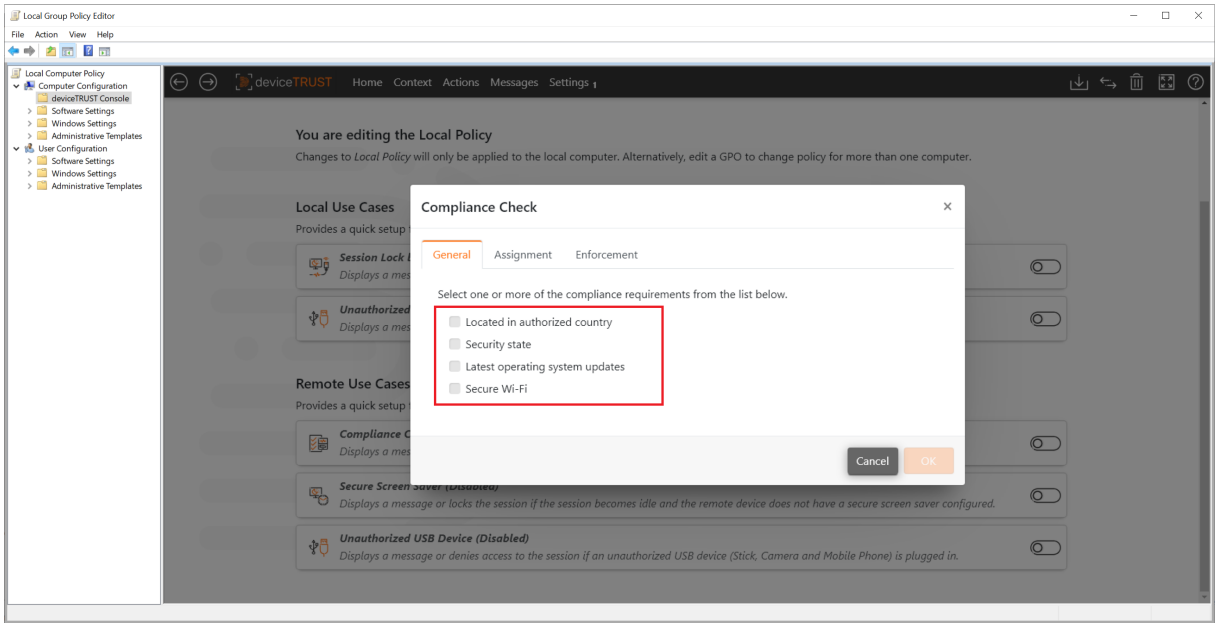
Within the Getting Started Guide, for simplicity, we will only install the deviceTRUST Client Extension on a Microsoft Windows device. Other device operating systems are also supported and an overview of how to install the deviceTRUST Client Extension on the particular operating system can be found on the Installation Client Extension page. Now follow the steps in the section Installing the Client Extension on Microsoft Windows device to complete the installation.

Step 6: Enable the Compliance Check use case

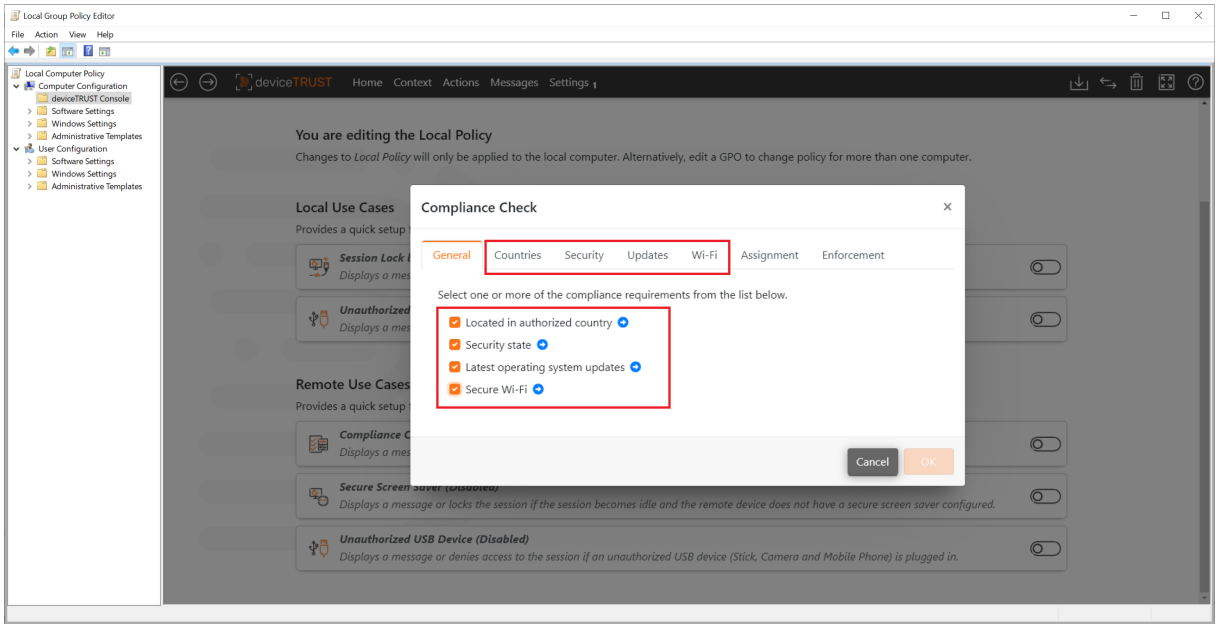
We will use the deviceTRUST Console to create a contextual security policy which controls access to the session depending upon the compliance state of the remote device. The deviceTRUST Console includes a set of use cases which can be used to quickly implement a use case. Launch the deviceTRUST Console and navigate to `REMOTE USE CASES` on the homepage.



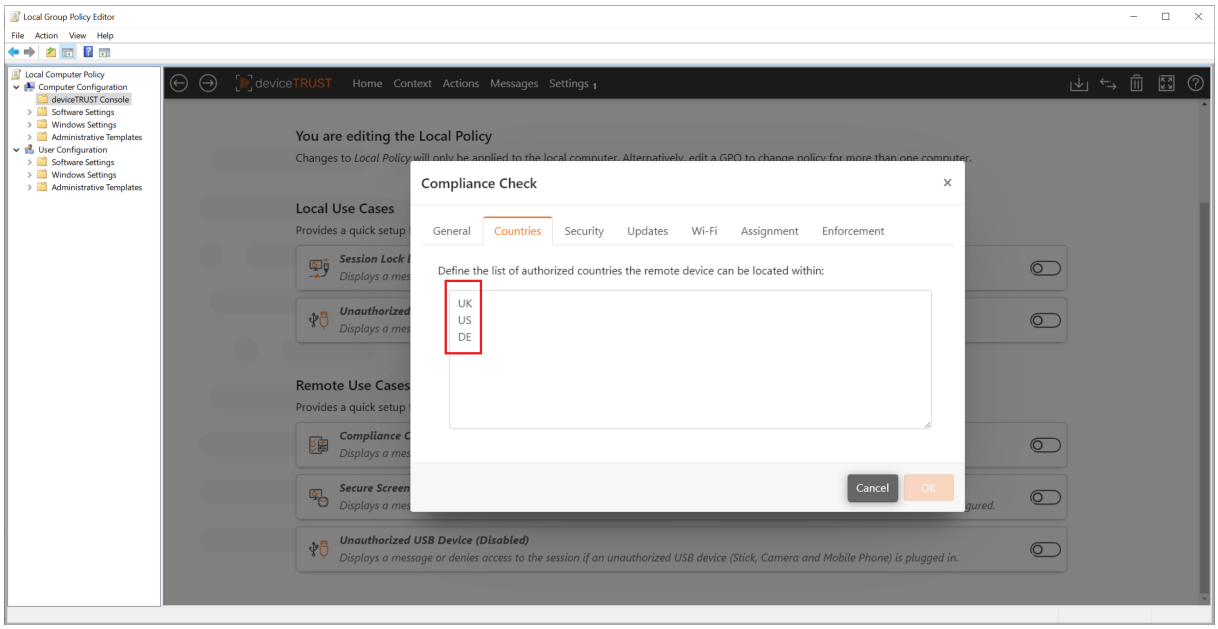
Select the `COMPLIANCE CHECK` use case, select on the `GENERAL` tab all options to be included in the compliance check.



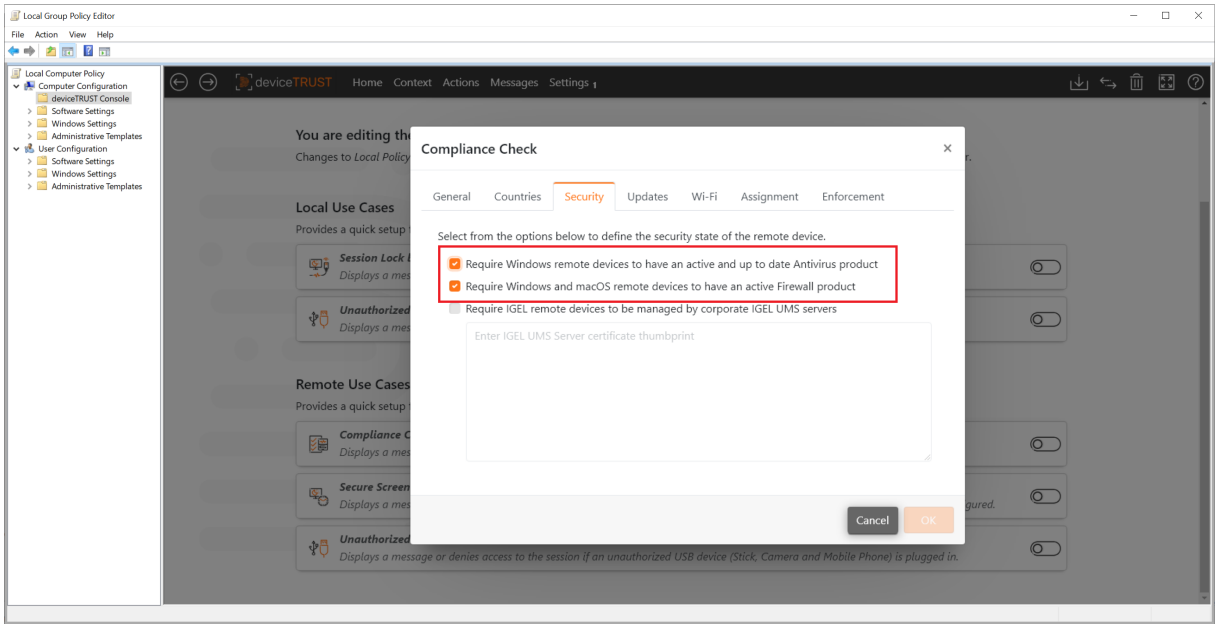
New configuration tabs will become visible.



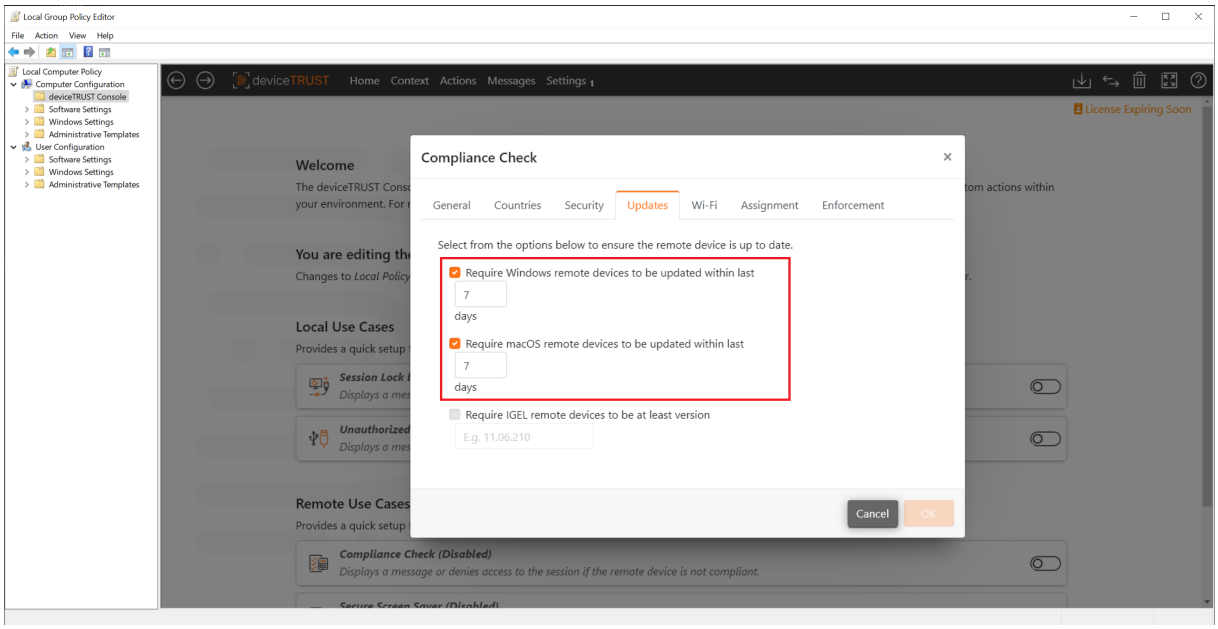
Click on the COUNTRY configuration tab and add all authorized countries using ISO 3166-1 Alpha-2 code.



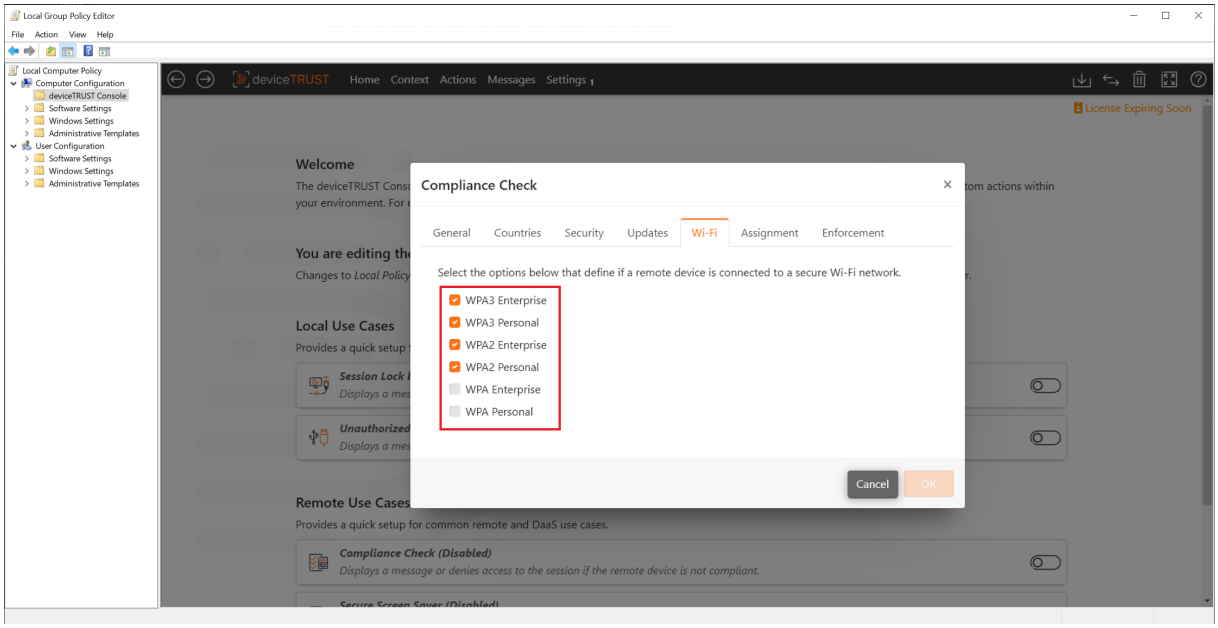
Click on the SECURITY configuration tab and enable the REQUIRE WINDOWS REMOTE DEVICES TO HAVE AN ACTIVE AND UP TO DATE ANTIVIRUS PRODUCT and REQUIRE WINDOWS AND MACOS REMOTE DEVICES TO HAVE AN ACTIVE FIREWALL PRODUCT options.



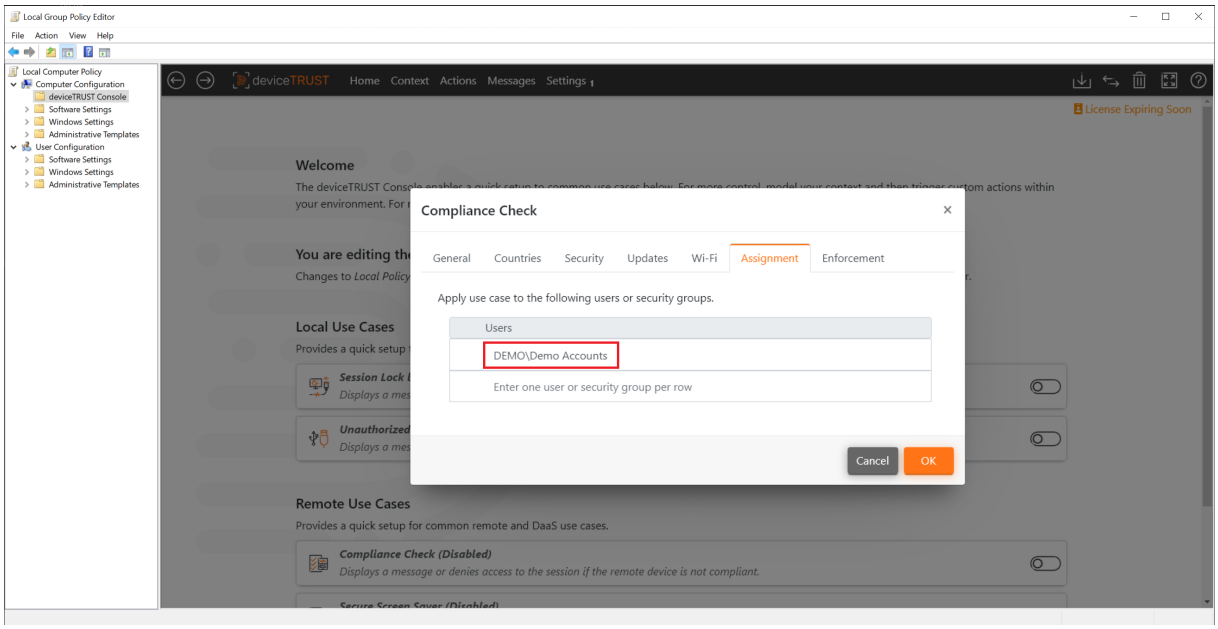
Click on the `UPDATES` configuration tab and enable the `REQUIRE WINDOWS REMOTE DEVICES TO BE UPDATED WITHIN THE LAST 7 DAYS` and `REQUIRE MACOS REMOTE DEVICES TO BE UPDATED WITHIN THE LAST 7 DAYS` options.



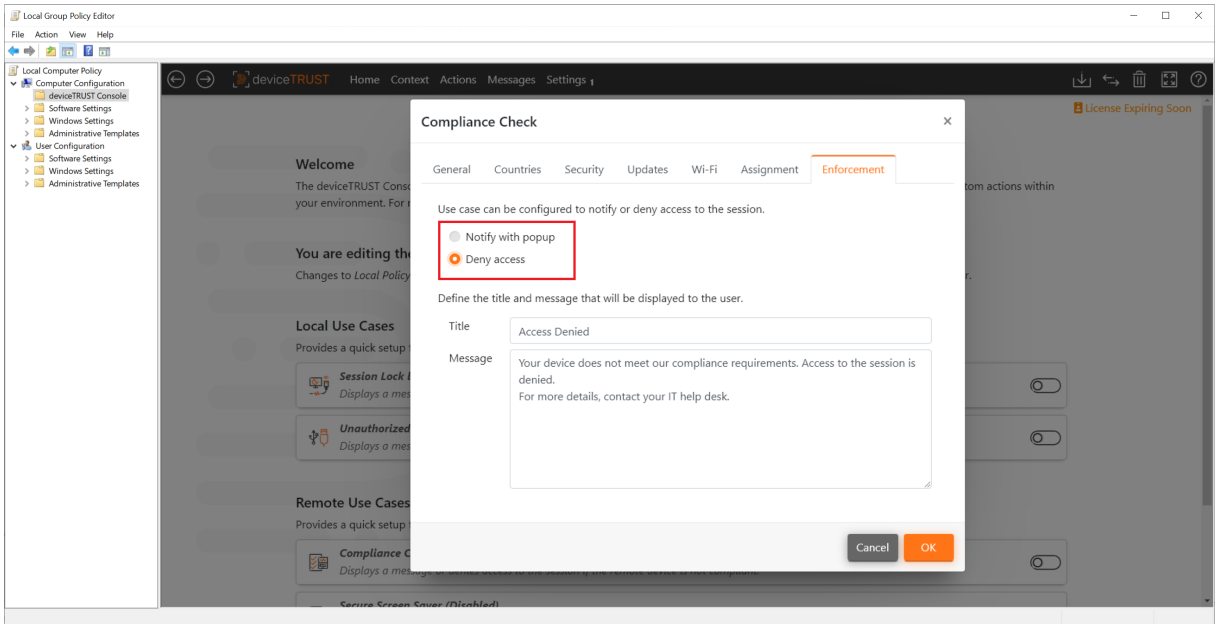
Click on the `WI-FI` configuration tab and enable the `WPA3 ENTERPRISE`, `WPA3 PERSONAL`, `WPA2 ENTERPRISE` and `WPA2 PERSONAL` options.



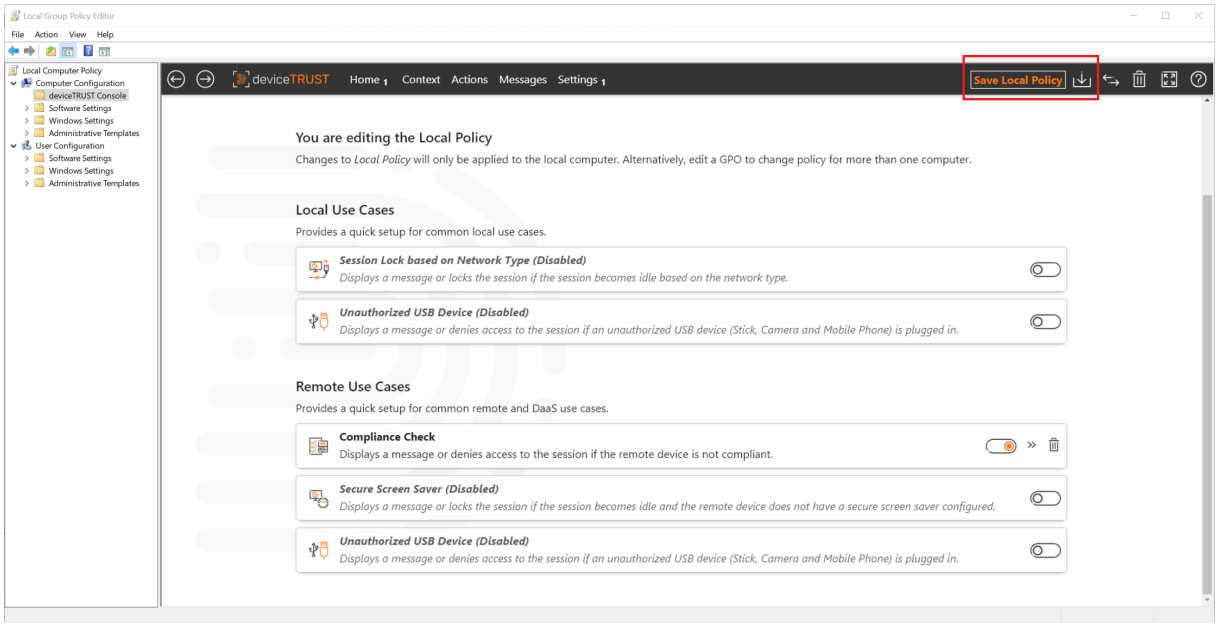
Click on the `ASSIGNMENT` configuration tab and add `USERS` and / or `SECURITY GROUPS` to apply the use case for.



Click on the `ENFORCEMENT` configuration tab and select `DENY ACCESS`.

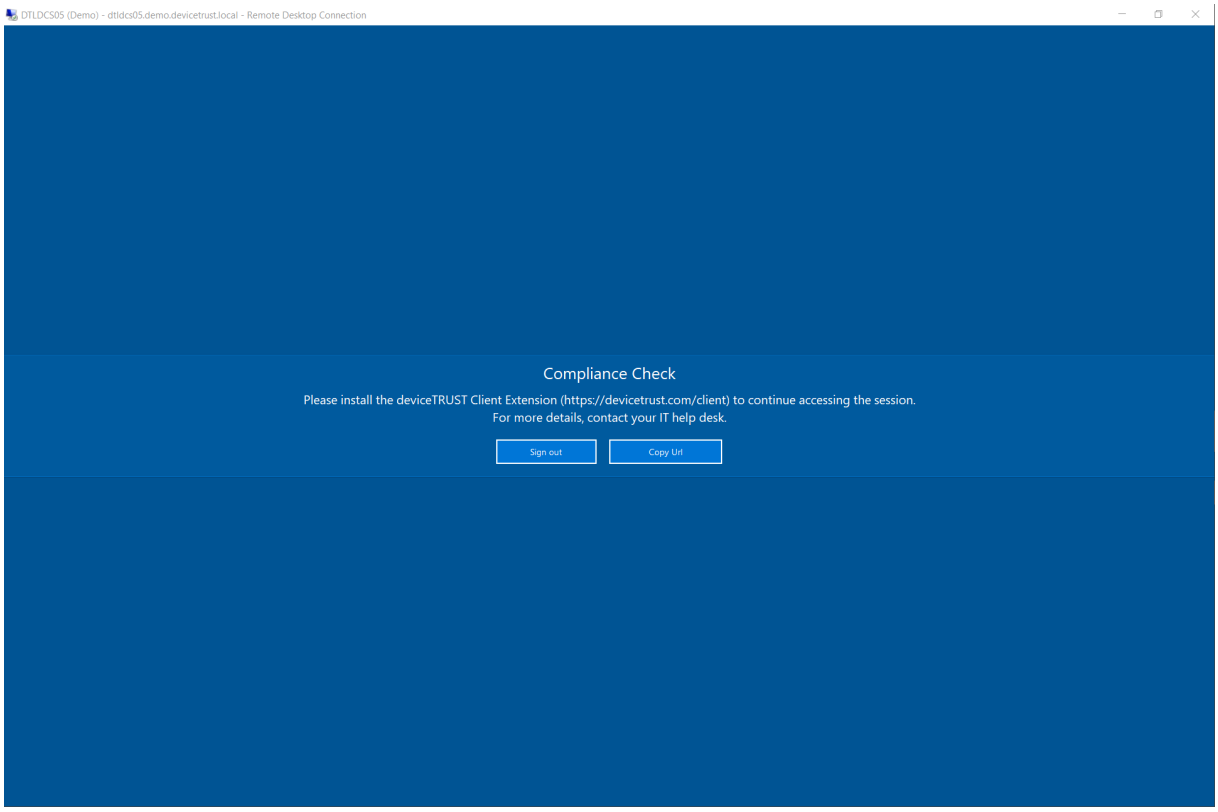


Click on `SAVE TO LOCAL COMPUTER POLICY` in the top right toolbar to save the complianc check use case to the local computer policy.



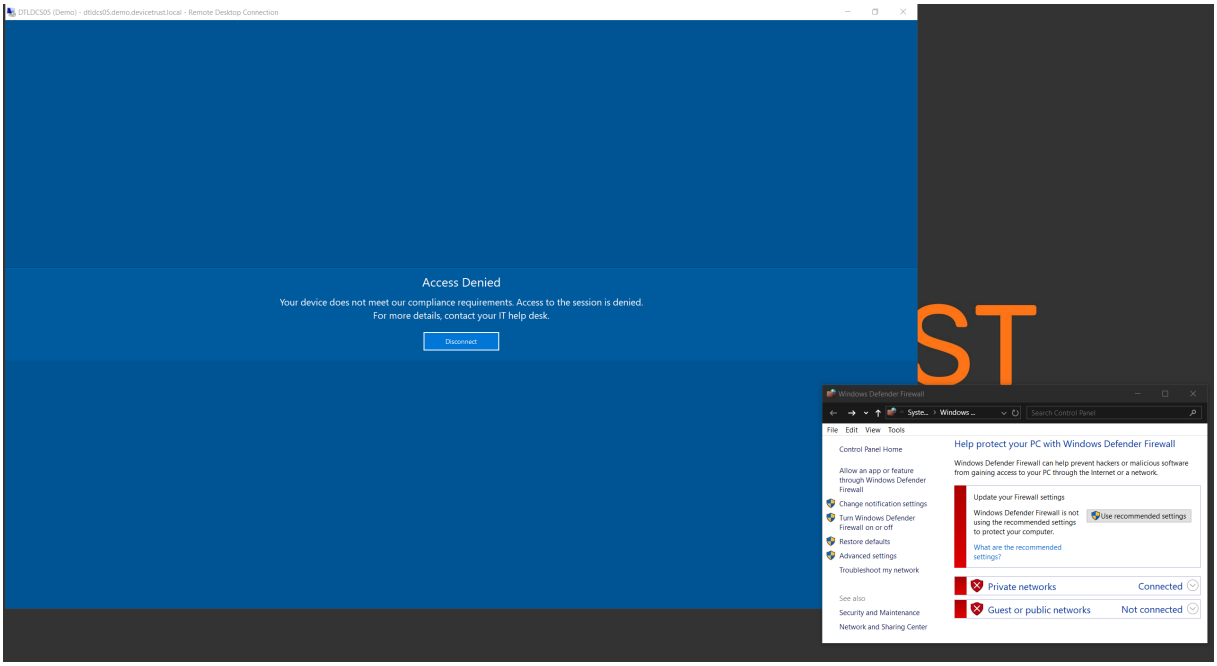
Step 7: Check that access is denied when the deviceTRUST Client Extension is not installed

From a device without the deviceTRUST Client Extension installed, connect to your remoting or DaaS host system. Because the remote device does not have an active deviceTRUST Client Extension, the access will be denied with the following message:



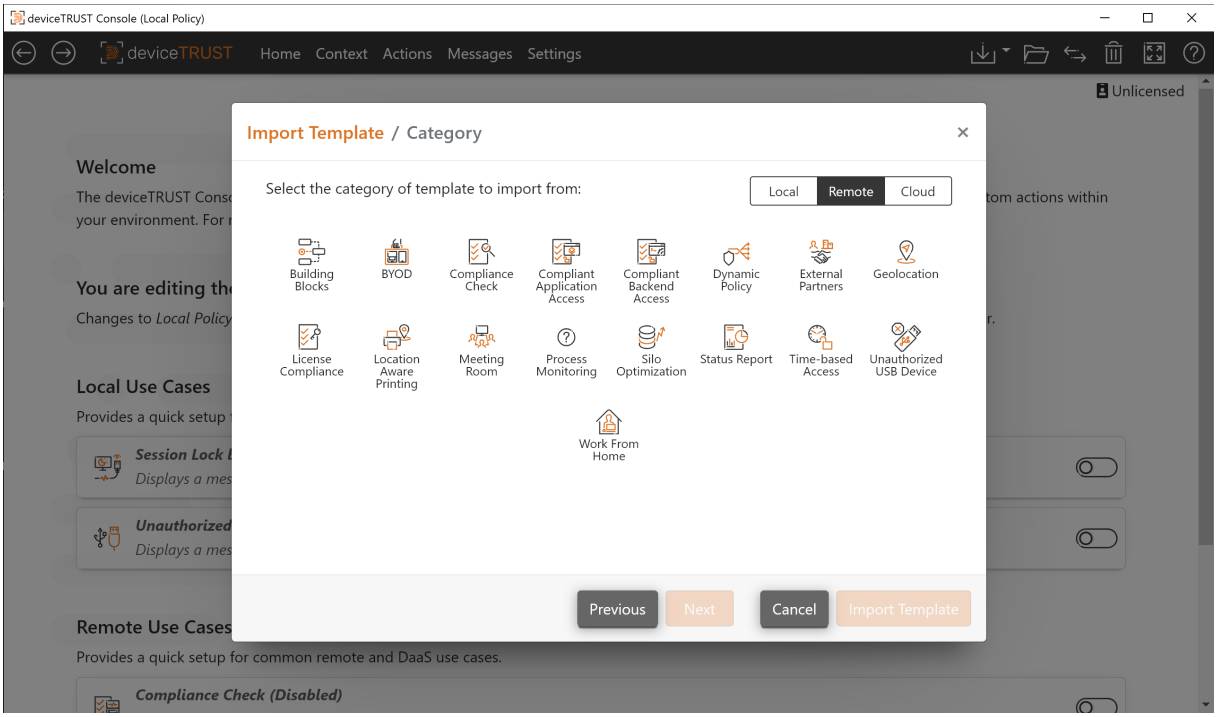
Step 8: Test the Compliance Check use case from a Microsoft Windows device

From a Microsoft Windows device with the deviceTRUST Client Extension installed, connect to your remoting or DaaS host system. Toggle the state of the Windows Defender Firewall to see how deviceTRUST can simply and dynamically control access to the session depending on the firewall state of the remote device.



Next steps

You have now successfully implemented your first use case with deviceTRUST for your remoting and DaaS environment. Feel free to check out our additional use cases provided on the deviceTRUST Console homepage under `REMOTE USE CASES`. In addition, the deviceTRUST Console gives you access to many more configuration Templates for a wide variety of use cases.



Troubleshooting

If your deviceTRUST installation or configuration does not work as expected, you can use the Troubleshooting guide to start troubleshooting.

Download deviceTRUST® Client Extensions

The latest client extensions for Windows, macOS and Ubuntu can be download by end users from the deviceTRUST Client Extension Download page.

Download deviceTRUST 21.1

The latest deviceTRUST 21.1 binaries can be downloaded from the link below:

Version	21.1.310
Release Date	18th October 2022
Link	https://storage.devicetrust.com/download/deviceTRUST-21.1.310.zip
SHA256 Hash	5F7352EE9038C544AC7F5B04000A3DE7569ABB070FAA8F774E

Download previous releases

Previous releases can be downloaded download.

Software components

After downloading the deviceTRUST software, you will find the following components within the compressed ZIP file:

Component	Description
DTAGENT-X64-RELEASE-x.x.x.x.MSI	The deviceTRUST 64-bit Agent installer.
DTAGENT-X86-RELEASE-x.x.x.x.MSI	The deviceTRUST 32-bit Agent installer.
DTCLIENT-EXTENSION-RELEASE-x.x.x.x.EXE	The deviceTRUST 32-bit and 64-bit Client Extension installer.
DTCONSOLE-X64-RELEASE-x.x.x.x.MSI	The deviceTRUST 64-bit Console installer.

Component	Description
DTCONSOLE-X86-RELEASE-x.x.x.x.MSI	The deviceTRUST 32-bit Console installer.
DTPOLICYDEFINITIONS-x.x.x.x.ZIP	The deviceTRUST ADMX policy definitions for configuring legacy options in the software from Microsoft Active Directory Group Policy (GPO).

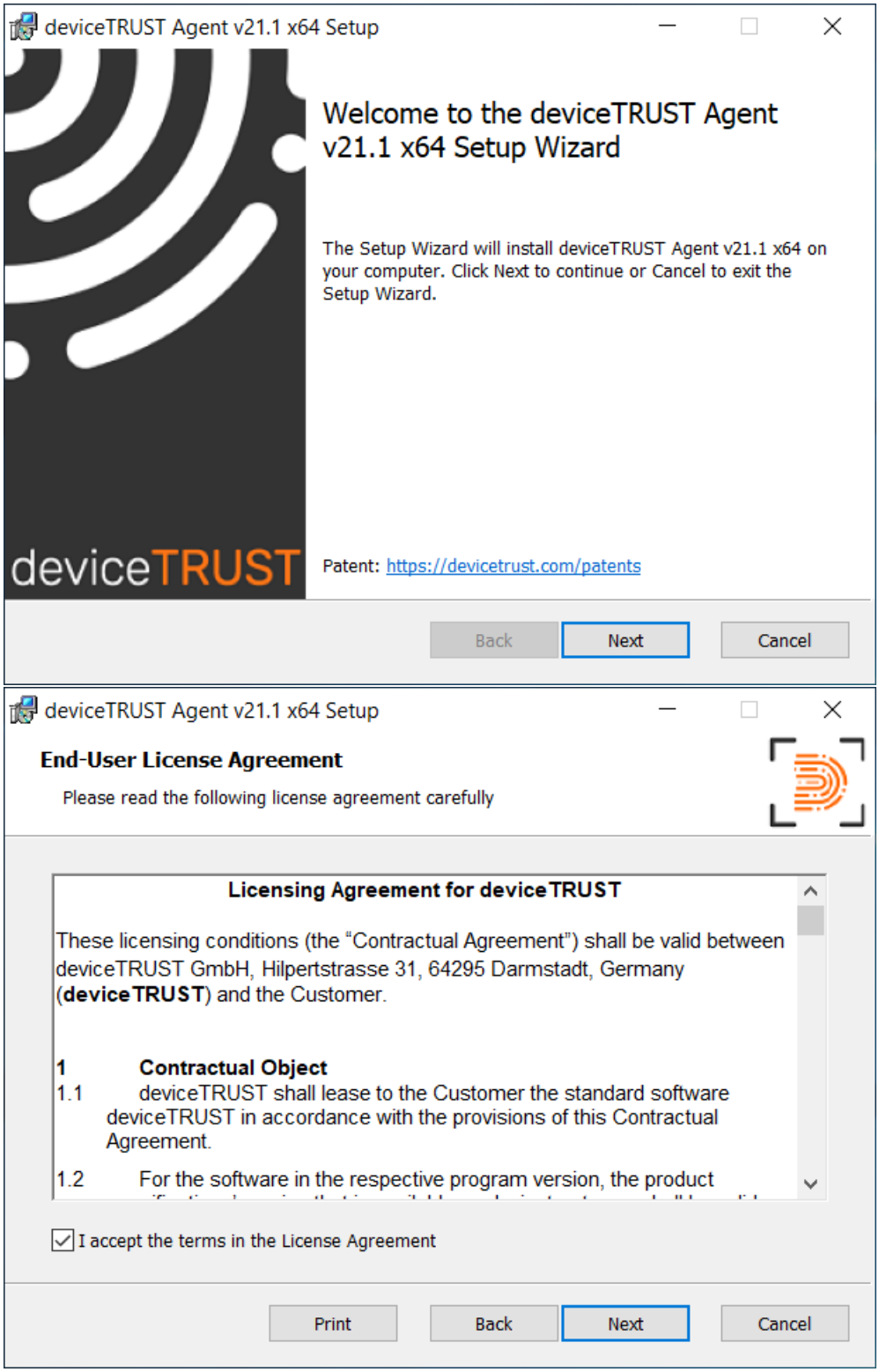
> **Note**
>
> ```
> deviceTRUST Agent, Console and Client Extension components
>    require administrative privileges for the installation.
> ```

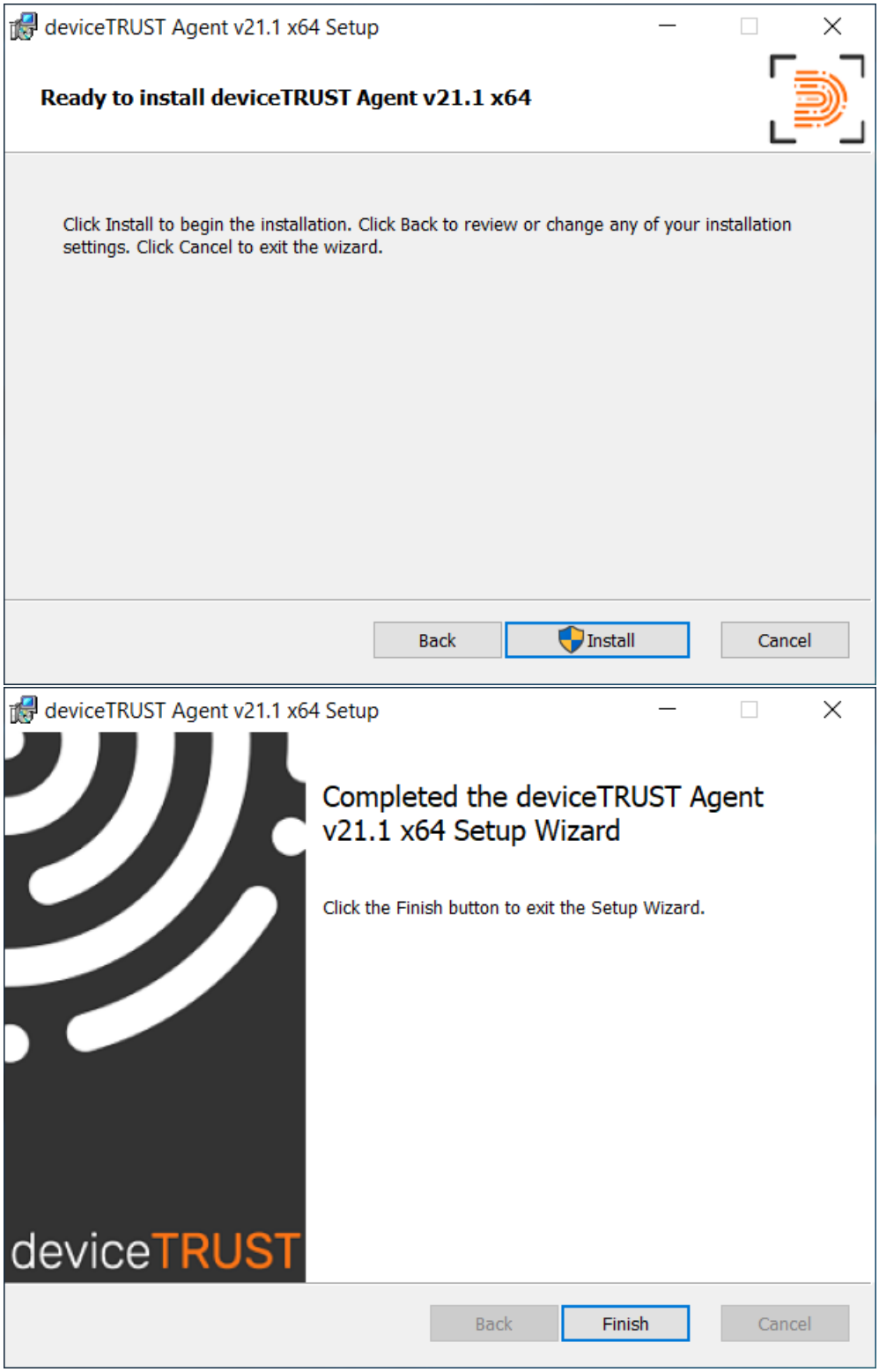
Installation

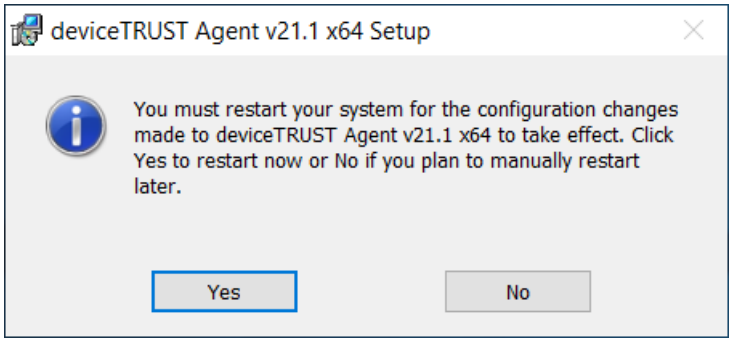
TABLE OF CONTENTS

Installing the deviceTRUST® Agent

The deviceTRUST Agent requires a user account with local administrative privileges to install the deviceTRUST Agent on the target system. The installation can be performed by following the steps of the deviceTRUST Agent installer.







Note:

- Installation path: `%PROGRAMFILES%\DEVICETRUST\AGENT`
- If the installation of the deviceTRUST Agent has finished successfully, a reboot is required to enable deviceTRUST to get system notifications to act on.
- If the `*Remote Desktop Services*` server role is added after installing the deviceTRUST Agent, the deviceTRUST Agent will need to be reinstalled.
- The deviceTRUST Agent will not function until a valid license is applied.

Citrix Virtual Channel Security

When using Citrix Virtual Apps and Desktops, you may need to edit the `Virtual channel allow list` policy to allow the deviceTRUST Agent to open a virtual channel to the deviceTRUST Client. More details can be found on the Knowledge Base.

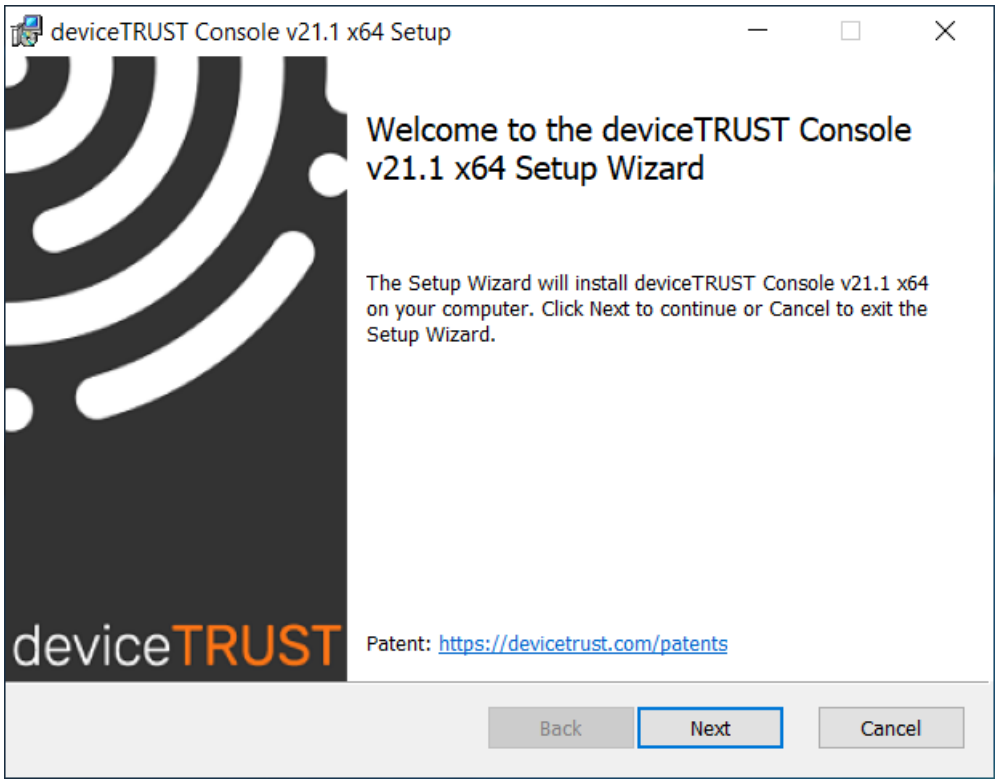
Unattended Installation

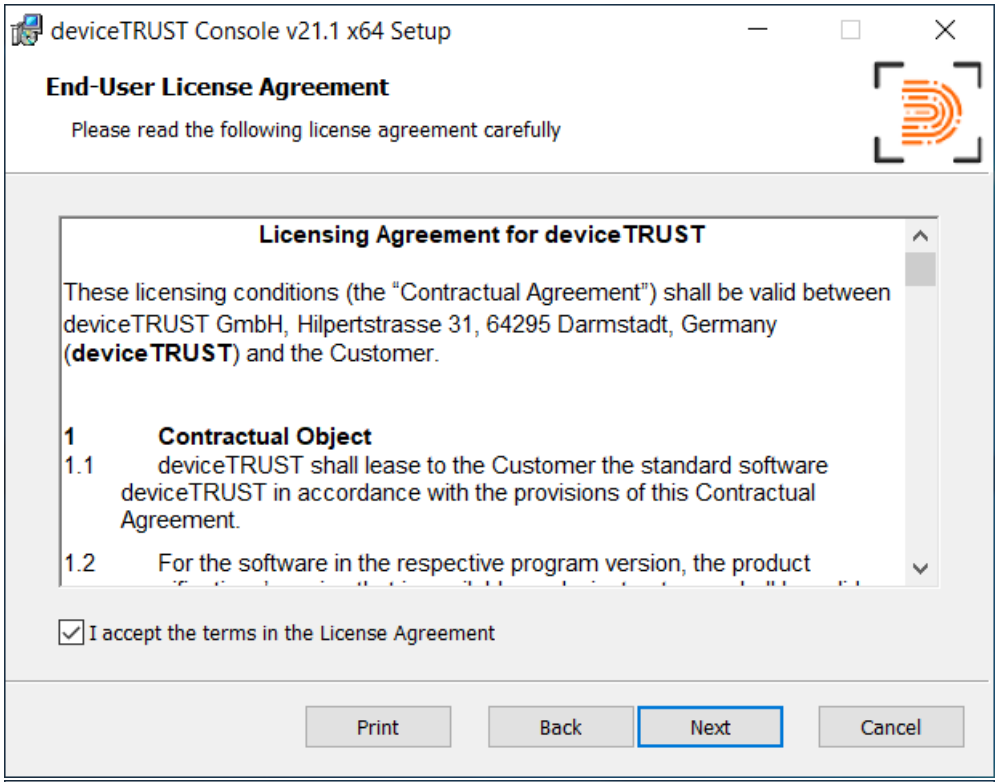
The deviceTRUST Agent can be installed unattended from the command line interface with the following options:

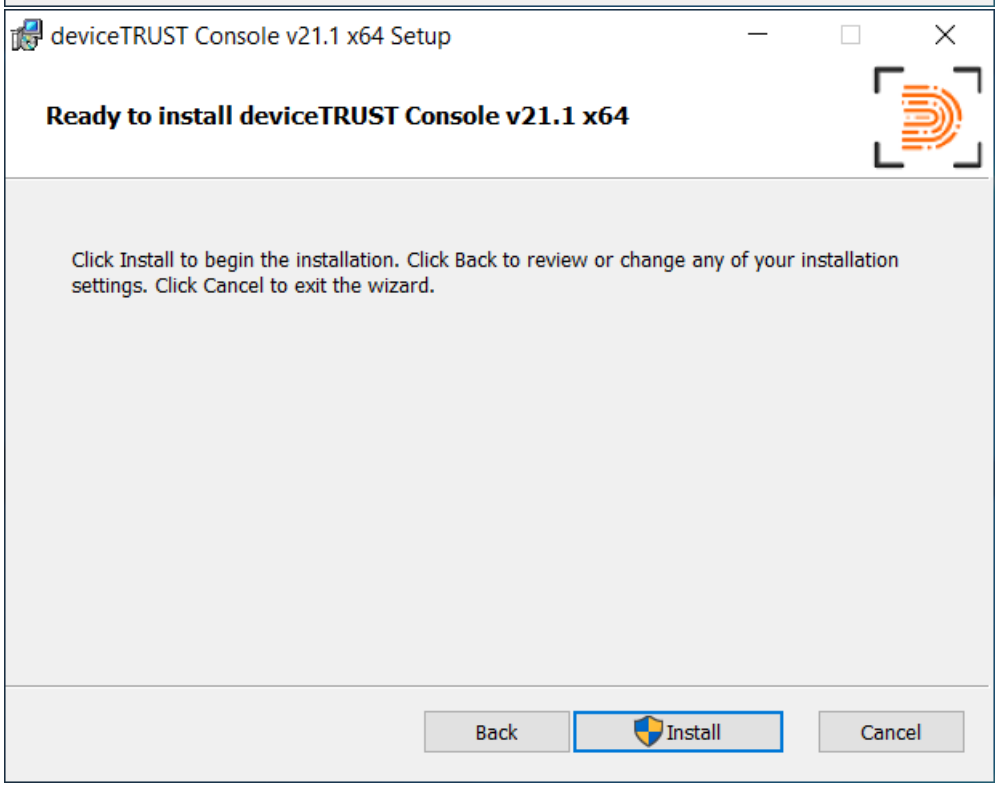
Component	Commandline
dtagent-x64-release-x.x.x.x.msi	The deviceTRUST 64-bit Agent installer file can be customized by common Microsoft Windows Installer parameters. An unattended installation can be achieved with the following parameters `MSIEXEC.EXE /I DTAGENT-X64-RELEASE-X.X.X.X.MSI /PASSIVE /FORCERESTART`
dtagent-x86-release-x.x.x.x.msi	The deviceTRUST 32-bit Agent installer file can be customized by common Microsoft Windows Installer parameters. An unattended installation can be achieved with the following parameters `MSIEXEC.EXE /I DTAGENT-X86-RELEASE-X.X.X.X.MSI /PASSIVE /FORCERESTART`

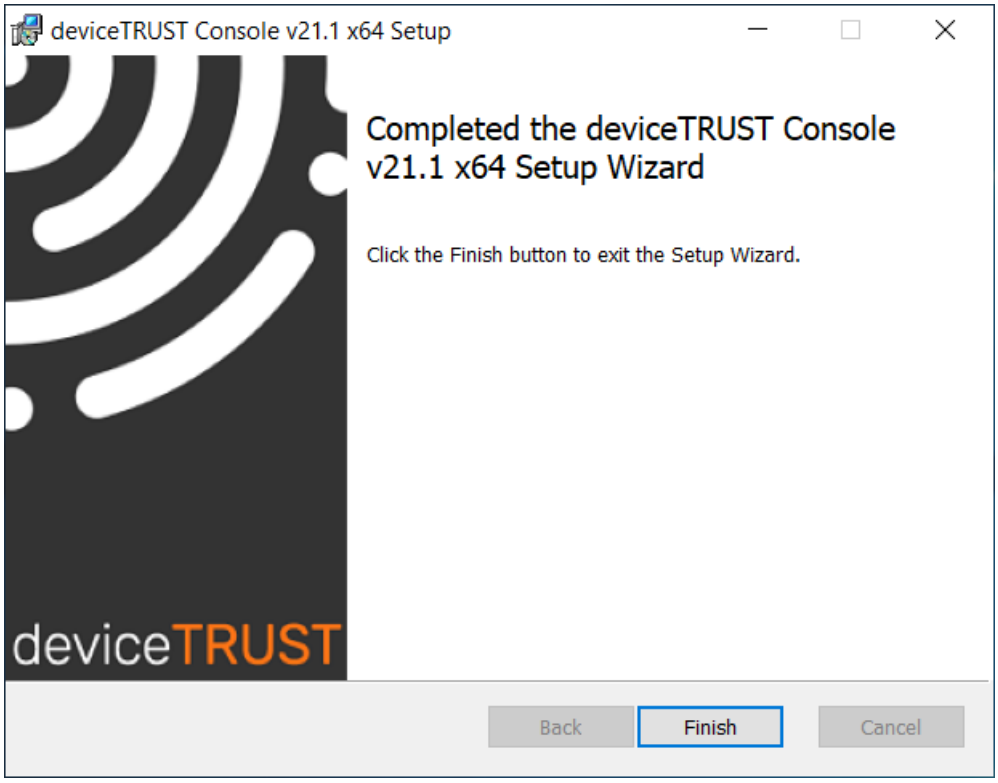
Installing the deviceTRUST® Console

The deviceTRUST Console requires a user account with local administrative privileges to install the deviceTRUST Console on the targeting system. The installation can be performed by following the steps of the deviceTRUST Console installer.









> **Note**
>
> Installation path: `%PROGRAMFILES%\DEVICETRUST\CONSOLE`

Unattended Installation

The deviceTRUST Console can be installed unattended from the command line interface with the following options:

Component	Commandline
dtconsole-x64-release-x.x.x.x.msi	The deviceTRUST 64-bit Console installer file can be customized by common Microsoft Windows Installer parameters. An unattended installation can be achieved with the following parameters `MSIEXEC.EXE /I DTCONSOLE-X64-RELEASE-X.X.X.X.MSI /PASSIVE`
dtconsole-x86-release-x.x.x.x.msi	The deviceTRUST 32-bit Console installer file can be customized by common Microsoft Windows Installer parameters. An unattended installation can be achieved with the following parameters `MSIEXEC.EXE /I DTCONSOLE-X86-RELEASE-X.X.X.X.MSI /PASSIVE`

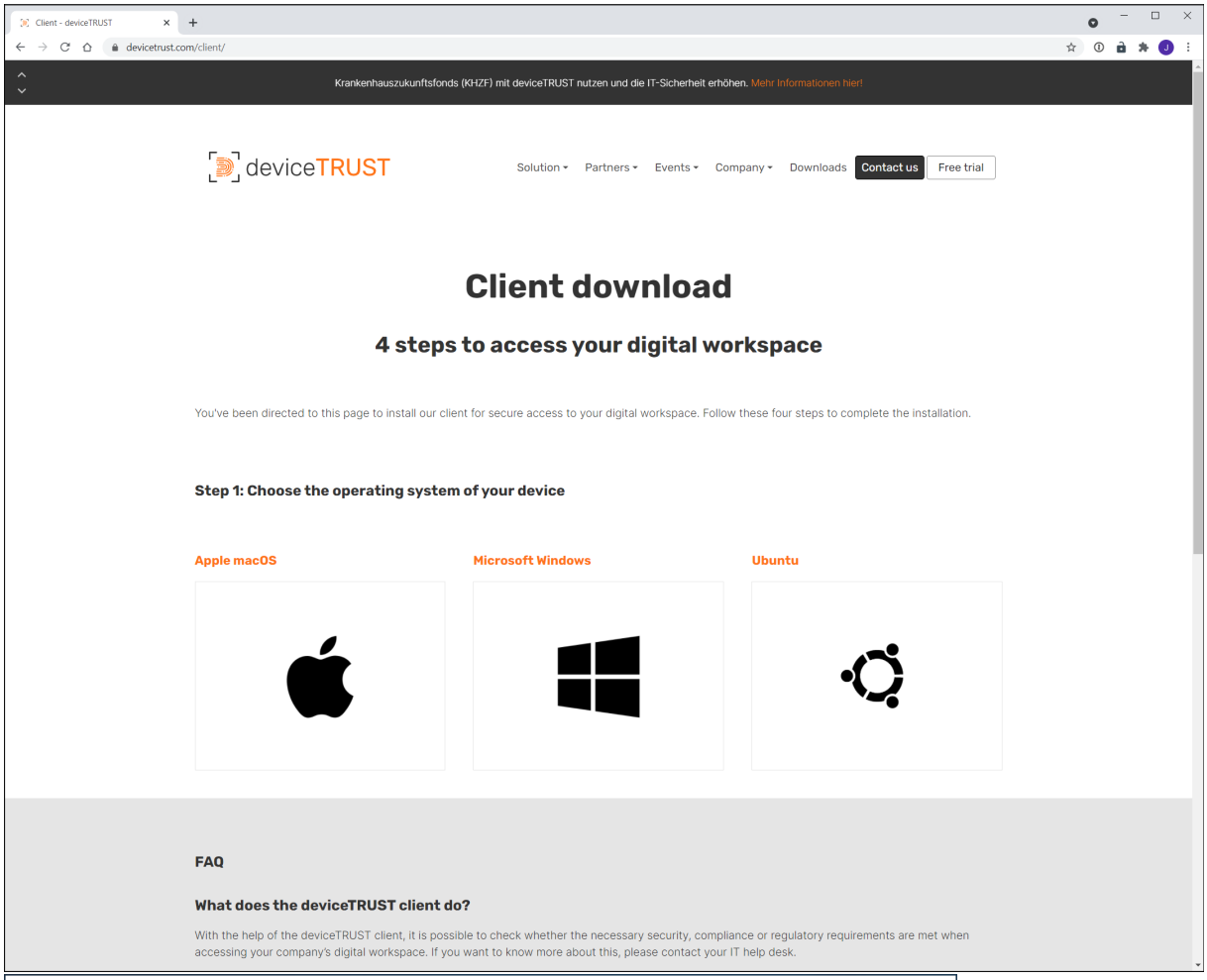
Installing the deviceTRUST® Client Extension

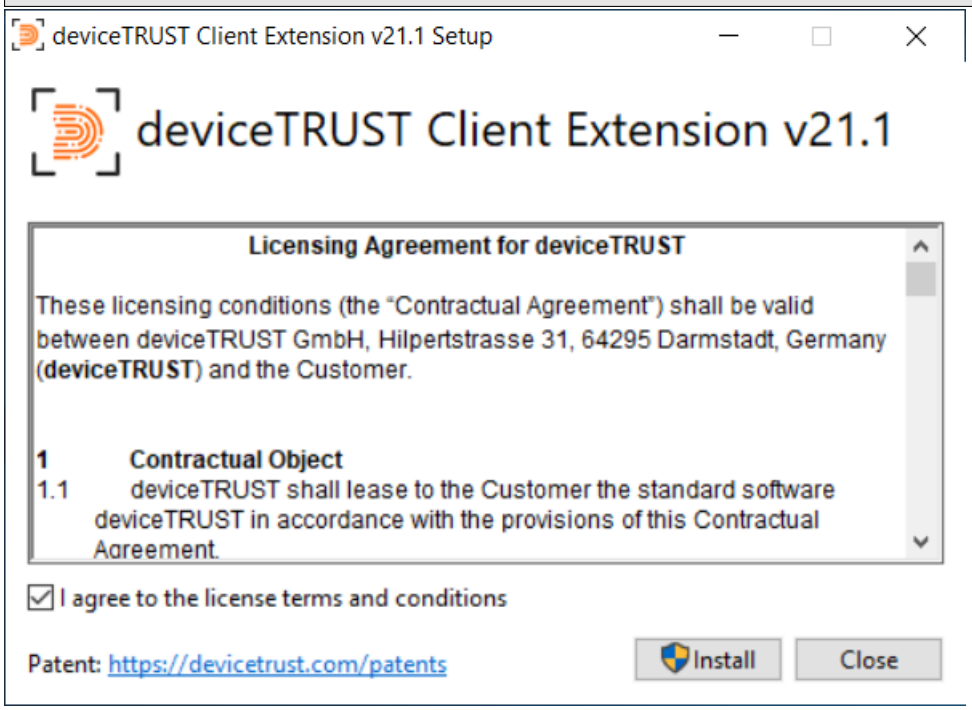
TABLE OF CONTENTS

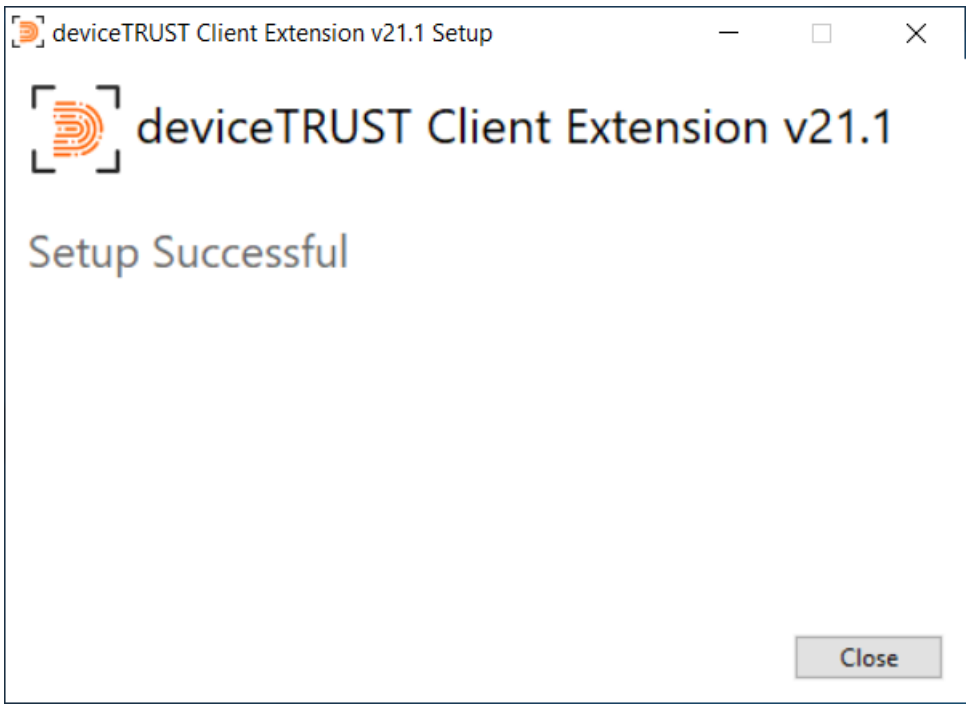
Client Extension installation on Microsoft Windows devices

The deviceTRUST Client Extension can be downloaded by following the steps on the deviceTRUST Client Extension Download page. The installation of the deviceTRUST Client Extension requires one-time administrative privileges of the user installing the deviceTRUST Client Extension.

The deviceTRUST® Client Extension for unmanaged Microsoft Windows devices supports an auto-update functionality. If a newer deviceTRUST Client Extension is available, the administrator of the remoting and DaaS environment can enforce a transparent update installation of the local deviceTRUST Client Extension.







> **Important**
>
> - When using Microsoft RDP or VMware Horizon, when installing with custom credentials the user must logoff from their session before the installation is effective.

Client Extension Installation Command Line Options

The following command line options are available to the deviceTRUST Client Extension installer:

- `DTCLIENT-EXTENSION-RELEASE-X.X.X.EXE /INSTALL|REPAIR|UNINSTALL` - Installs, repairs or uninstalls the deviceTRUST Client Extension.
- `DTCLIENT-EXTENSION-RELEASE-X.X.X.EXE /PASSIVE|QUIET` - Passive displays a minimal user interface with no prompts for input. Quiet displays no user interface.
- `DTCLIENT-EXTENSION-RELEASE-X.X.X.EXE /NORESTART` - Surpresses any attempt to restart the local device.
- `DTCLIENT-EXTENSION-RELEASE-X.X.X.EXE /LOG <log.txt>` - Logs to a specific file. By default, logs are created within %TEMP%.

> **Note:**
>
> - One or more of the above options can be combined.
> - Installation path: %PROGRAMFILES%\DEVICETRUST\CLIENT
> - The deviceTRUST Client Extension installer installs both the 32-bit and 64-bit deviceTRUST Client Extension components depending on the Windows operating system platform.

Unattended Installation

The deviceTRUST Client Extension installer can be installed unattended from the command line interface with the following options:

Component	Commandline
dtclient-extension-release-x.x.x.exe	An unattended installation can be achieved with the following parameters `DTCLIENT-RELEASE-X.X.X.EXE /INSTALL /QUIET`

Active Setup Components

The deviceTRUST Client Extension uses Active Setup to make changes to the users profile. This involves the creation of the following Active Setup components.

deviceTRUST Channel Registration 64-bit component

Registry Key: `HKLM\Software\Microsoft\Active Setup\Installed Components\\{ 2f780db6-903f-4c83-9d60-9d138e054dc9 }`

This Active Setup component is created during installation, and is responsible for the registration of the 64-bit Microsoft RDP Virtual Channel within the users profile. When not using the 64-bit Microsoft RDP Virtual Channel, it is safe to remove this key after the deviceTRUST Client Extension has been installed.

deviceTRUST Channel Registration 32-bit component

Registry Key: `HKLM\Software\Wow6432Node\Microsoft\Active Setup\Installed Components\\{ 8a4445d6-867e-4e60-8cc5-05bc2e3a2512 }` (or `HKLM\Software\Microsoft\Active Setup\Installed Components\\{ 8a4445d6-867e-4e60-8cc5-05bc2e3a2512 }` on a 32-bit OS)

This Active Setup component is created during installation, and is responsible for:

- The registration of the 32-bit Microsoft RDP Virtual Channel.
- The registration of the 32-bit Citrix ICA® Virtual Channel with per-user installations of Citrix Workspace™ within the users profile.
- The registration of the 32-bit Amazon WorkSpaces Virtual Channel with per-user installations of Amazon WorkSpace within the users profile.
  When not using 32-bit Microsoft RDP or per-user installations of Citrix Workspace App or Amazon

WorkSpaces Client, it is safe to remove this key after the deviceTRUST Client Extension has been installed.

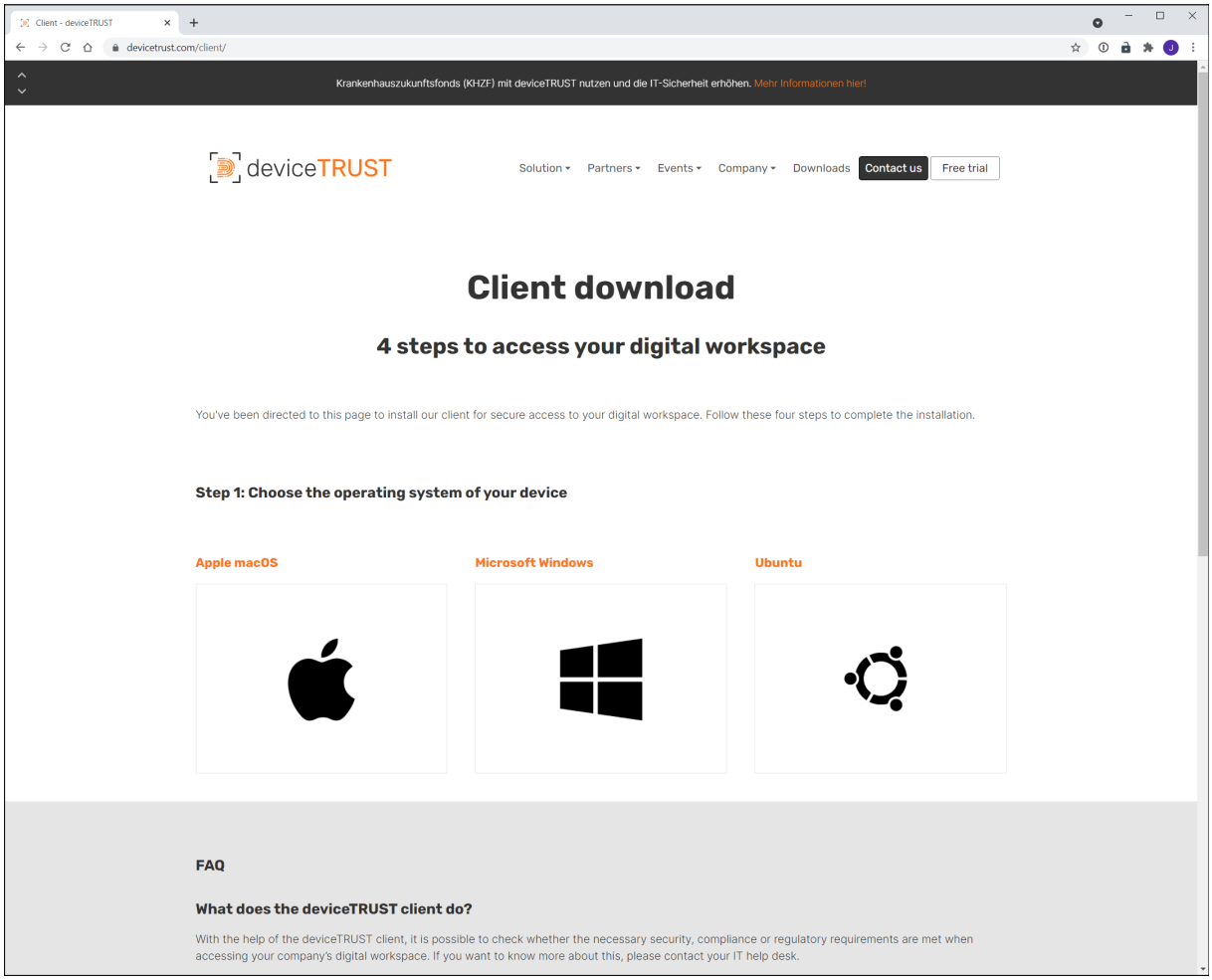
***deviceTRUST Citrix Cleanup 32-bit* component**

Registry Key: `HKLM\Software\Wow6432Node\Microsoft\Active Setup\Installed Components\\{ 3d65bd68-a307-4d64-af9c-17bc9c4b36e3 }`

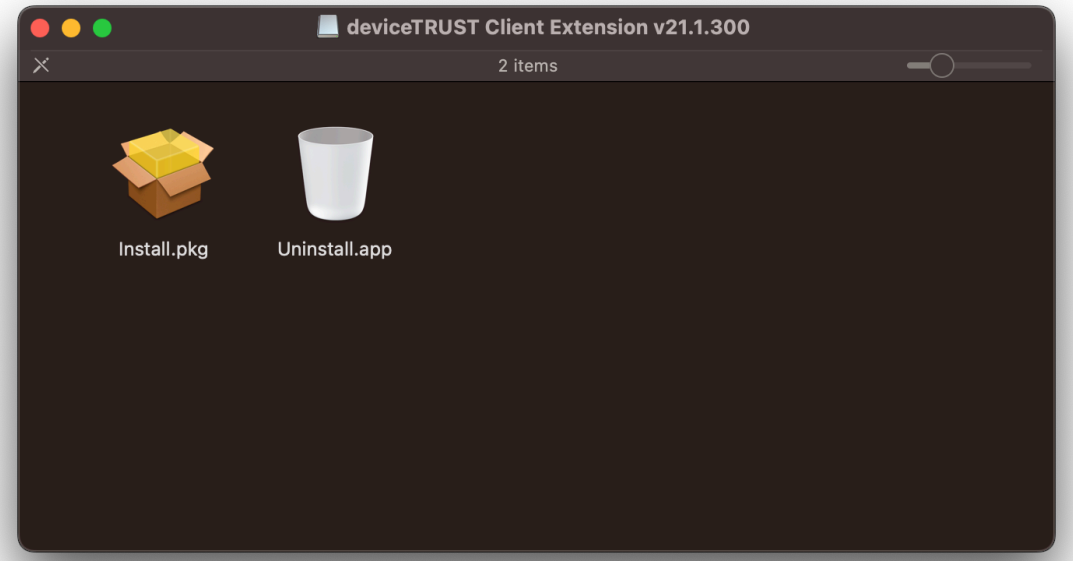
This Active Setup component is created during uninstallation and is responsible for removing the registration of the 32-bit Citrix ICA Virtual Channel with per-user installations of Citrix Workspace within the users profile. When not using per-user installations of Citrix Workspace, it is safe to remove this key after the deviceTRUST Client Extension has been uninstalled.

Client installation on Apple macOS devices

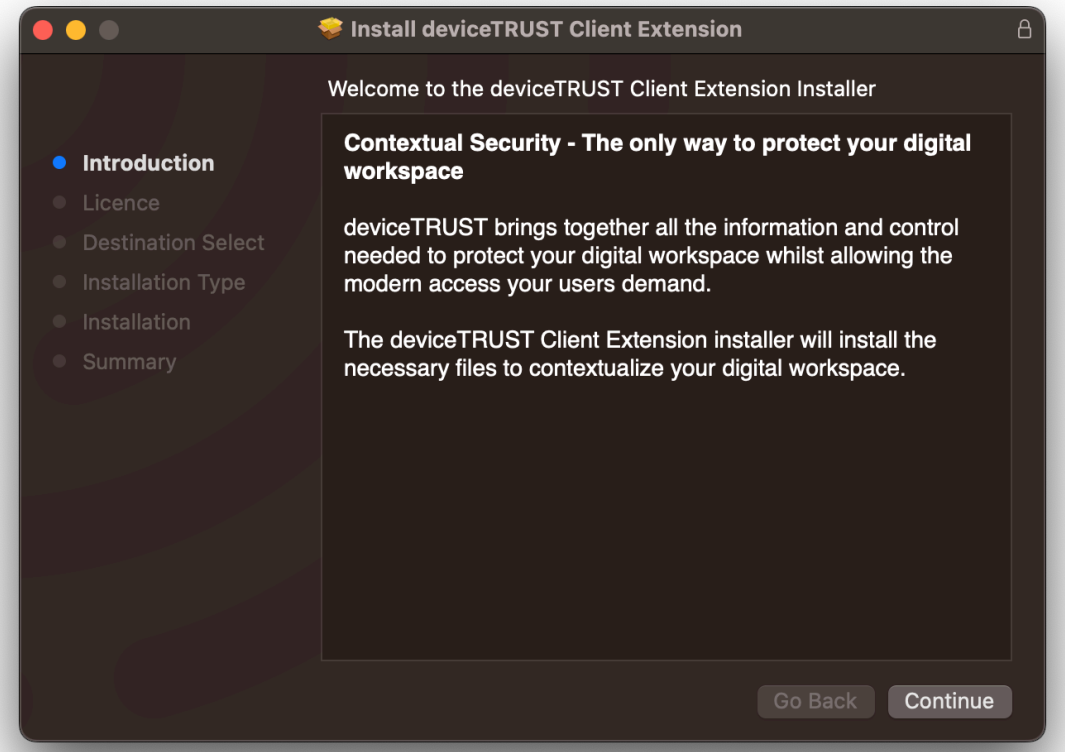
The deviceTRUST macOS Client can be downloaded by following the steps on the deviceTRUST Client Extension Download page.



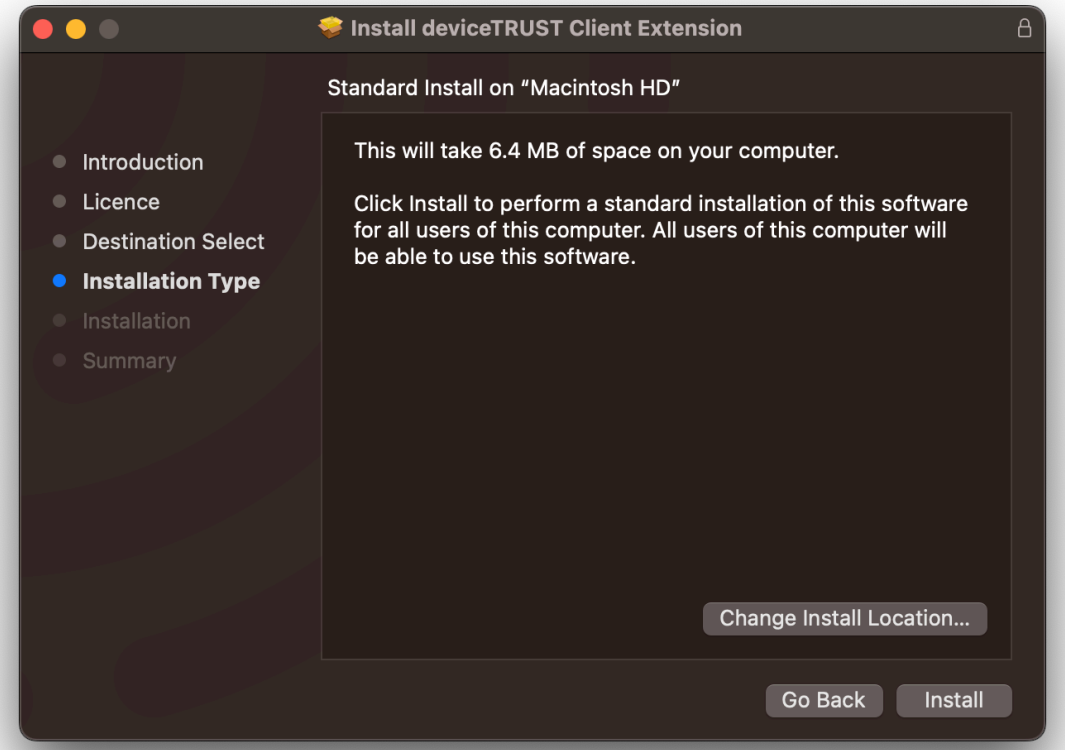
To access the installer, double click on the downloaded ‘dtclient-extension-macos-universal-release-W.X.Y.Z.dmg’file.

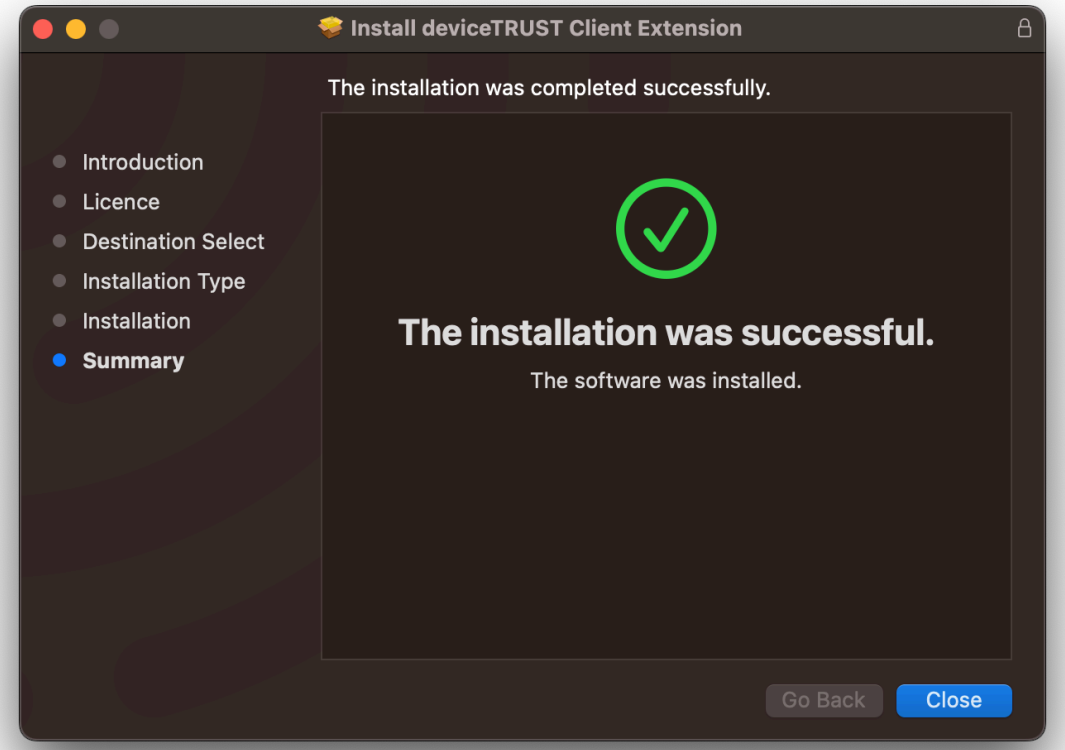


Launch the installer by double clicking on othe ‘Install.pkg’file.



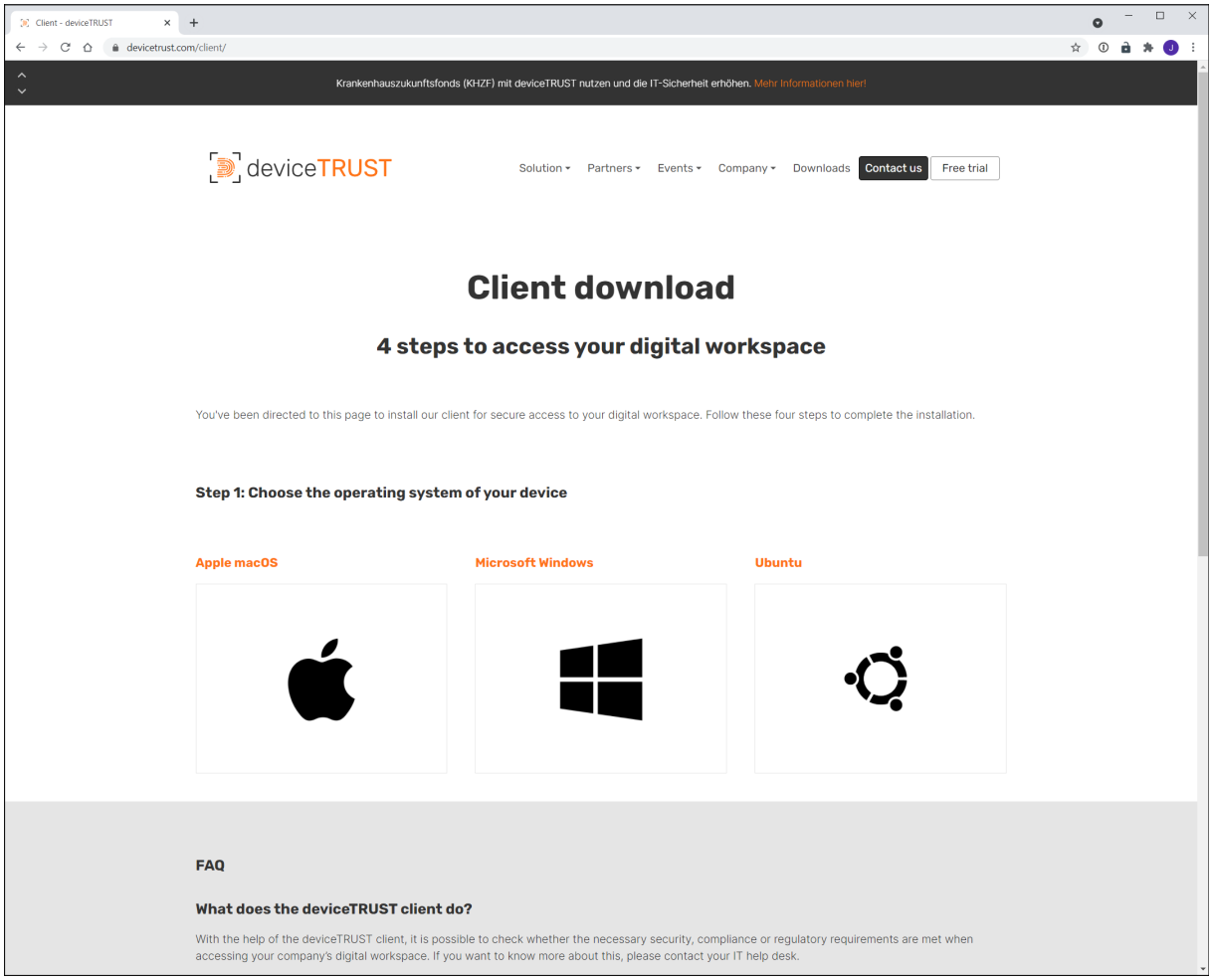
Continue through the installation steps until complete.



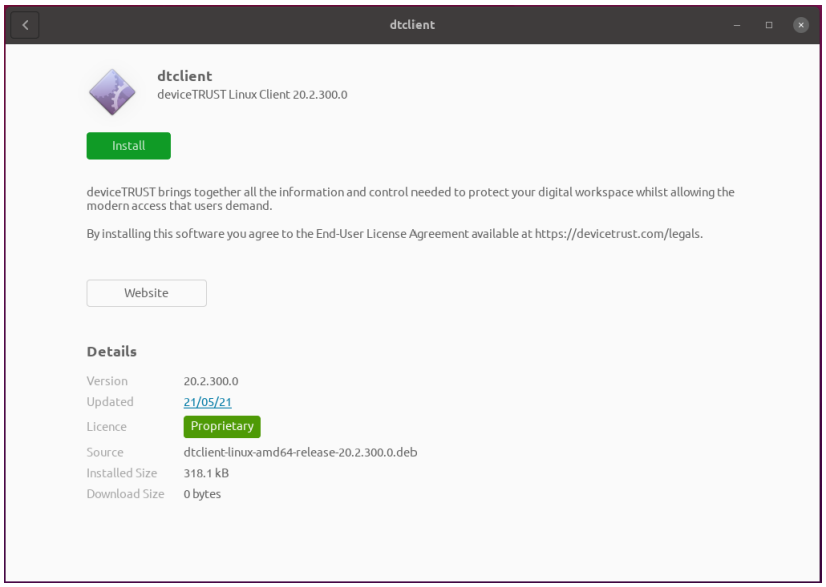


Client installation on Ubuntu devices

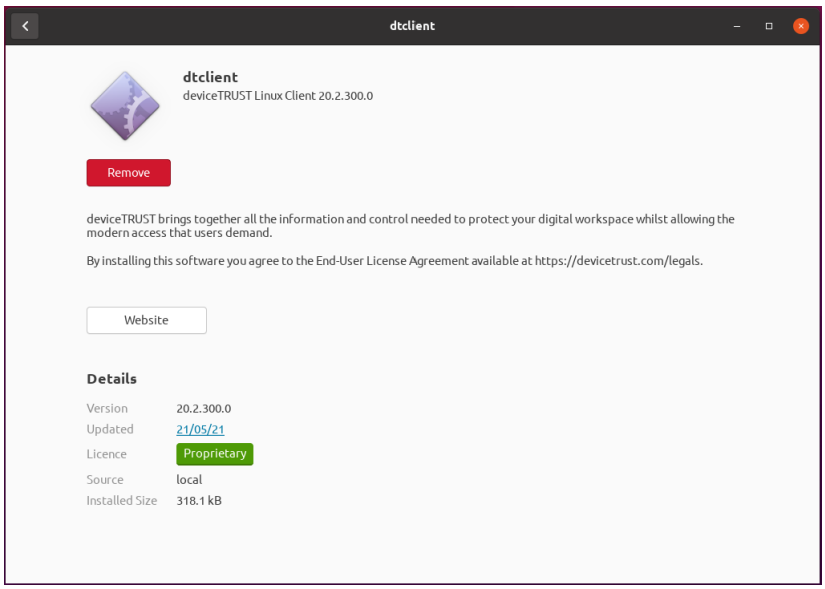
The deviceTRUST Ubuntu Client can be downloaded by following the steps on the deviceTRUST Client Extension Download page.



To access the installer, double click on the downloaded 'dtclient-linux-amd64-release-W.X.Y.Z.deb' file.



Click *Install* to complete the installation. An *Authentication Required* popup will be displayed and administrative credentials must be entered.



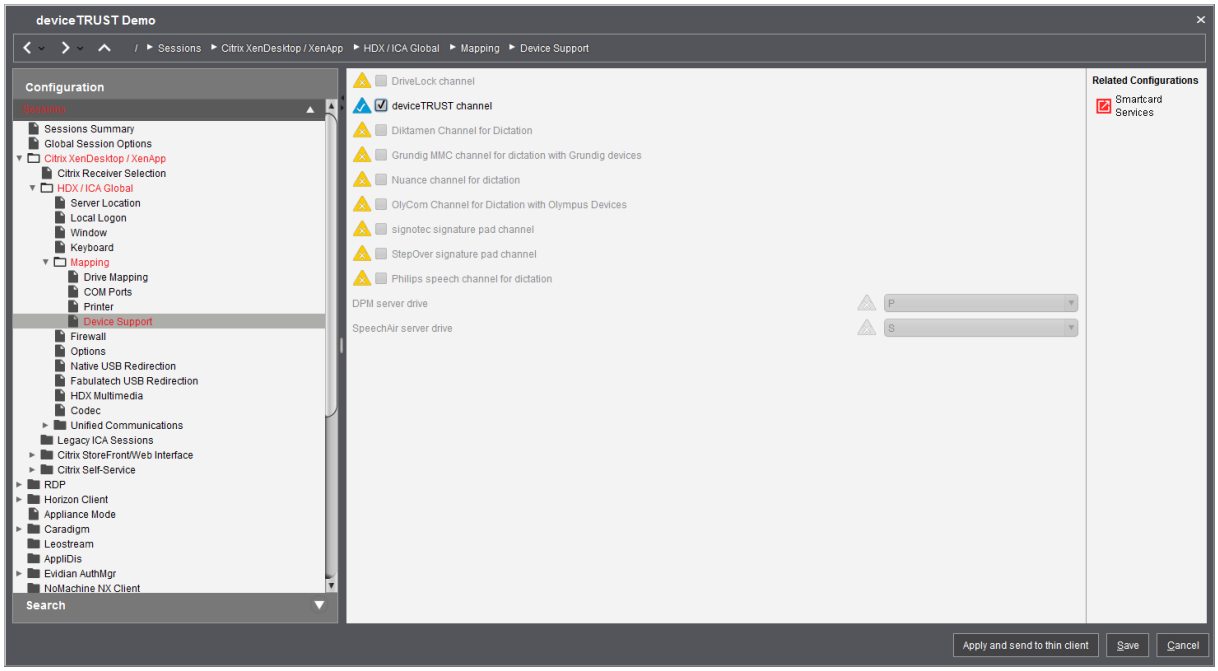
Client installation on IGEL OS devices

deviceTRUST is already integrated within the IGEL OS release 10.03.500 or higher and supports the Microsoft Remote Desktop Protocol (RDP) and the Citrix Independent Computing Architecture (ICA). Within the default IGEL profile, deviceTRUST is not enabled and needs to be enabled for both remoting protocols independently.

> **Note:**
>
> - To enable deviceTRUST® within the IGEL OS you can use the IGEL Universal Management Suite (UMS) or the local administrative interface.
> - IGEL devices must be restarted after enabling deviceTRUST.

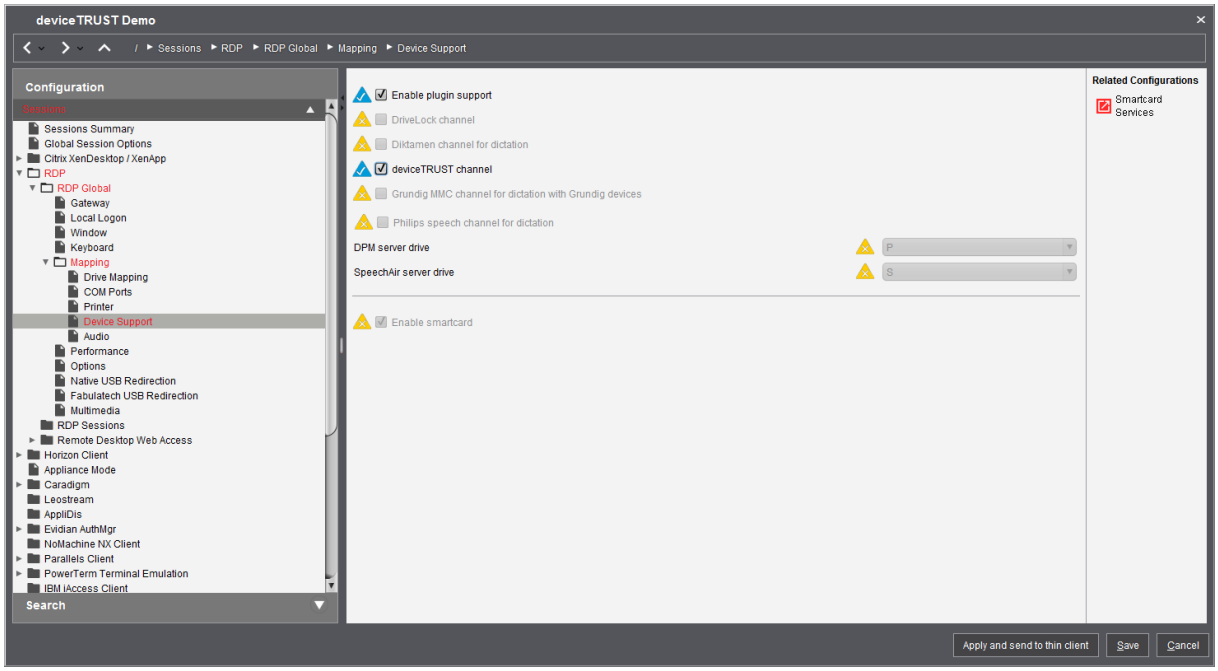
Enable deviceTRUST for Citrix Independent Computing Architecture (ICA®)

Open the target profile (local or through the UMS) and navigate to SESSIONS\CITRIX XENDESKTOP / XENAPP\HDX / ICA GLOBAL\MAPPING and activate the DEVICETRUST CHANNEL option.



Enable deviceTRUST for Microsoft Remote Desktop Protocol (RDP)

Open the target profile (local or through the UMS) and navigate to SESSIONS\RDP\RDP GLOBAL\MAPPING and activate the DEVICETRUST CHANNEL option.



Client installation on Unicon™ eLux devices

Import the deviceTRUST® certificate chain

The deviceTRUST eLux Client is signed with a certificate that must first be trusted by both ELIAS and the eLux devices. Full details of how to import certificates into ELIAS and deploy them to eLux devices can be found by searching for 'Organizing certificates for package validation' within the eLux documentation. The latest deviceTRUST certificate chain can be downloaded Download certificate chain.

> **Note**
>
> The import of the deviceTRUST certificate chain is only necessary when the eLux® environment is configured to check the signatures of packages. However, it is good practice to ensure that the packages match the certificates.

Within ELIAS:

- Import `devicetrust-gmbh-20210321.pem` into `Trusted Issuer`.
- Import `globalsign-extended-validation-codesigning-ca-sha256-g3-20240615.pem` and `globalsign-intermediate-20280128.pem` into `Intermediate CA`.
- Import `globalsign-20280128.pem` into `Trusted Root CA`.

Within the Scout console:

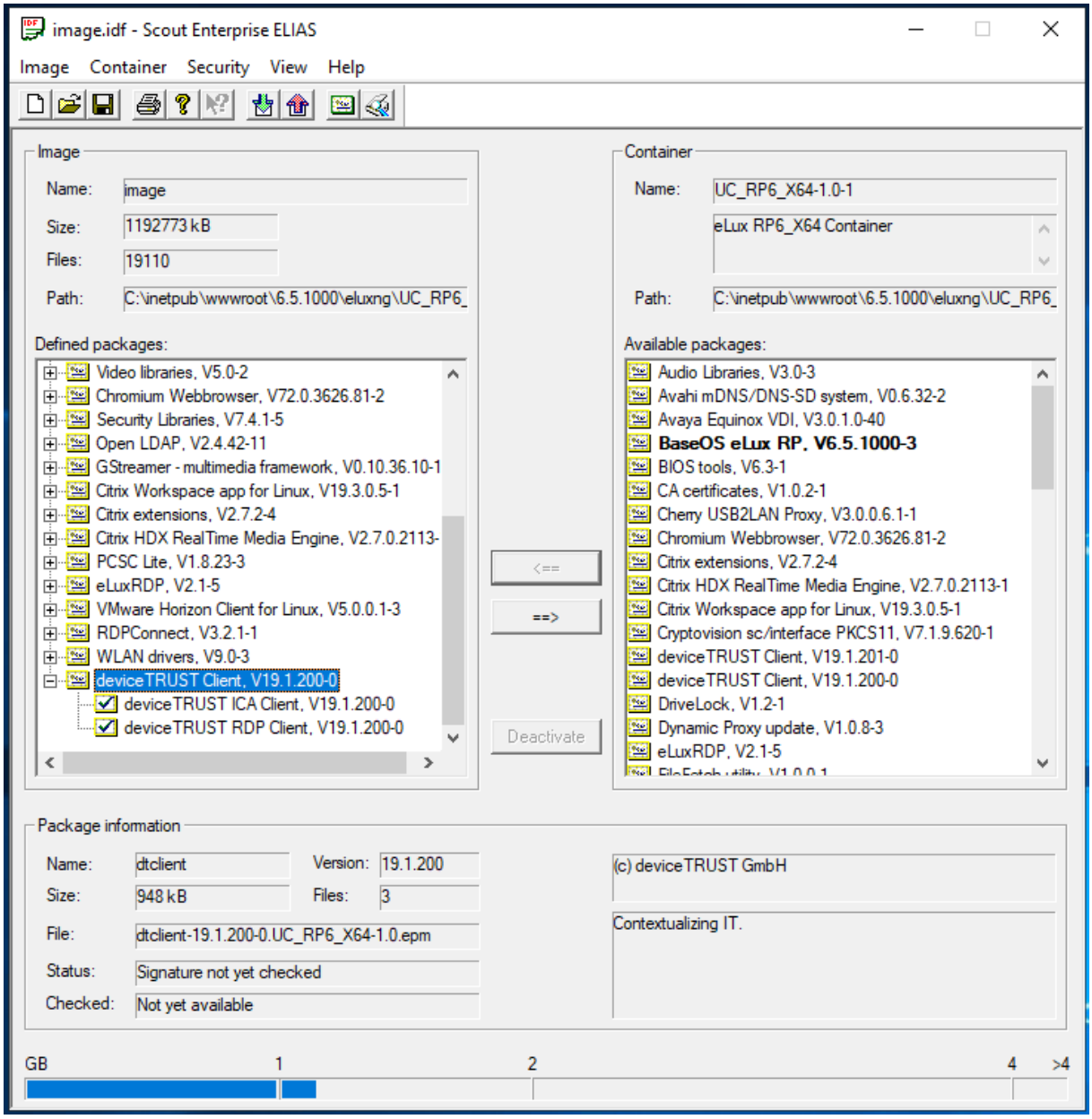
- Deploy all of the above `pem` files into the `/setup/cacerts/` folder of the eLux device.

Install the package

Within ELIAS:

- Ensure that the required 'deviceTRUST Client' package has been imported into the package library.
- If you've imported the deviceTRUST certificate chain, then check the signatures of the package.
- Assign the 'deviceTRUST Client' package to the image.

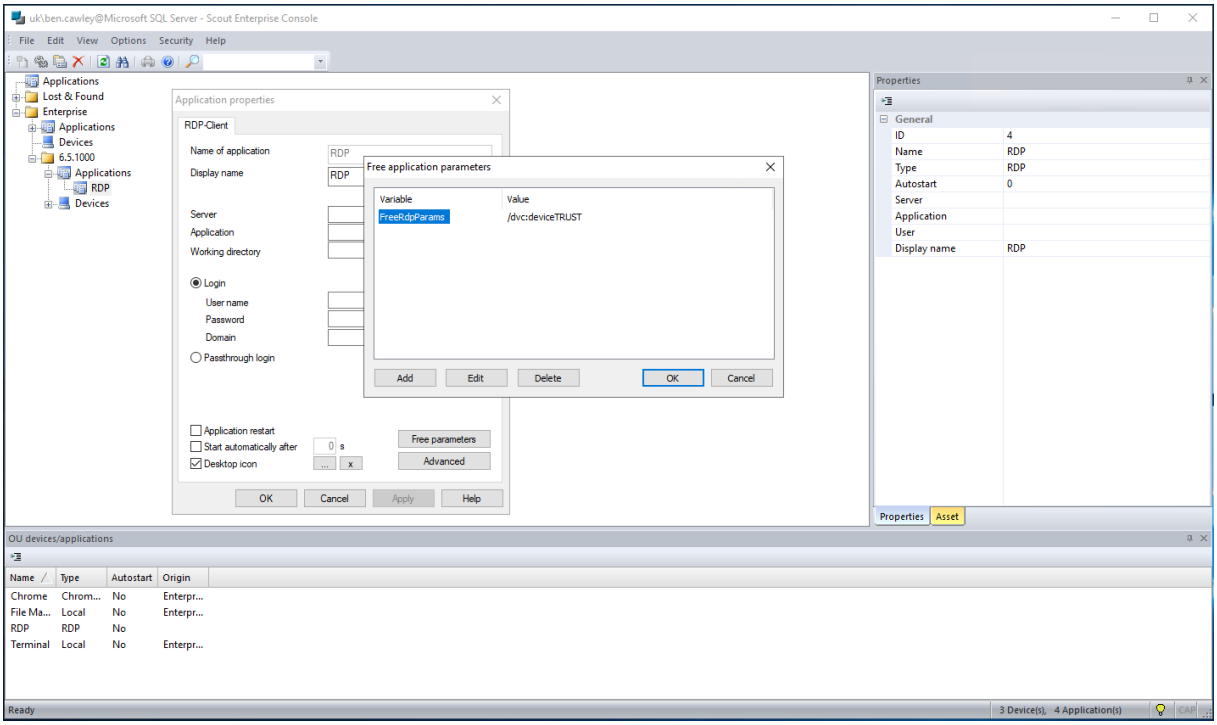
For ICA support, ensure the 'deviceTRUST ICA Client, Vx.x.x-x' option is active under the defined package 'deviceTRUST Client, Vx.x.x-x'.
For RDP support, ensure the 'deviceTRUST RDP Client, Vx.x.x-x' option is active under the defined package 'deviceTRUST Client, Vx.x.x-x'.



Within the Scout console:

- Update your eLux devices.

To enable the RDP virtual channel, within your eLux Scout console navigate to Enterprise > [eLux version number] > Applications > RDP, and select properties, then ‘Free parameters’. Add variable ‘FreeRdpParams’with value ‘/dvc:deviceTRUST’.

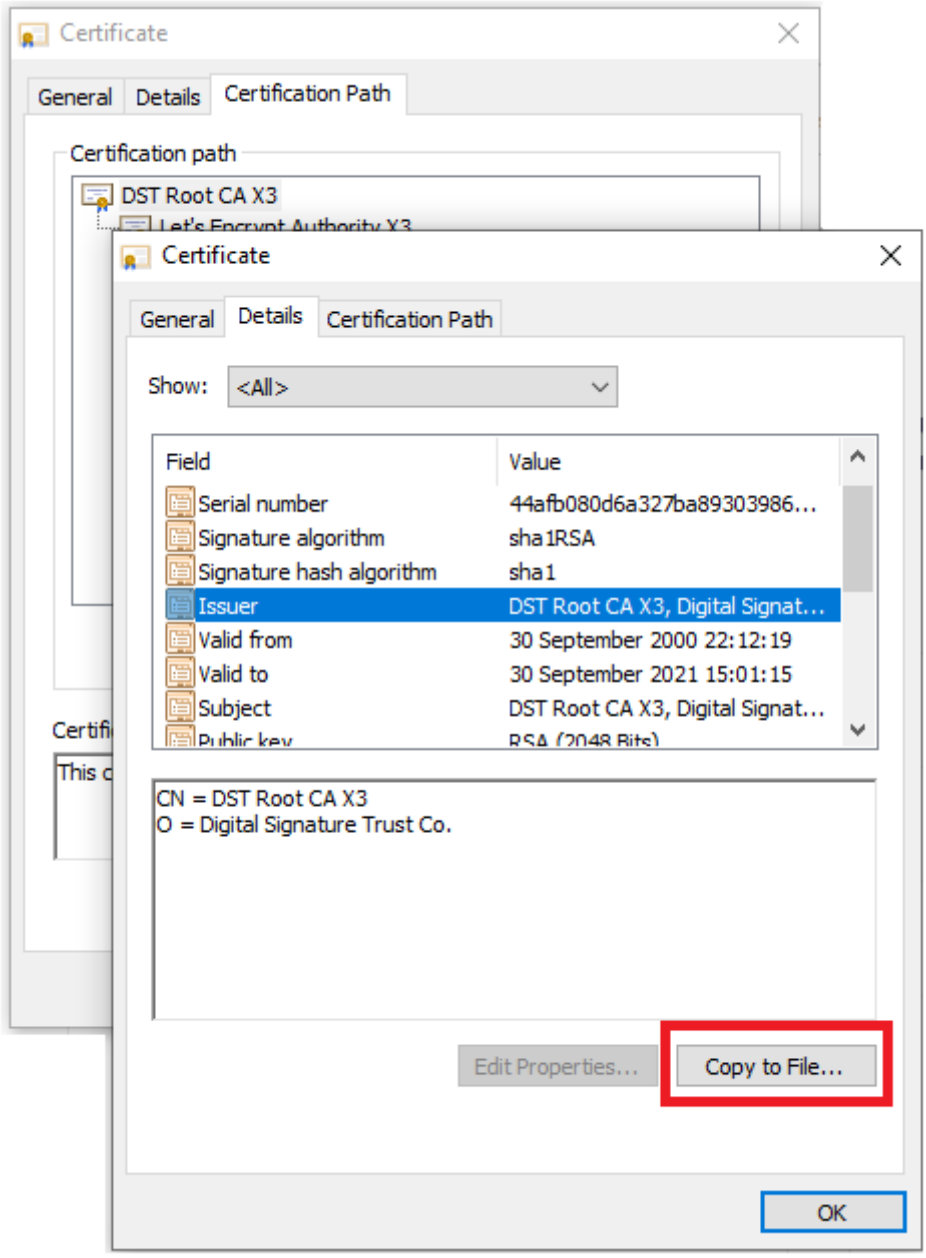


Next time you connect using ICA® or RDP to a server with a licensed deviceTRUST Host, the properties of your eLux device will be available within your virtual session.

Deploy the ripe.stat root certificate

To ensure the deviceTRUST eLux client can successfully obtain whois information you will need to ensure that the required root certificate is deployed to all your eLux devices.

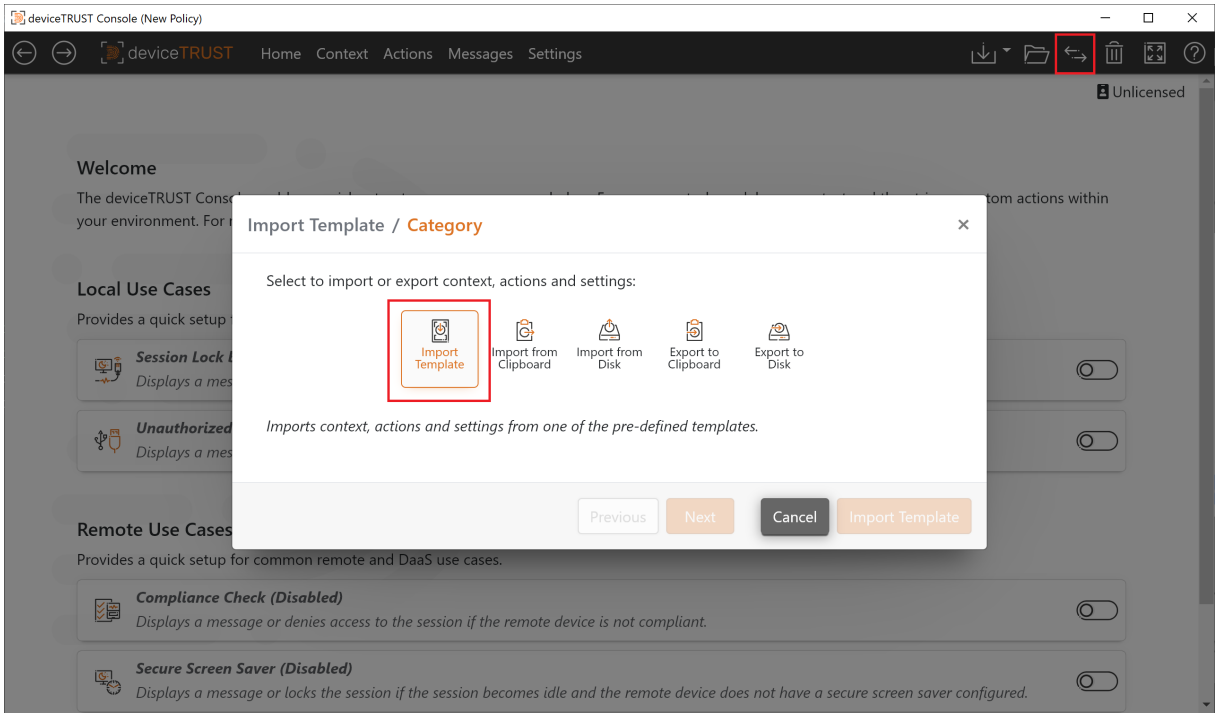
To obtain the root certificate, navigate to stat.ripe.net within your browser and save the certificate to disk as a base-64 encoded X. 509 file.



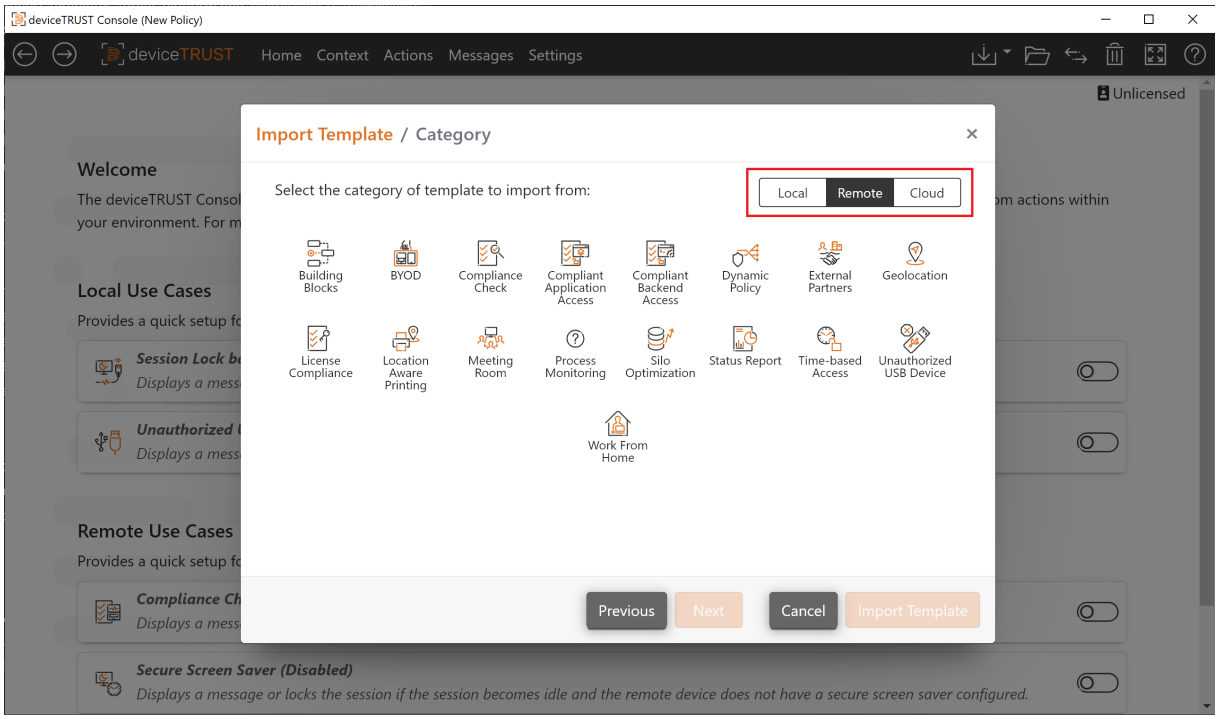
Within the Scout console select 'Advanced device configuration\Files'on the device container, import the certificate and deploy to your device certificate store location.

Templates

The deviceTRUST® Console includes a set of templates which can be used to quickly implement a use case. Launch the deviceTRUST Console and select `SHARING` in the top right of the navigation bar and then `IMPORT TEMPLATE`.



The deviceTRUST use cases are summarized in the following categories for each target platform. Use the filter to select the desired target platform.



Local Templates

The deviceTRUST® use cases for local devices are summarized within the following categories.

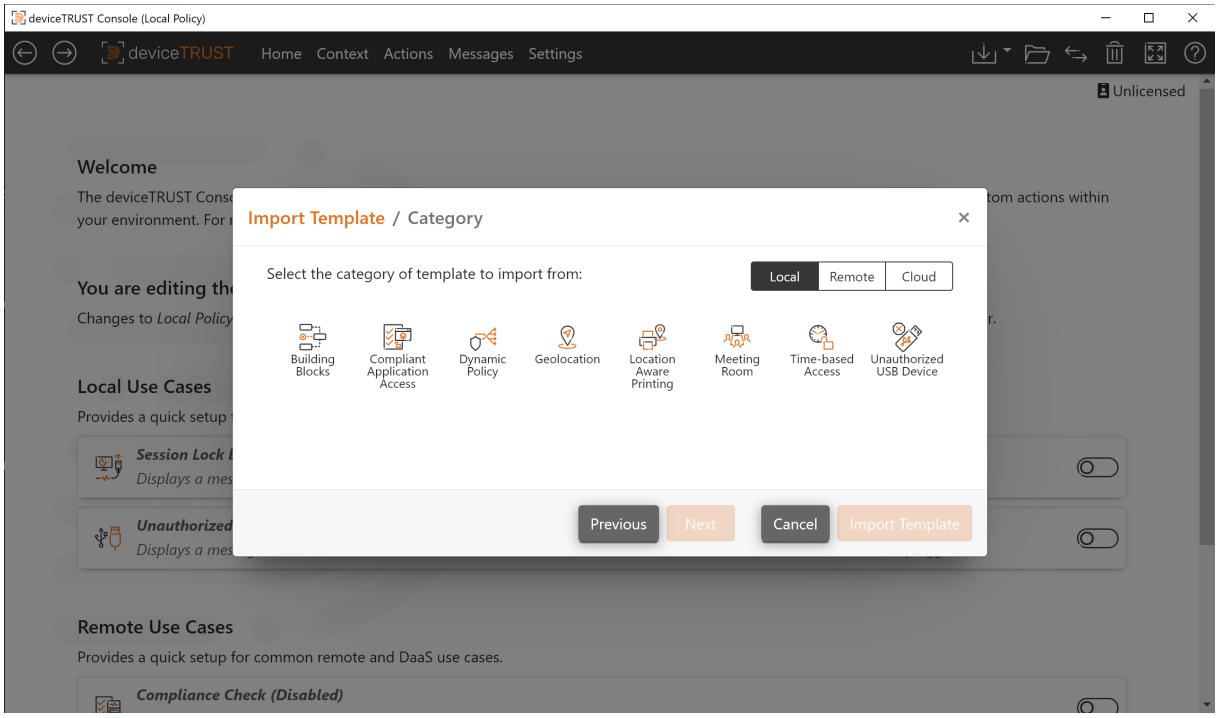


TABLE OF CONTENTS

Remote Templates

The deviceTRUST® use cases for remoting and DaaS are summarized within the following categories.

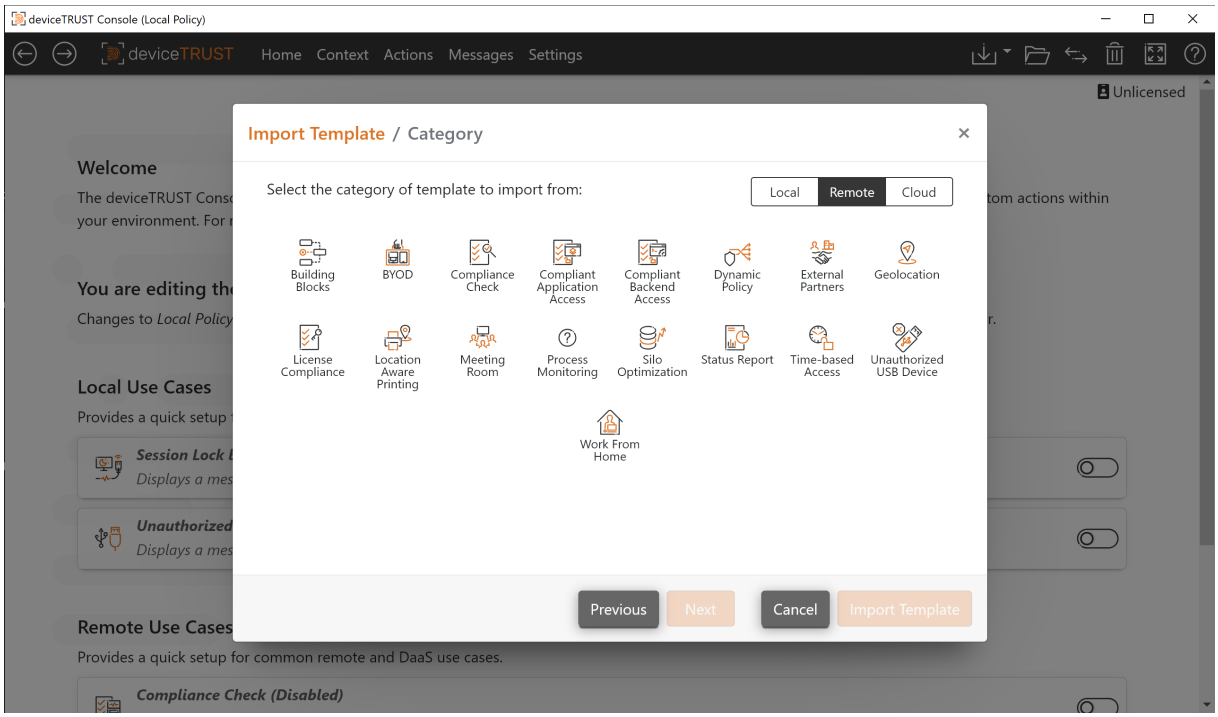


TABLE OF CONTENTS

- Building Blocks - Individual contexts and actions that can be used as building blocks within your configuration.
- BYOD (Bring Your Own Device) - Controls access to the session or applications when compliance requirements for BYOD users are not satisfied.
- Compliance Check - Display a message or denies access to the session when compliance requirements on a remote device are not satisfied.
- Compliant Application Access - Controls access to applications within the session.
- Compliant Backend Access - Display a message or denies access to backend servers when compliance requirements are not satisfied.
- Dynamic Policy - Applies a dynamic policy within the session.
- External Partners - Controls access to the session and applications when compliance requirements for external partners are not satisfied.
- Geolocation - Validates and controls access based on geolocation information of the remote device.
- License Compliance - Controls access to applications within the session if the remote device is

not licensed.

- Location Aware Printing - Maps network printers and defines a default printer based on the device placement within a building.
- Meeting Room - Controls an idle period for devices based upon whether the device is placed within a meeting room.
- Process Monitoring - Controls access to applications or the session based on running processes on the remote device.
- Silo Optimization - Reduces the number of silos by controlling application access for remote devices within a single silo.
- Status Report - Reports the status of the remote device to various destinations.
- Time-based Access - Controls access to the session or applications when accessed outside of working hours.
- Unauthorized USB Device - Denies access to the session when an unauthorized USB device is plugged in.
- Work From Home - Validates and controls access based on the remote device for home office users.

Cloud Templates

The deviceTRUST® use cases for cloud connected devices are summarized within the following categories.

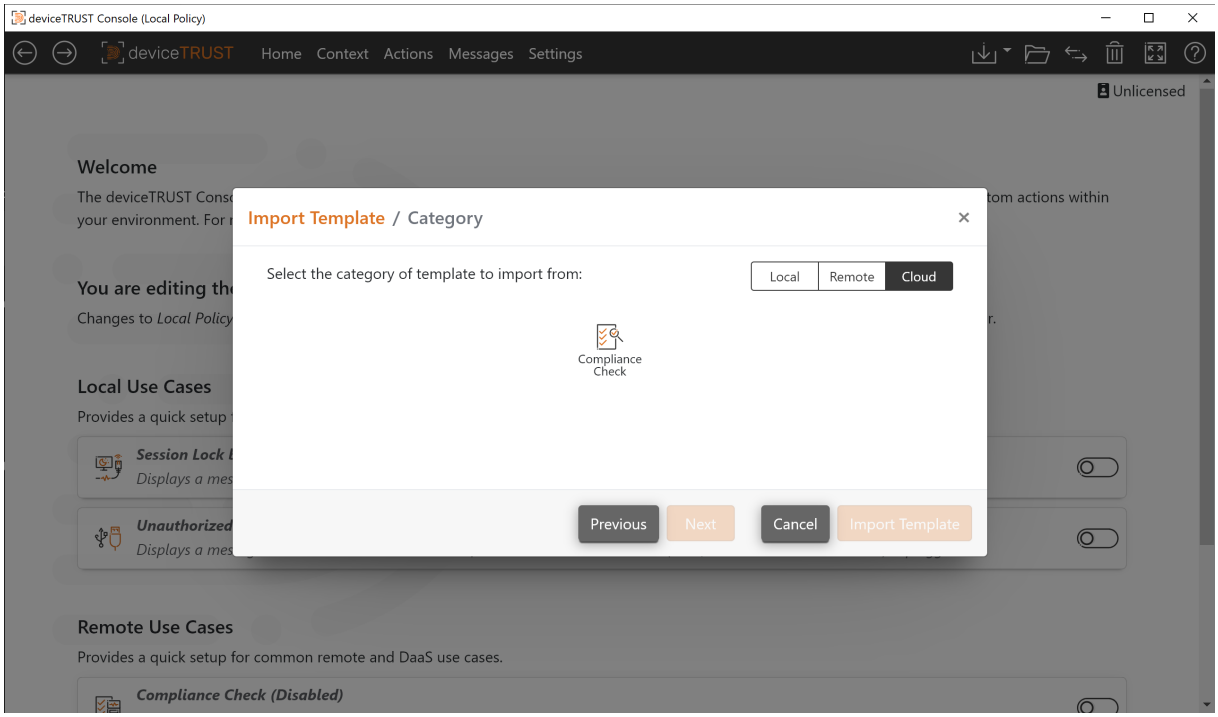


TABLE OF CONTENTS

Compliance Check - Display a message or denies access to the session when compliance requirements on a remote device are not satisfied.

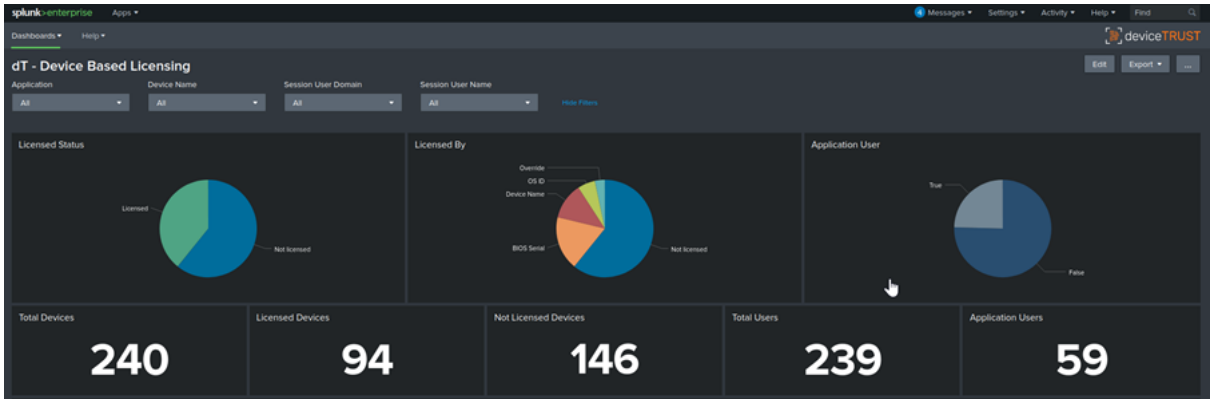
Reporting

TABLE OF CONTENTS

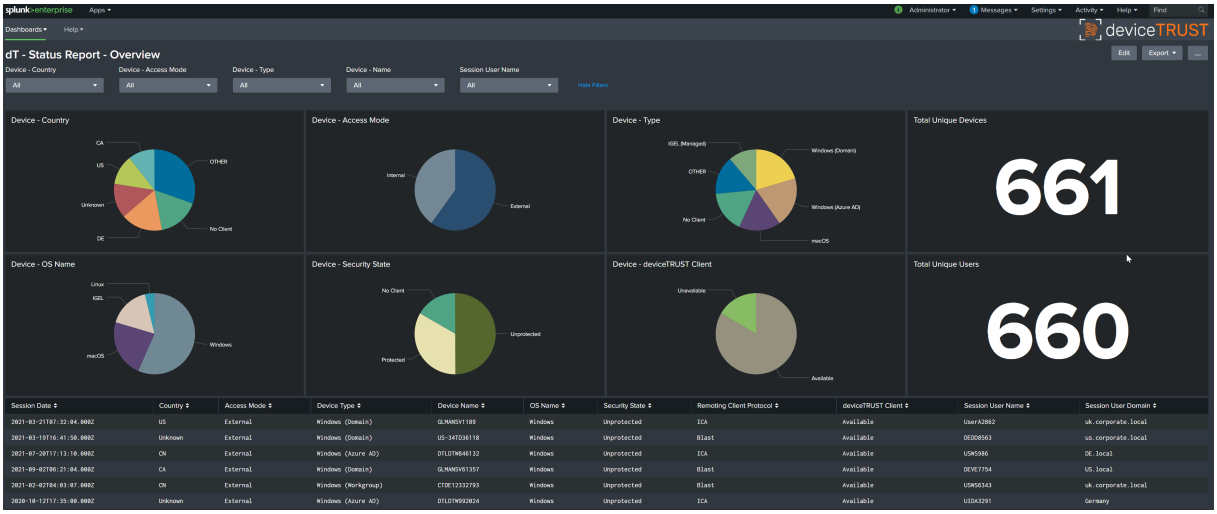
Splunk Dashboards

deviceTRUST® includes a Splunk app to easily create a Splunk dashboard to monitor the contextual status of your remoting and DaaS environment.

The License Compliance Templates can be used with the Splunk app to monitor or enforce Device-Based Licensing requirements for one or more applications:



The Splunk Status Report Template can be used to monitor the status of your remoting and DaaS environment:



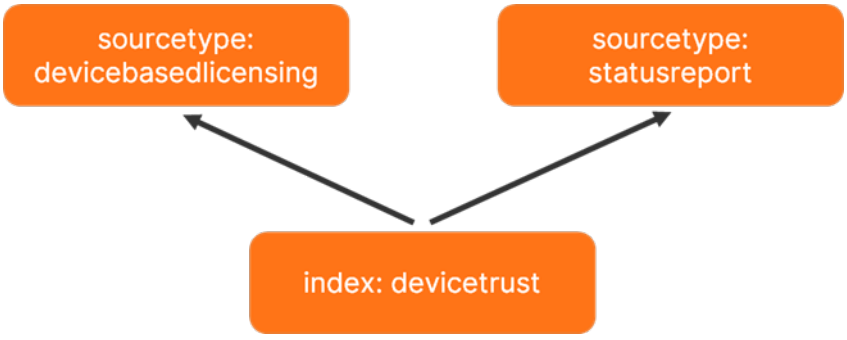
The following steps will be performed:

Step 1: Creating the Splunk Index

Splunk collects data in *indexes* which holds the log data and make it searchable. There are different methods for working with indexes. Data can, for example, flow into one common index. Alternatively, multiple indexes can be created, one for each use case or scenario.

The deviceTRUST reports use one common index `devicetrust` for storing the data. The separation for the different uses cases is done by applying *sourcetypes* `devicebasedlicensing` and `statusreport`.

Our deviceTRUST reports are built on the described combination of index and sourcetypes. If your implementation differs, the reports will have to be adjusted accordingly. Make sure to let us know, we'll happily assist.

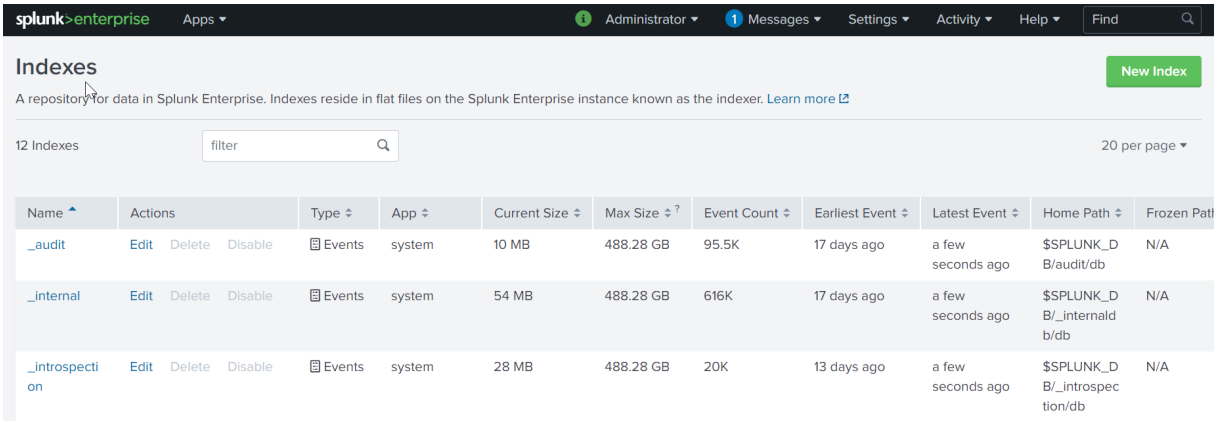


You can either create the index `devicetrust` manually or by importing our prepared app `dt_index`.

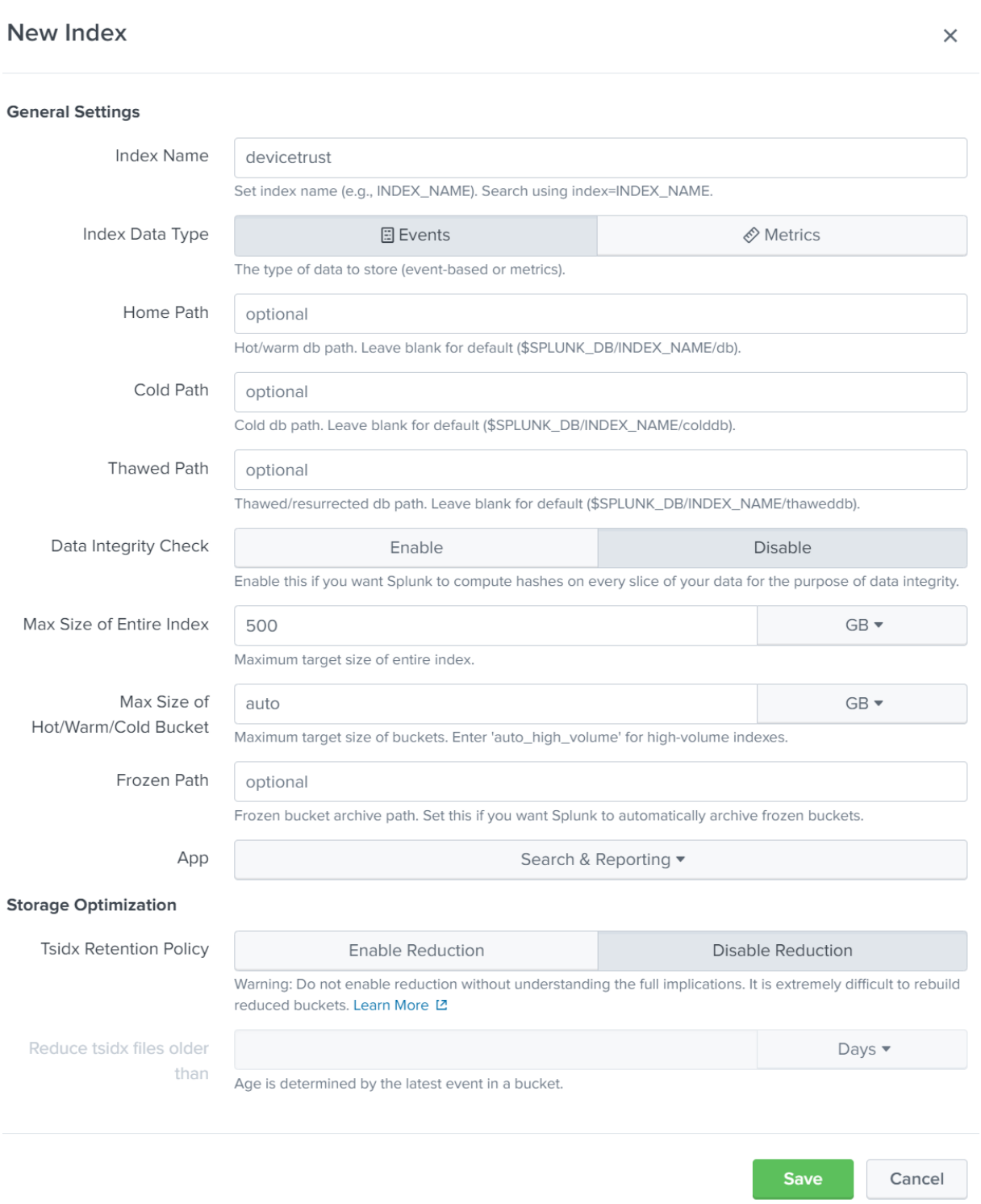
Step 1.1: Manually creating the Splunk Index

Implementing the index manually does not require any special configuration.

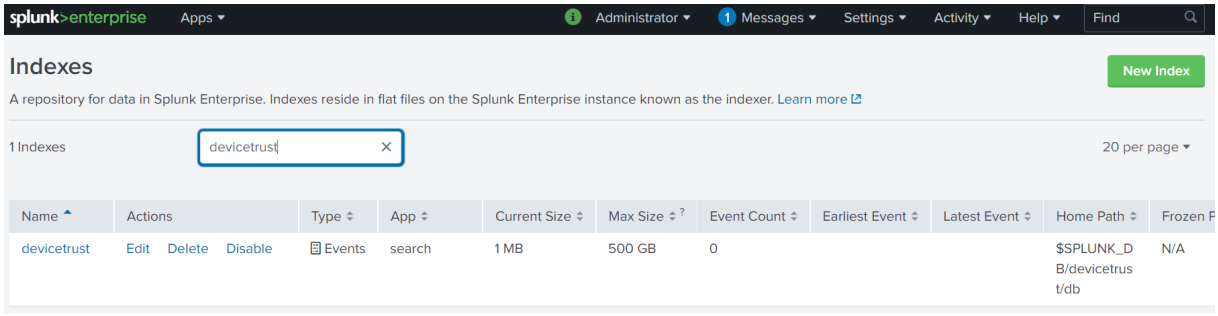
- Open your Splunk management GUI
- Navigate to `Settings\Indexes` and click `New Index`.



- Set the name to `devicetrust` and all other options are optional.



- Click Save and the index will be created.



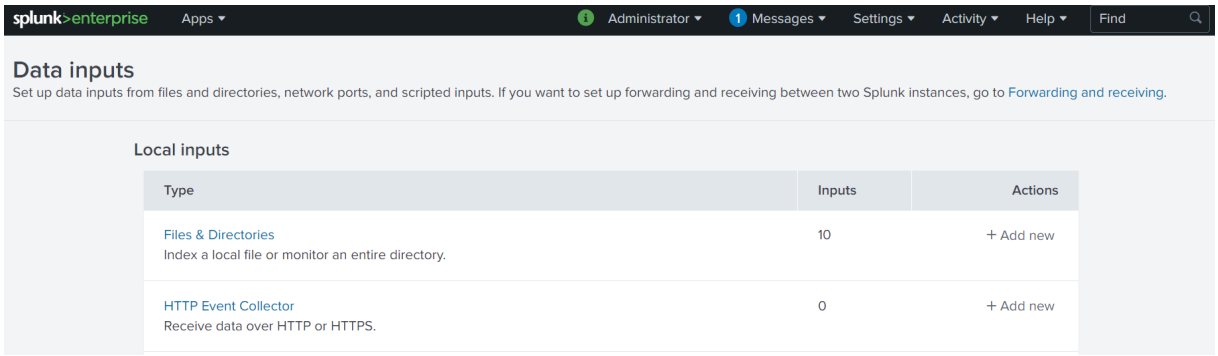
Step 1.2: Creating the Splunk Index by installing the app

To create the index from the app, please refer to Step 3.2: Installing the Splunk app within this guide. The app `dt_index` does not contain any elements besides the index definition for the index `devicetrust`.

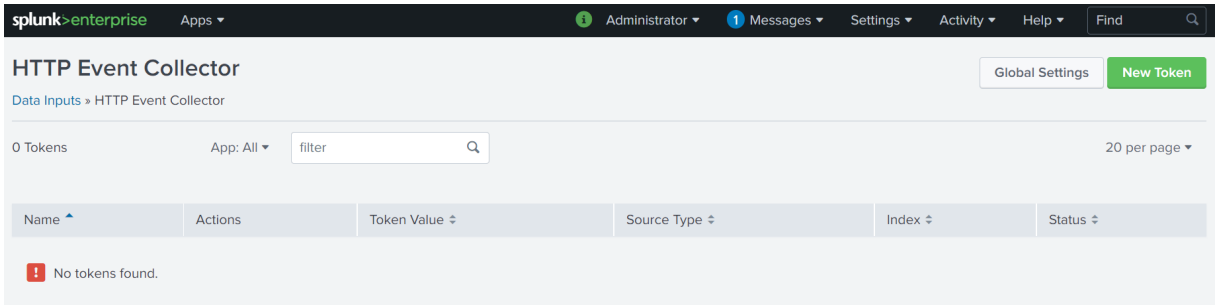
Step 2: Creating the Splunk Data Inputs

Data is sent to the Splunk server by using REST API calls. Splunk needs to be configured to accept http-based inputs. An authentication token is generated, that will be added to the deviceTRUST Console configuration later.

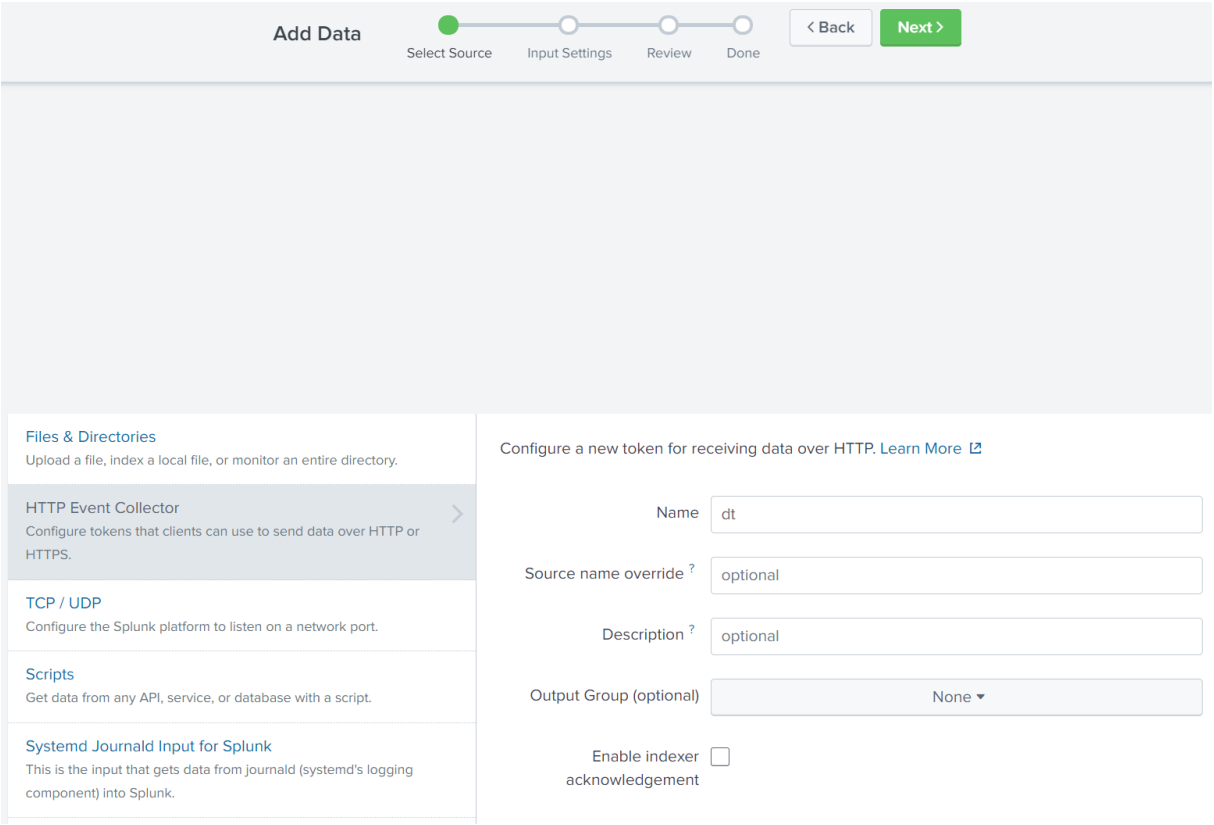
- Open your splunk management GUI and navigate to `Settings\Data Inputs`.



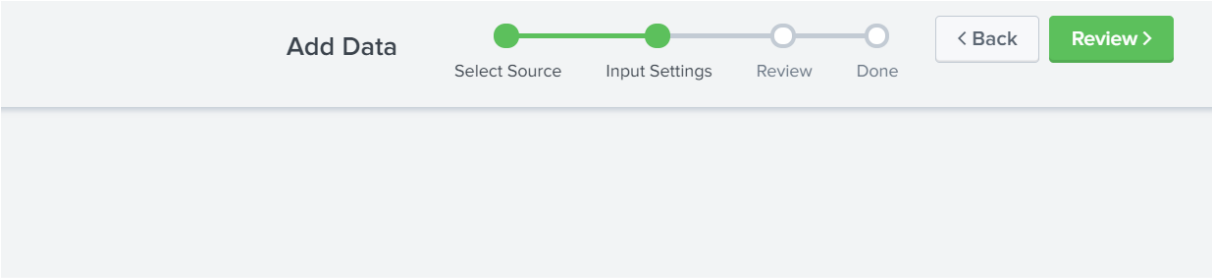
- Select `HTTP Event Collector`
- Click ‘New Token



- Set a `name` of your choice.
- All other settings are optional.



`Input settings` does, for example, allow to restrict access for this data input to certain indexes. Any of these settings may be relevant for your environment. The function of the deviceTRUST reports will not be affected by setting them.



Input Settings

Optionally set additional input parameters for this data input as follows:

Source type

The source type is one of the default fields that the Splunk platform assigns to all incoming data. It tells the Splunk platform what kind of data you've got, so that the Splunk platform can format the data intelligently during indexing. And it's a way to categorize your data, so that you can search it easily.

Automatic | Select | New

App context

Application contexts are folders within a Splunk platform instance that contain configurations for a specific use case or domain of data. App contexts improve manageability of input and source type definitions. The Splunk platform loads all app contexts based on precedence rules. Learn More

App Context: App Exporter (app_exporter)

Index

The Splunk platform stores incoming data as events in the selected index. Consider using a "sandbox" index as a destination if you have problems determining a source type for your data. A sandbox index lets you troubleshoot your configuration without impacting production indexes. You can always change this setting later. Learn More

Select Allowed Indexes

Available item(s) — add all »

- devicetrust
- history
- main
- summary

Selected item(s) « remove all

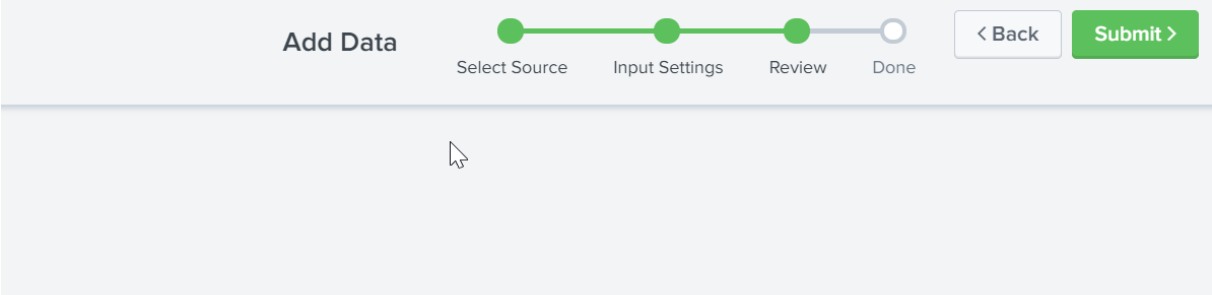
Select indexes that clients will be able to select from.

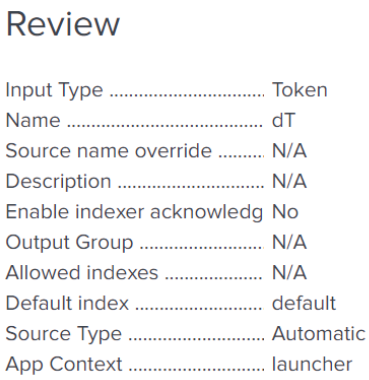
Default Index: Default — Create a new index

FAQ

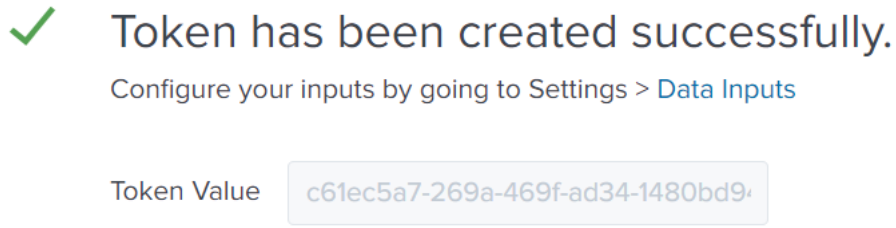
- How do indexes work?
- How do I know when to create or use multiple indexes?

- Review the Settings.
- Click `Submit`.

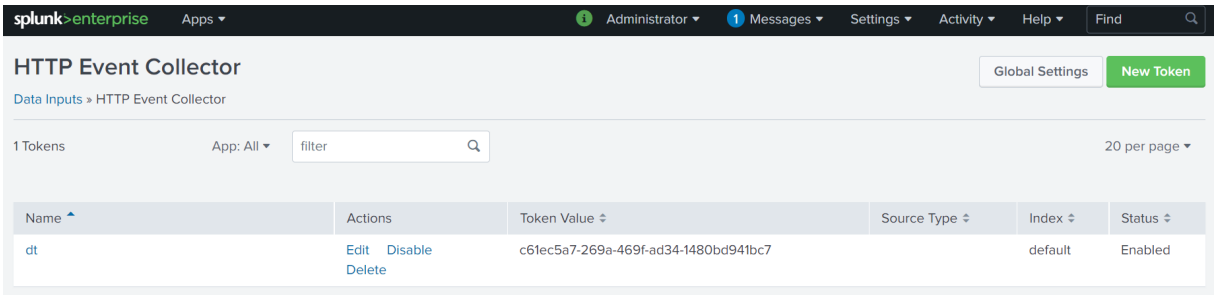




- The created token is shown.



- The created token is also be displayed in the token overview page.

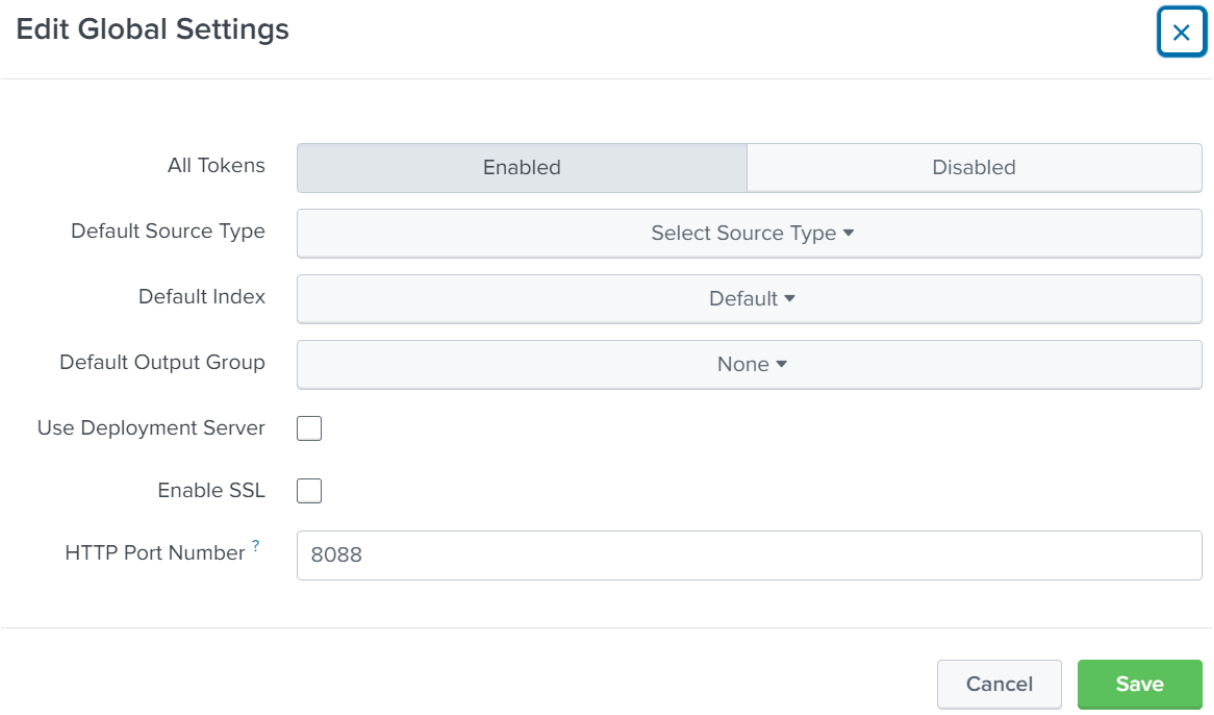


Additionally, the http input method has to be configured. In the most basic configuration, we make sure SSL is not active and all tokens are enabled.

These settings may differ in your environment. The function of the deviceTRUST reports will not be

affected by setting them accordingly.

- Open your splunk management GUI and navigate to `Settings\Data Inputs\HTTP Data Collector`
- Click `Global Settings`
- Set `All Tokens` to Enabled.
- Set `Enable SSL` to Off.

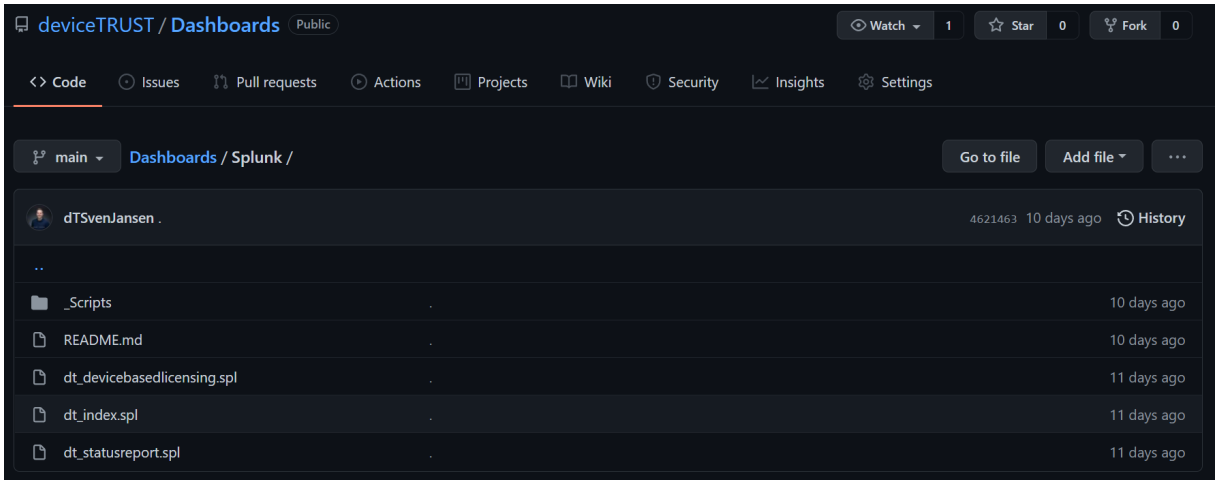


Step 3: Importing the Splunk app

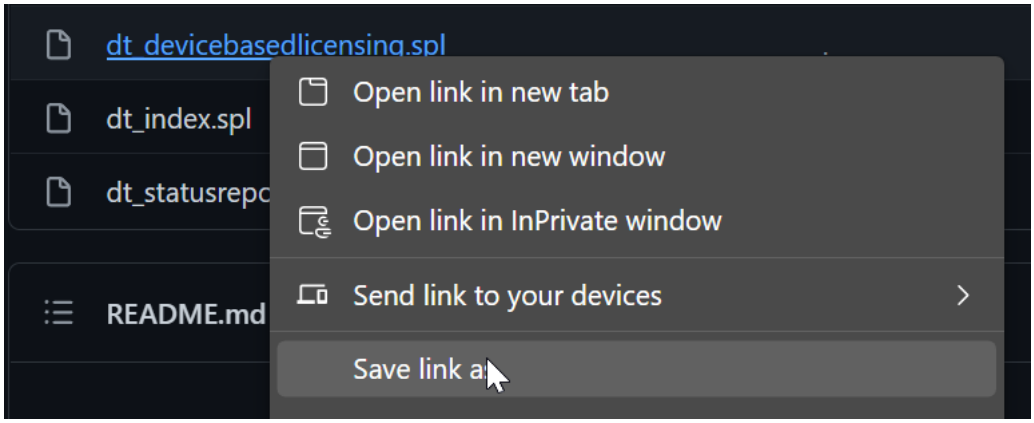
Step 3.1: Downloading the Splunk app

The deviceTRUST reports are delivered as Splunk apps. The apps is available from the deviceTRUST GitHub repository .

- Navigate to the repository and find the Splunk apps.



- The apps are provided as “spl”files which should be downloaded.



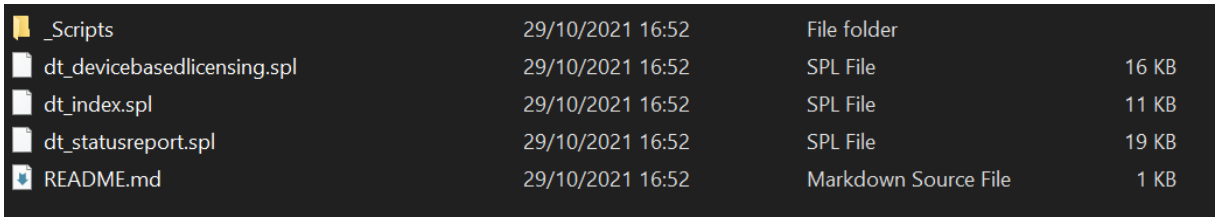
- Alternatively the “spl”files can be synchronized via the GIT command line.


```
PS C:\_Data\GIT> git clone https://github.com/deviceTRUST/Dashboards.git
Cloning into 'Dashboards'...
remote: Enumerating objects: 205, done.
remote: Counting objects: 100% (205/205), done.
remote: Compressing objects: 100% (140/140), done. eceiving objects:  13% (27/205)
remote: Total 205 (delta 72), reused 184 (delta 54), pack-reused 0
Receiving objects: 100% (205/205), 319.08 KiB | 3.39 MiB/s, done.
Resolving deltas: 100% (72/72), done.
PS C:\_Data\GIT> cd .\Dashboards\Splunk\
PS C:\_Data\GIT\Dashboards\Splunk> ls


    Directory: C:\_Data\GIT\Dashboards\Splunk


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d-----        10/29/2021   4:52 PM                _Scripts
-a----        10/29/2021   4:52 PM          15444 dt_devicebasedlicensing.spl
-a----        10/29/2021   4:52 PM          11195 dt_index.spl
-a----        10/29/2021   4:52 PM          19415 dt_statusreport.spl
-a----        10/29/2021   4:52 PM            810 README.md
```

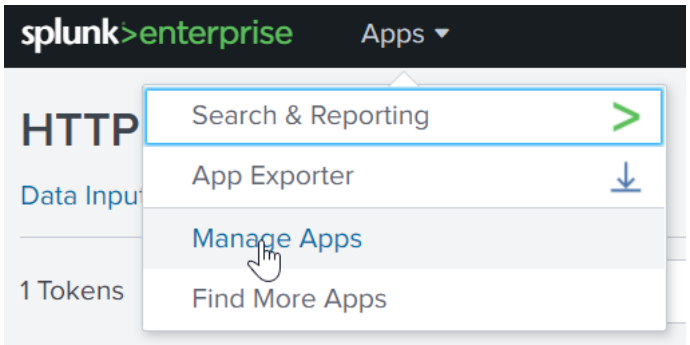
- With the apps available locally, they can be imported into Splunk.



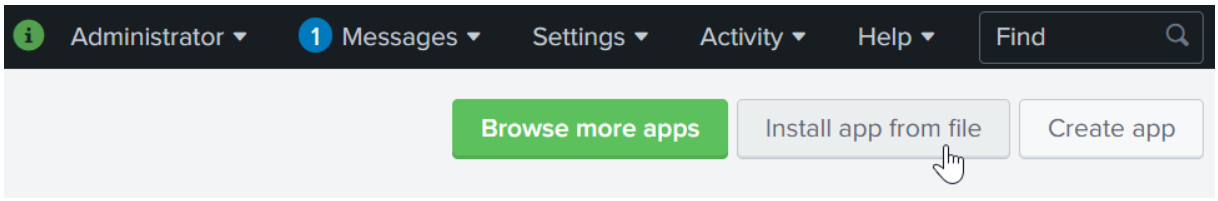
Step 3.2: Installing the Splunk app

All three deviceTRUST apps are installed the same way. Thus, the app installation is described by using one example.

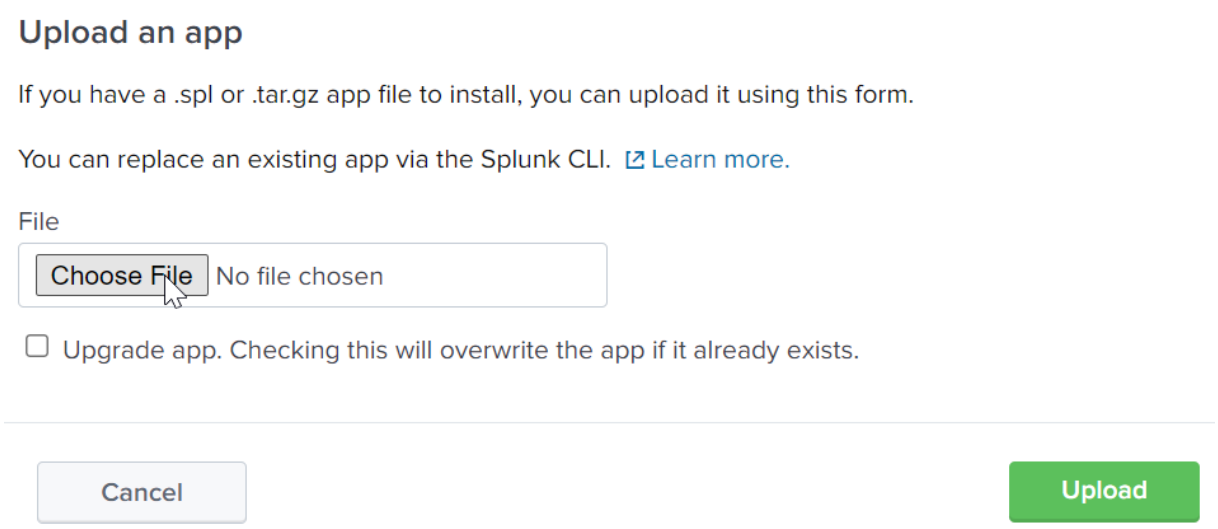
- Open your Splunk management console.
- Click `Manage Apps` in the apps menu.



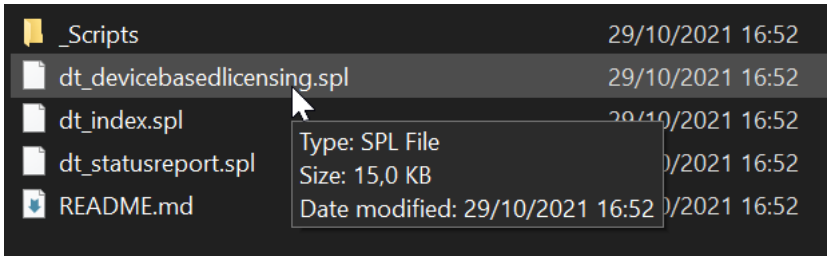
- Click `Install app from file`.



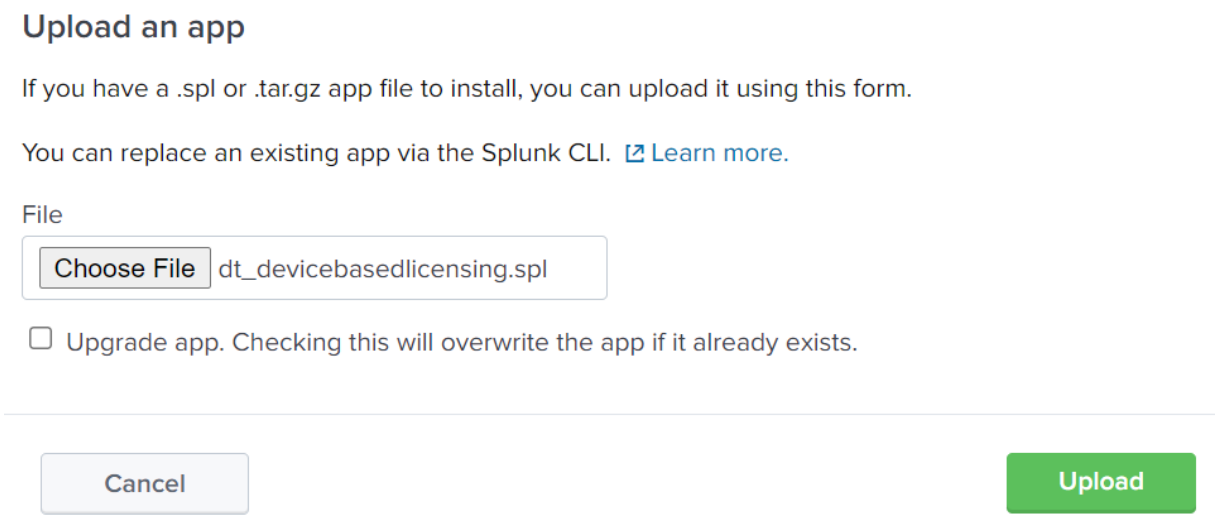
- Click `Choose File`.



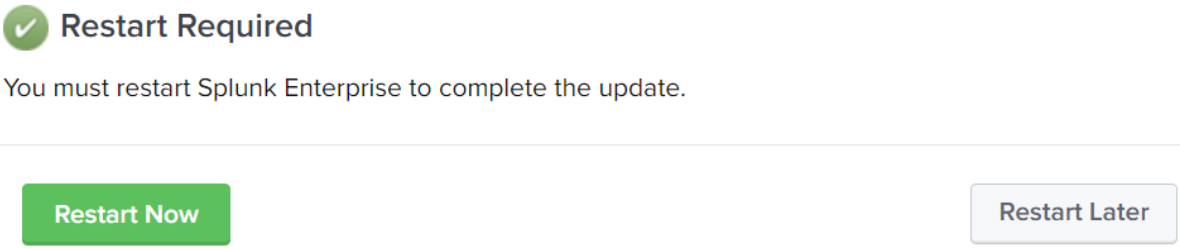
- Select your app file.



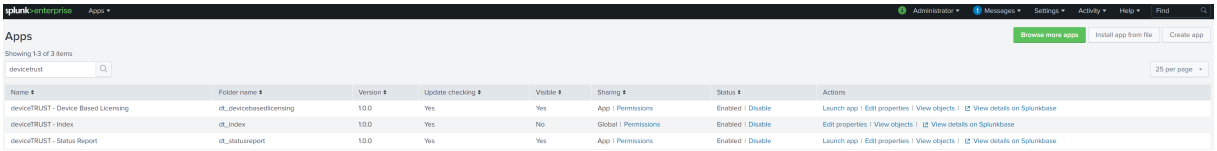
- Confirming your selection.
- Optionally select to upgrade apps, if applicable.
- Click `Upload`.



- Splunk will ask to restart the service. This is only required after importing the last app.
- Choose `Restart Now` or `Restart Later` accordingly.



- The menu `Apps\Manage Apps` displays all apps that are installed in your Splunk environment.



Step 4: Configuring deviceTRUST

After the the index has been created, the data input object added and all apps have been imported, Splunk is ready to accept, store and compute data for the deviceTRUST reports.

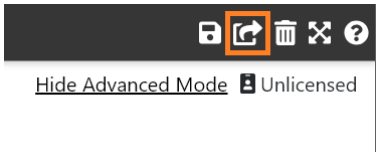
As both reports `Device-Based Licensing` and `Status Report` differ in their details, both are described here separately.

Step 4.1: The Device-Based Licensing Report

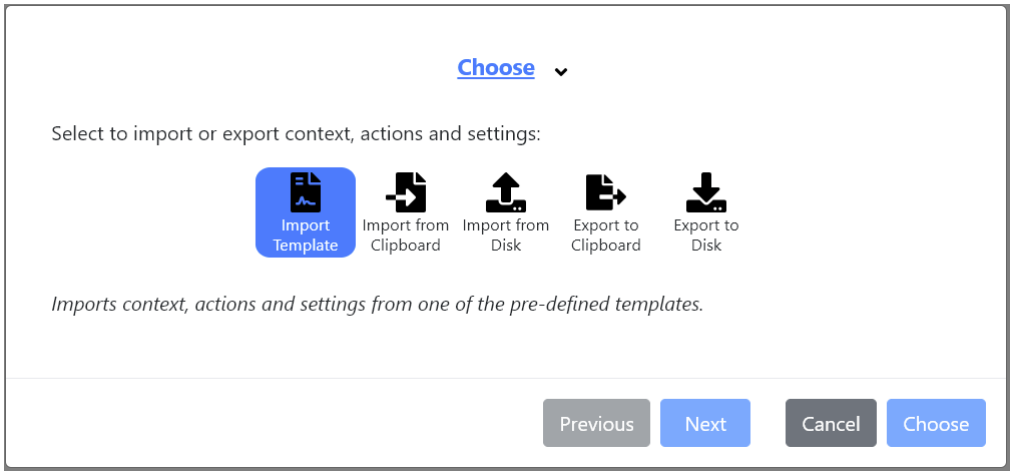
This step describes the configuration to be added to the deviceTRUST Console to send `Device-Based Licensing` data to Splunk.

The integrated templates contain all elements that are required to fully configure the agent for the Device-Based Licensing of five example applications. These five example applications can be easily edited for your own applications, or cloned to represent new applications. The elements are contexts, actions, messages and settings.

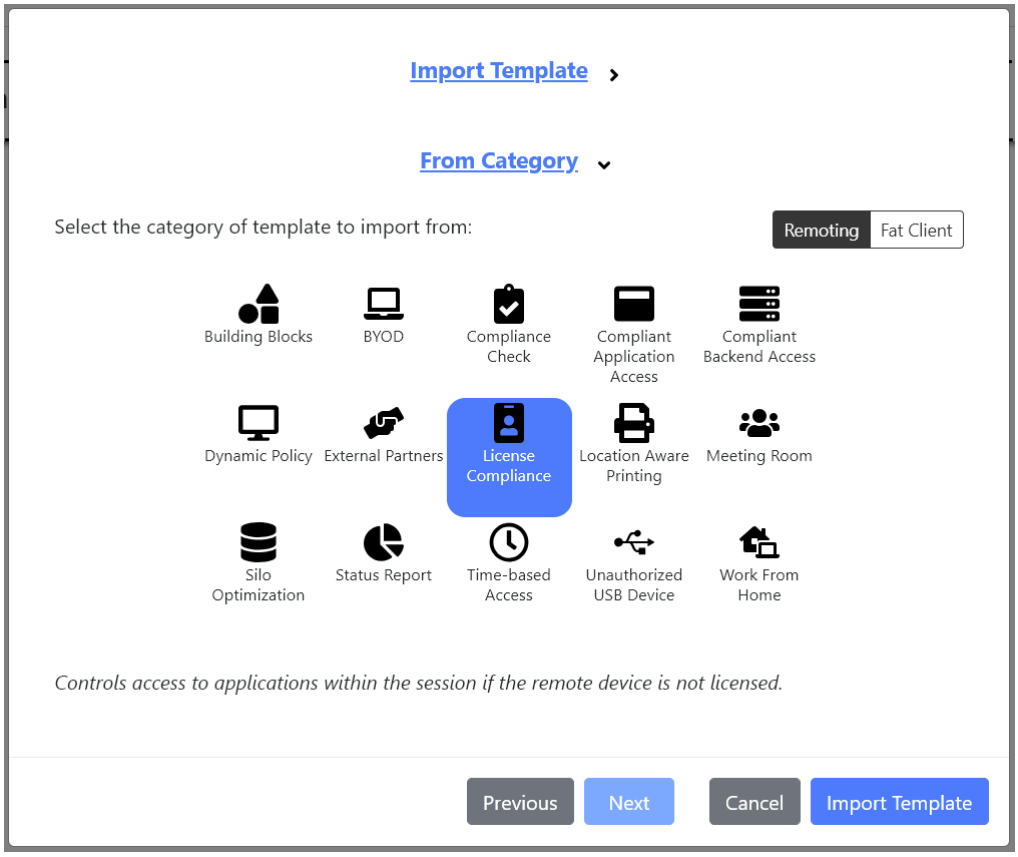
- Open the `deviceTRUST Console`.
- Click `Sharing` in the top right menu. You may need to click `Show Advanced View` if this button is not visible.



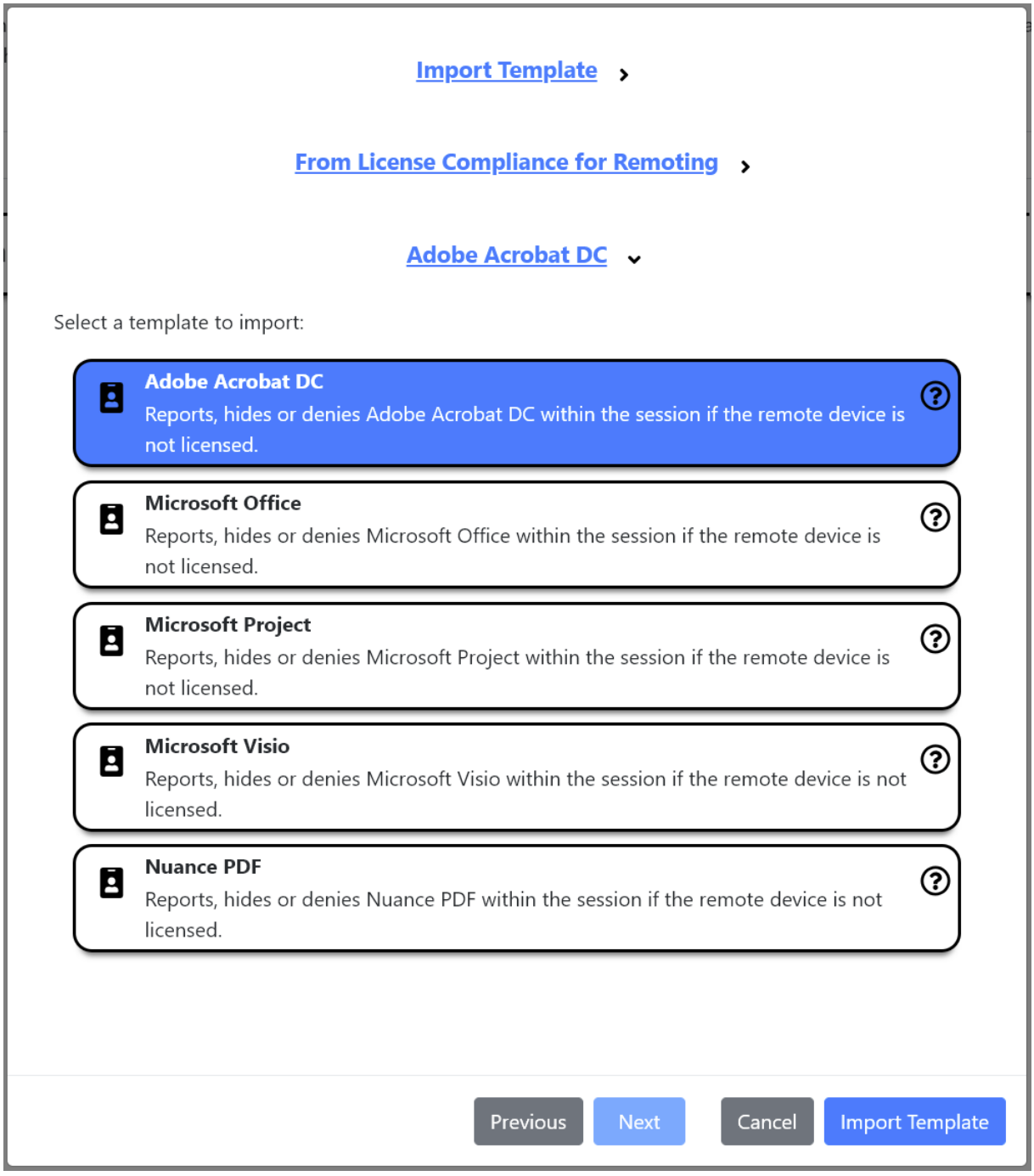
- Select `Import Template`



- The report `Device-Based Licensing` can be found within the template category `License Compliance` when `Remoting` is selected.

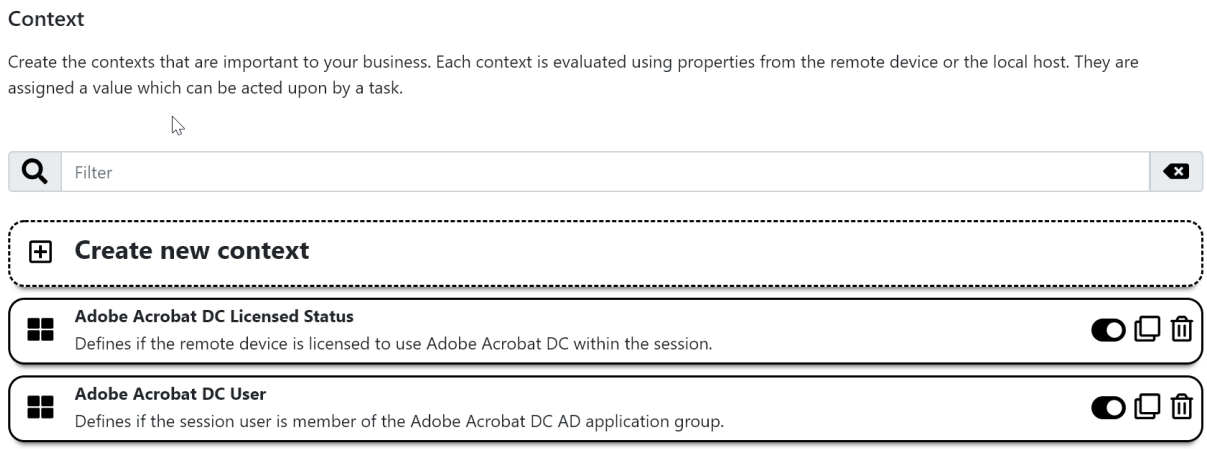


- The category contains templates for different example applications. We use Adobe Acrobat DC as an example here.



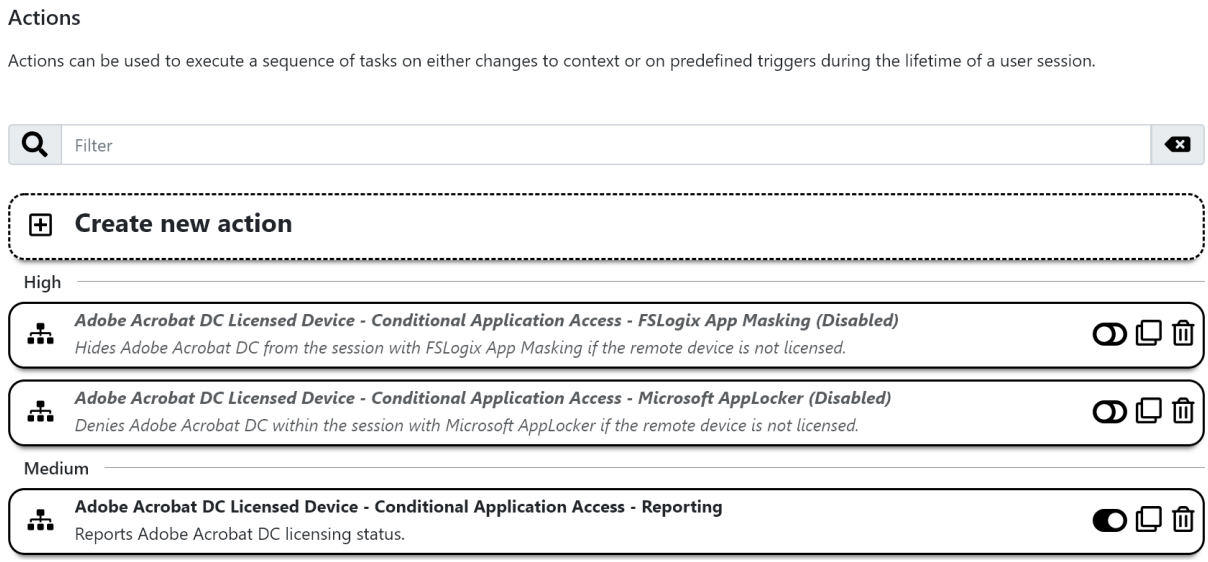
Two contexts are included within the template:

- `Adobe Acrobat DC Licensed Status` to evaluate the device license status.
- `Adobe Acrobat DC User` to define if the accessing user is licensed to use the software.



Three actions are included within the template:

- `Adobe Acrobat DC Licensed Device - Conditional Application Access - FSLogix App Masking` is used for application control via `FSLogix App Masking` and can be ignored or deleted for the reporting use case.
- `Adobe Acrobat DC Licensed Device - Conditional Application Access - Microsoft AppLocker` is used for application control via `Microsoft AppLocker` and can be ignored or deleted for the reporting use case.
- `Adobe Acrobat DC Licensed Device - Conditional Application Access - Reporting` the only action required for reporting.

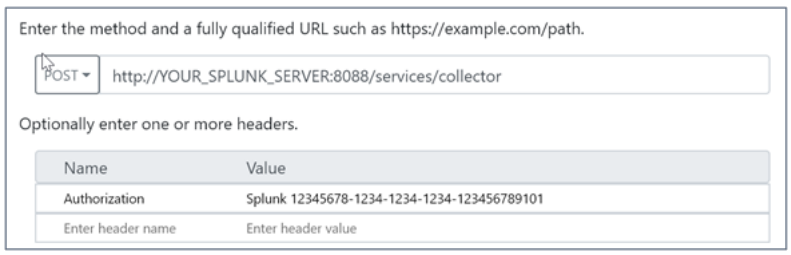


- The action contains uses the `Web Request` task to send data to Splunk. The `Audit Event`, `Custom Process` and also the `Web Request` tasks for ELK Stack and Graylog can be deleted, as we are configuring Splunk here.

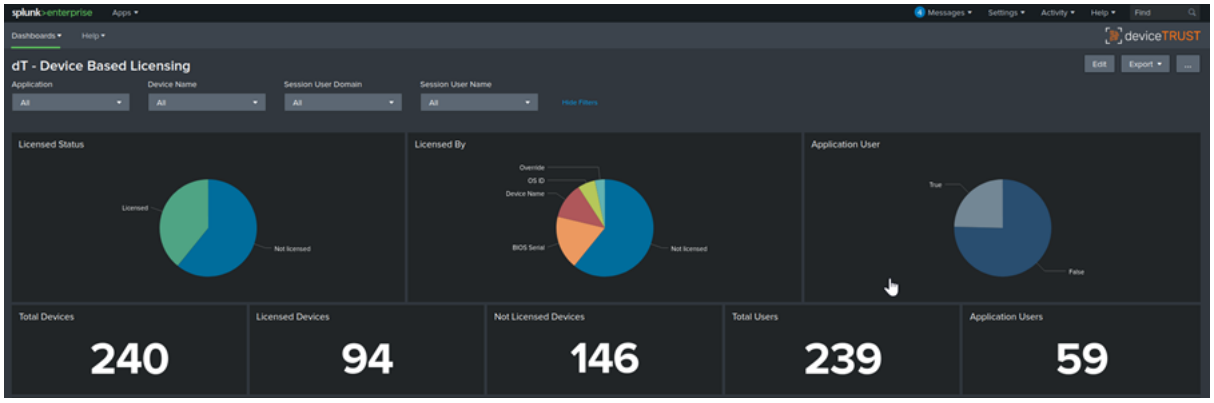
Adobe Acrobat DC Licensed Device - Conditional Application Access - Reporting

Reports Adobe Acrobat DC licensing status.

- The Splunk `Web Request` task needs to be edited to suit your environment. If you do not require SSL transport, you'll only have to configure your `server's fqdn` and the `authorization token` (http data input). Make sure to leave the keyword `splunk` and only add/change your authorization token's GUID.



The use case has successfully been configured. deviceTRUST will now send `Device-Based Licensing` data on every access to the Splunk server. The data will be presented using dashboards.

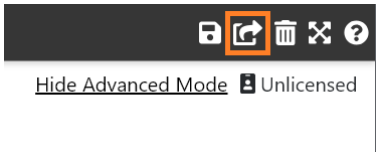


Step 4.2: The Status Report

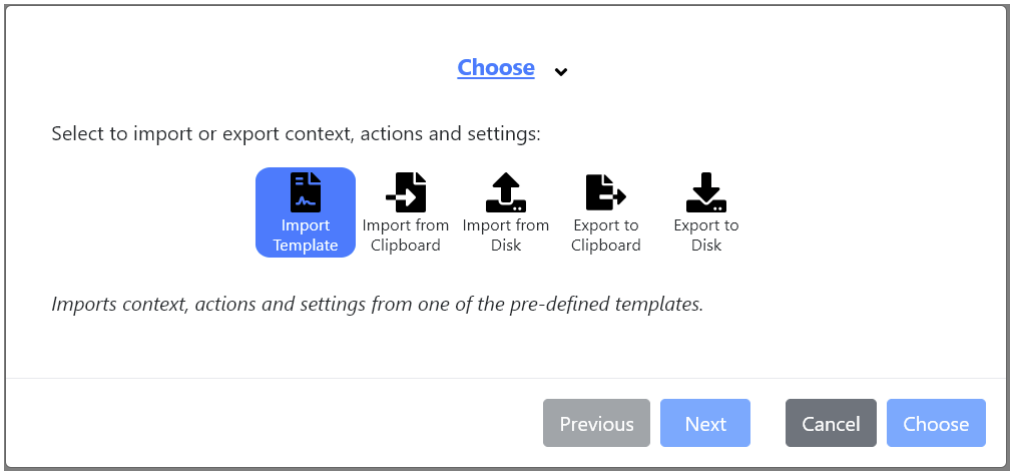
This step describes the configuration, that is to be added to the deviceTRUST console to send `Status Report` data to splunk.

Our integrated templates contain all elements that are required to fully configure the agent for the Device-Based Licensing of five example applications. The elements are contexts, actions, messages and settings.

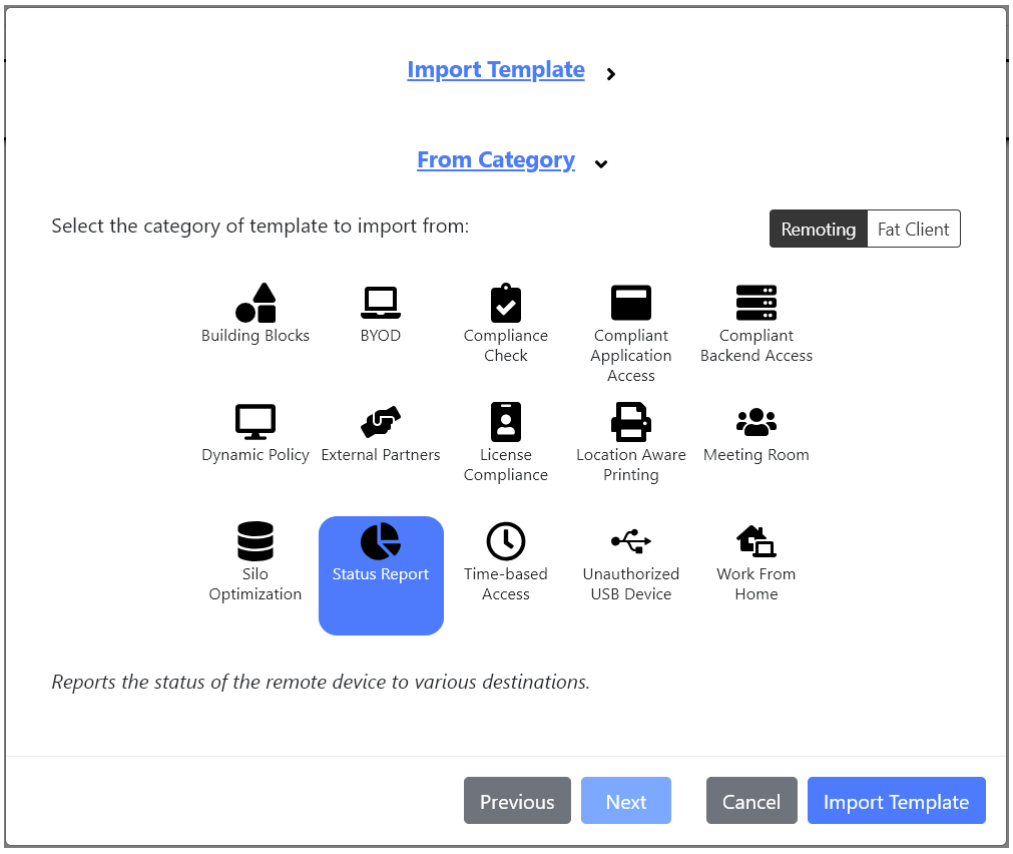
- Open the deviceTRUST console and click `Sharing` in the top right menu.



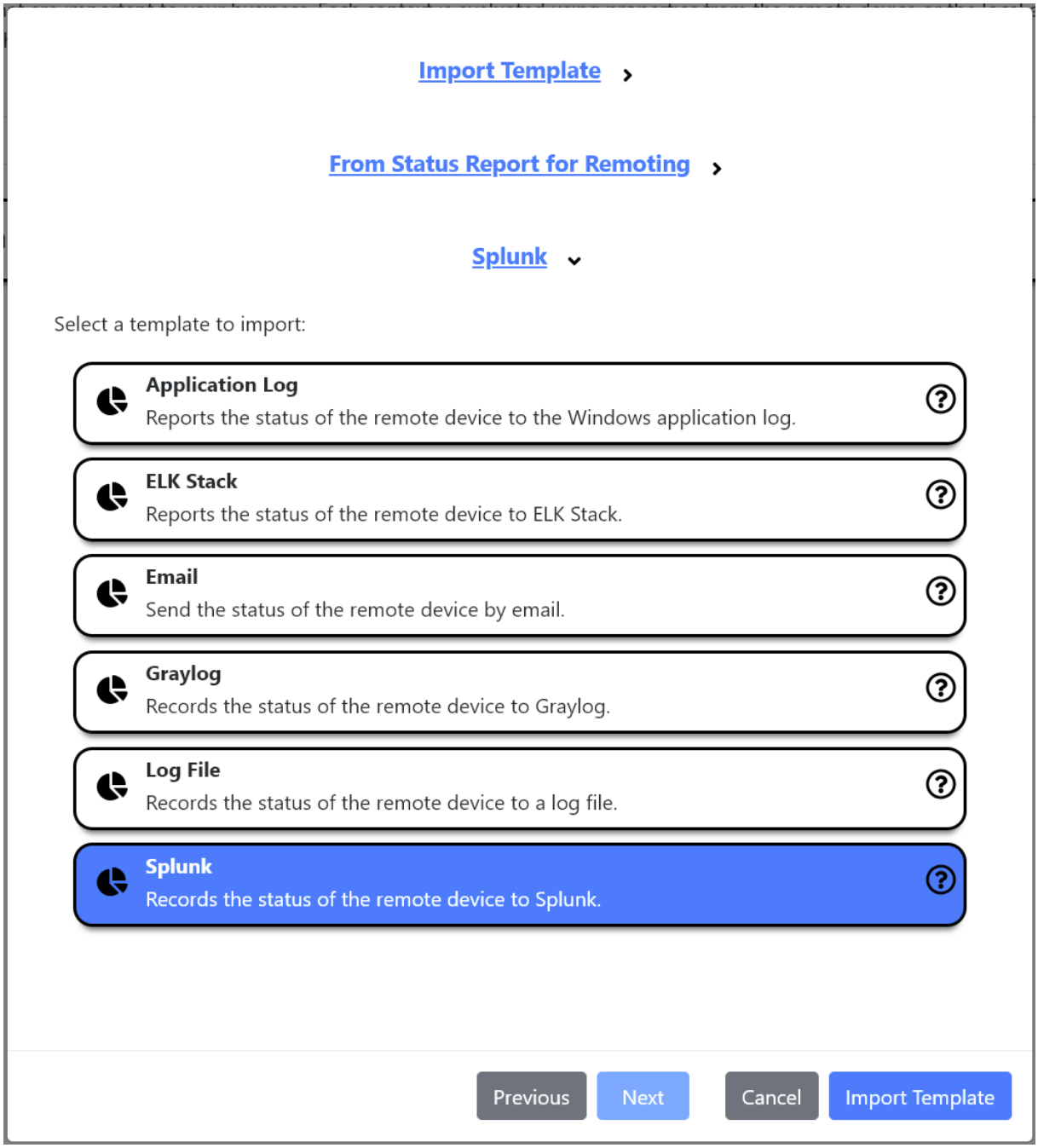
- Select `Import Template`



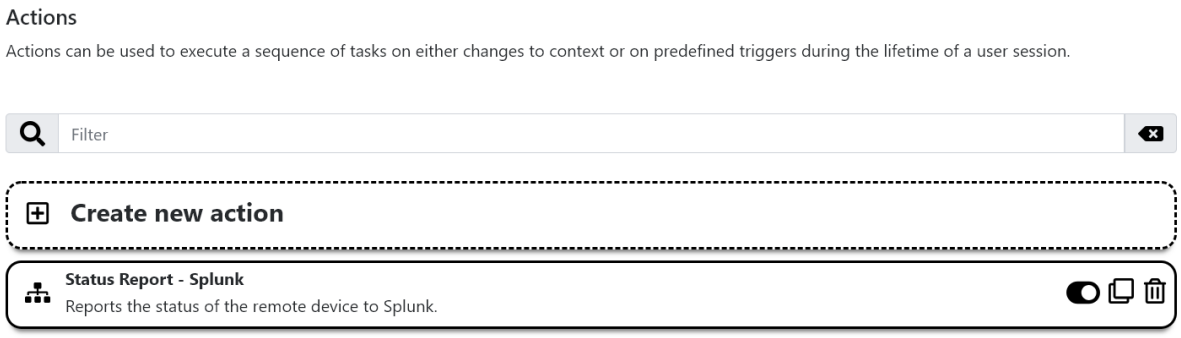
- The `Status Report` template can be found within the template category `Status Report` when `Remoting` is selected.



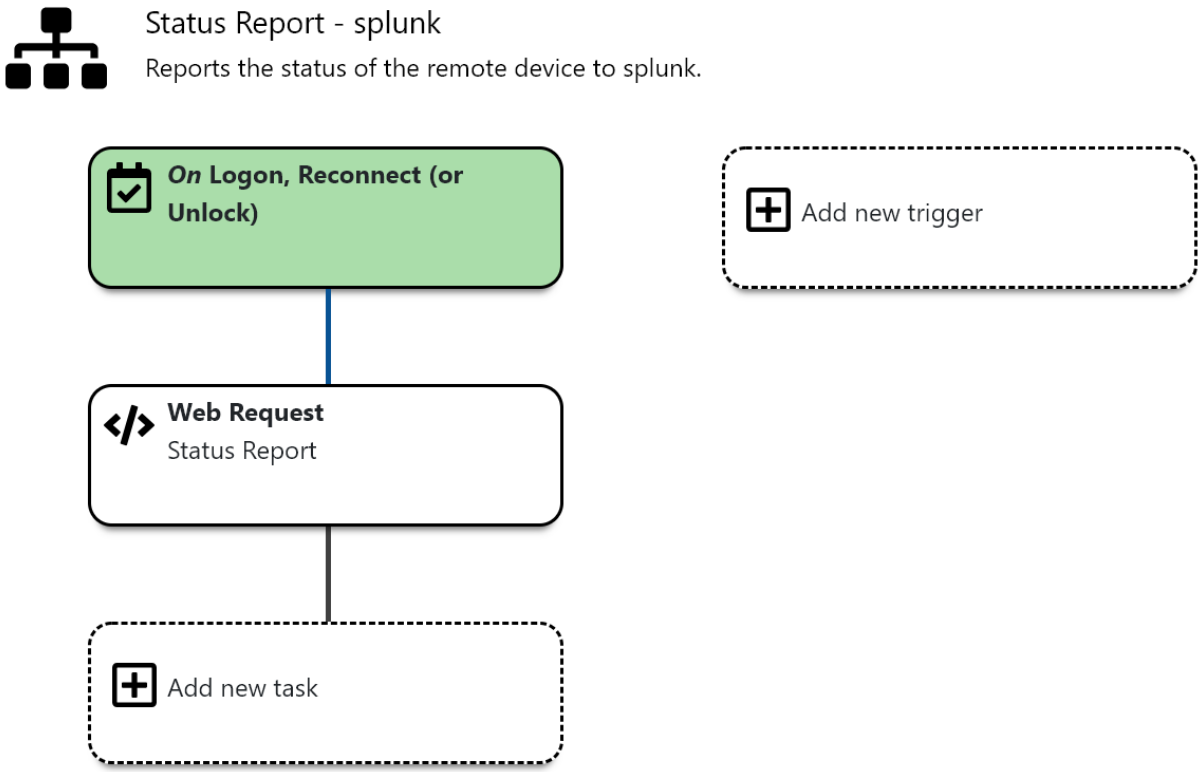
- This category contains templates for several ways of storing data. Choose `Splunk`.



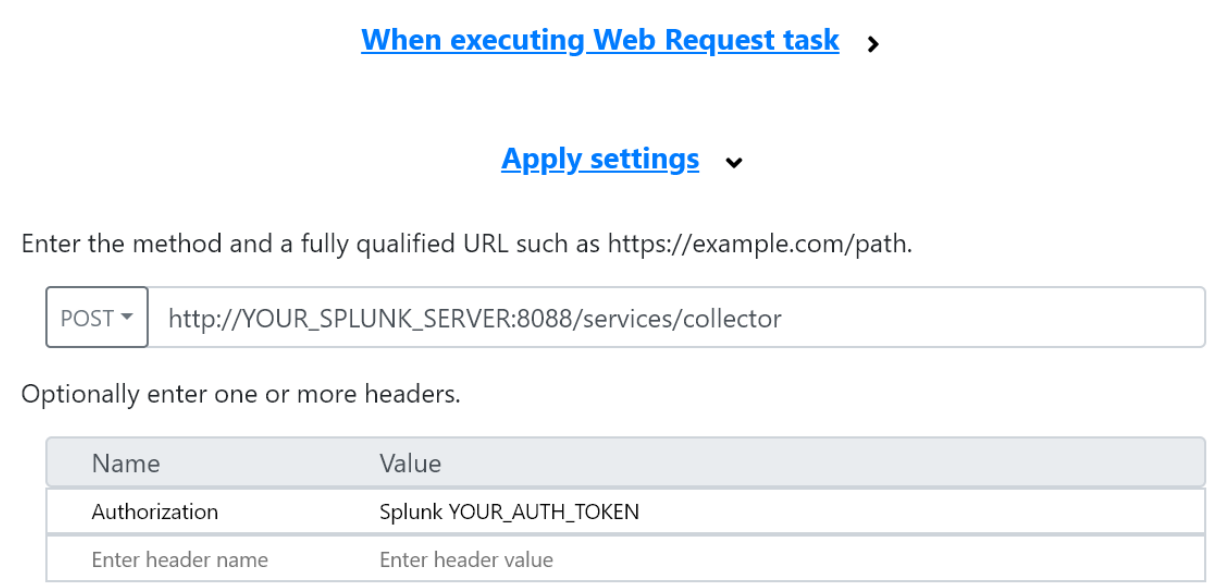
- The imported template consists of 50 contexts and one action.
- The Action `Status Report – Splunk` collects all relevant data and sends them over to Splunk.



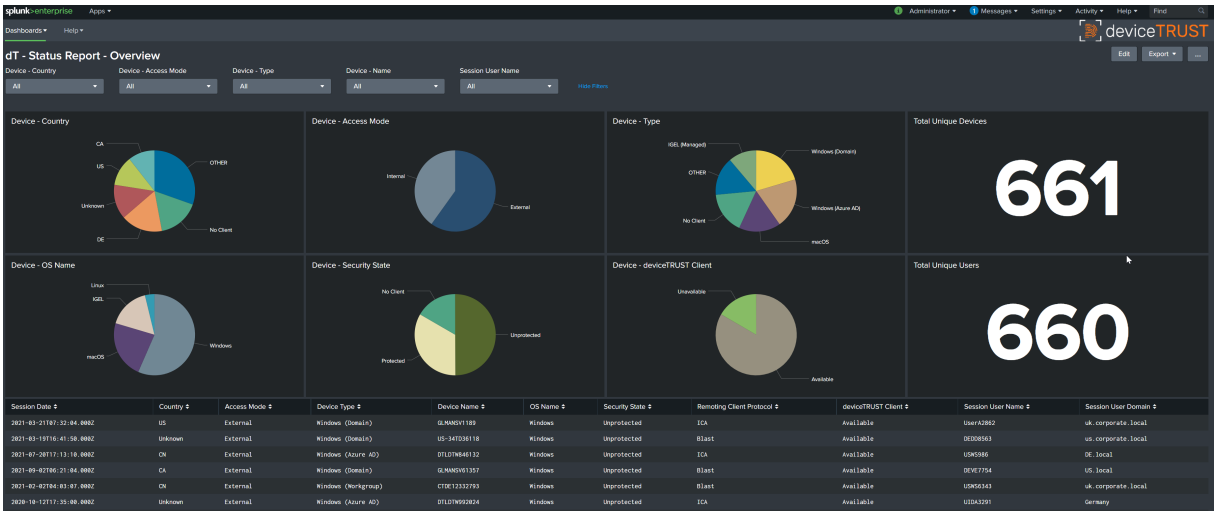
- Sending data to Splunk is configured by using a `Web Request` task.



The Web Request task needs to be edited to suit your environment. If you do not require SSL transport, you'll only have to configure your `server's fqdn` and the `authorization token` (http data input). Make sure to leave the keyword `splunk` and only change your authorization token's GUID.



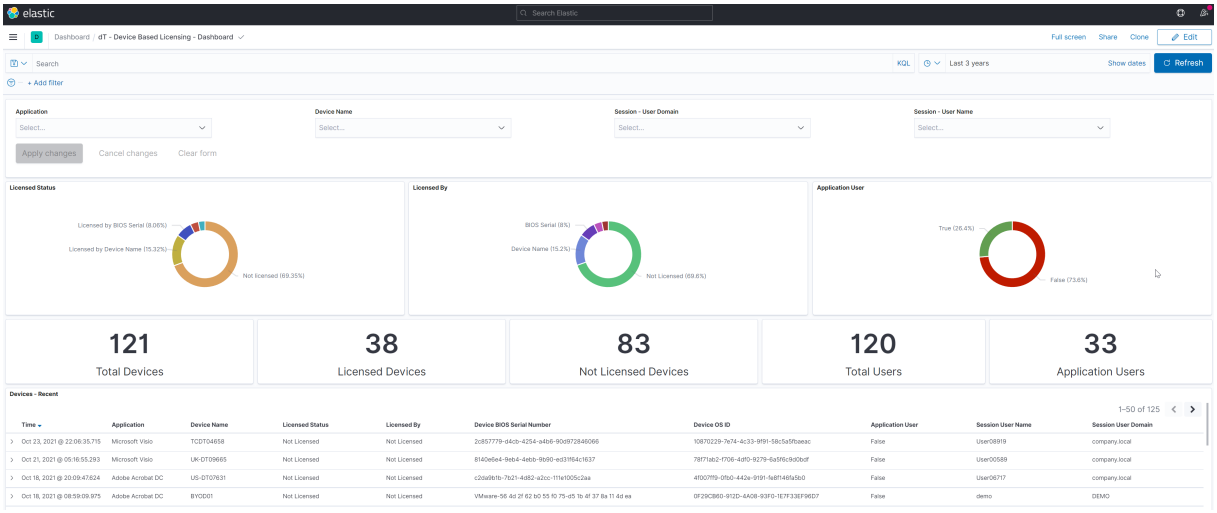
The use case has successfully been configured. deviceTRUST will now send `Status Report` data on every access to the remoting platform to the splunk server. There, the data will be presented using dashboards.



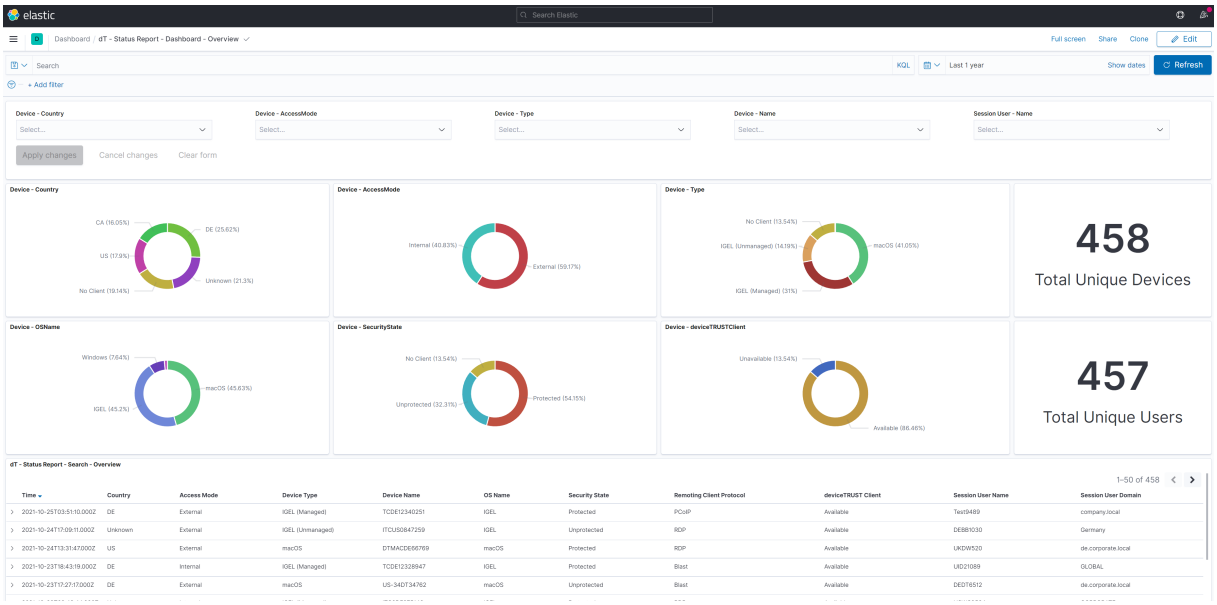
ELK Stack Dashboards

deviceTRUST® includes components for an ELK Stack to easily create a dashboard to monitor the contextual status of your remoting and DaaS environment.

The License Compliance Templates can be used with your ELK Stack to monitor or enforce Device-Based Licensing requirements for one or more applications:



The ELK Stack Status Report Template can be used to monitor the status of your remoting and DaaS environment:



Step 1: Components

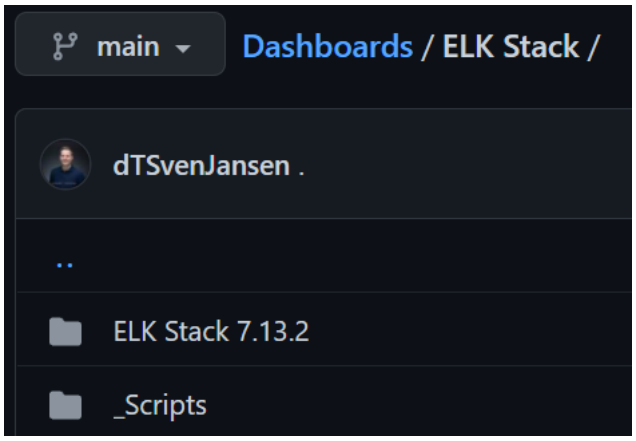
The deviceTRUST dashboards for ELK Stack consist of several components. All elements that need to be imported on the ELK Stack side can be found on GitHub. The configuration for the deviceTRUST Agent is available as a template within the deviceTRUST Console.

Step 1.1: Components - GitHub

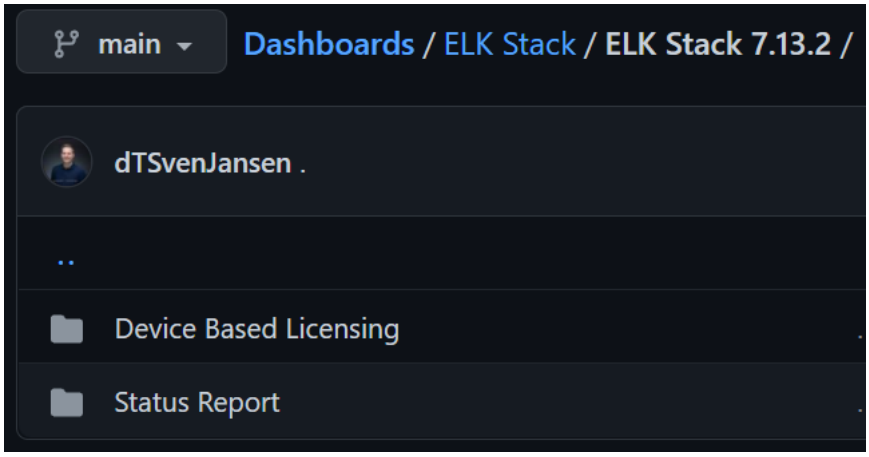
All required components for ELK Stack can be found on our GitHub repository .

> **Note:**
>
> Saved objects for ELK are not backward compatible. Please use only versions matching or older than your system.

- Select the version number matching your ELK Stack system and download the files. You can of course also clone the whole repository if you like.

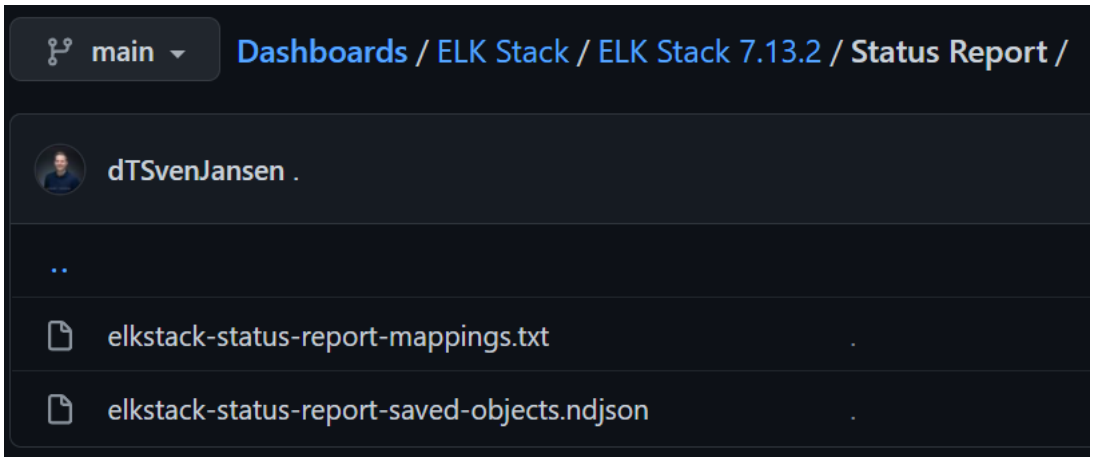


Every version folder contains folders for each use case. Each of these use case folders contains the relevant stored object and mapping files.



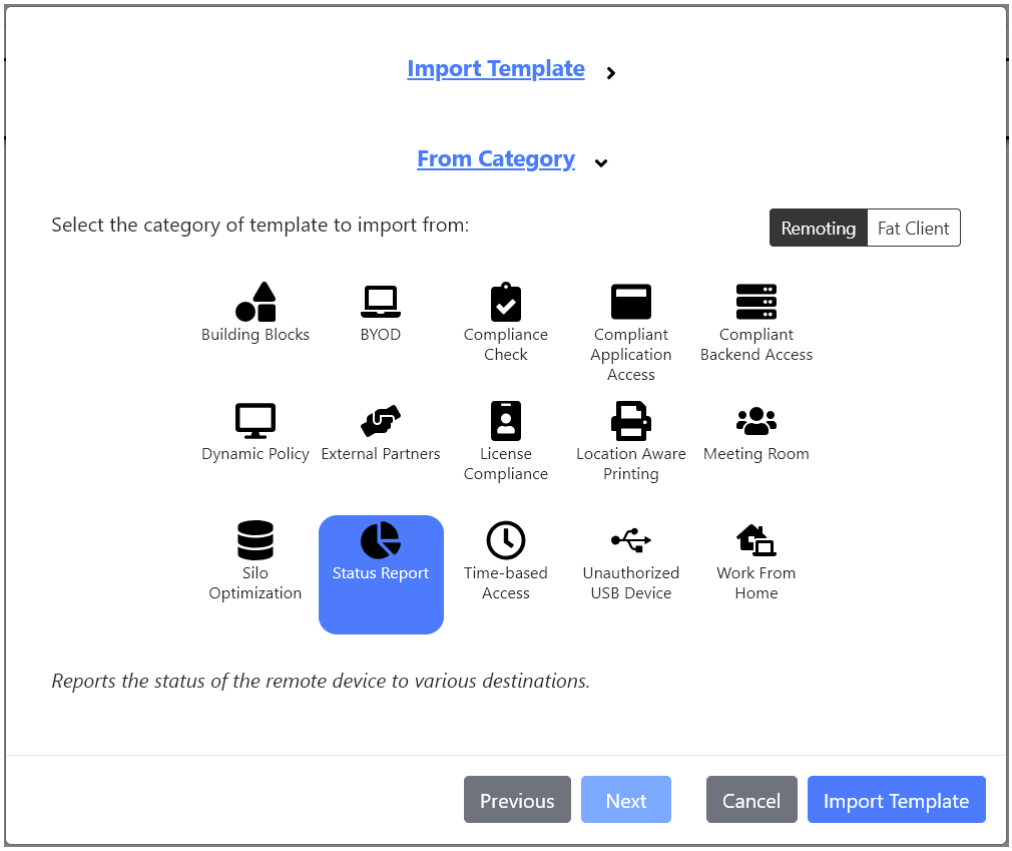
Each use case folder contains the following files:

- `elkstack-<use case>-mappings.txt` contains data definitions for the data being sent to ELK Stack index mapping.
- `elkstack-<use case>-saved-objects.ndjson` contains all objects that are required to store, search and visualize data, such as indexes, scripted fields, dashboards and searches.



Step 1.2: Components - Templates

deviceTRUST must be configured to send the required data for each use case. Templates for both use cases are included in the deviceTRUST Console.



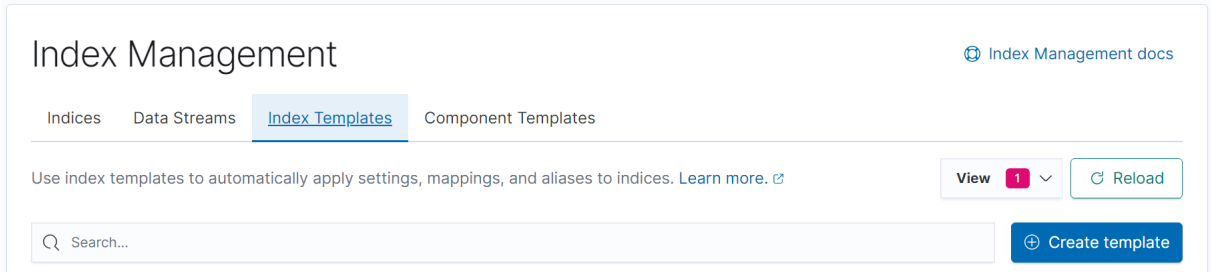
Step 2: The Device-Based Licensing Report

This part of the guide relates to `Device-Based Licensing`. It lists all steps that are required to configure the ELK Stack, and also how to configure the deviceTRUST Agent.

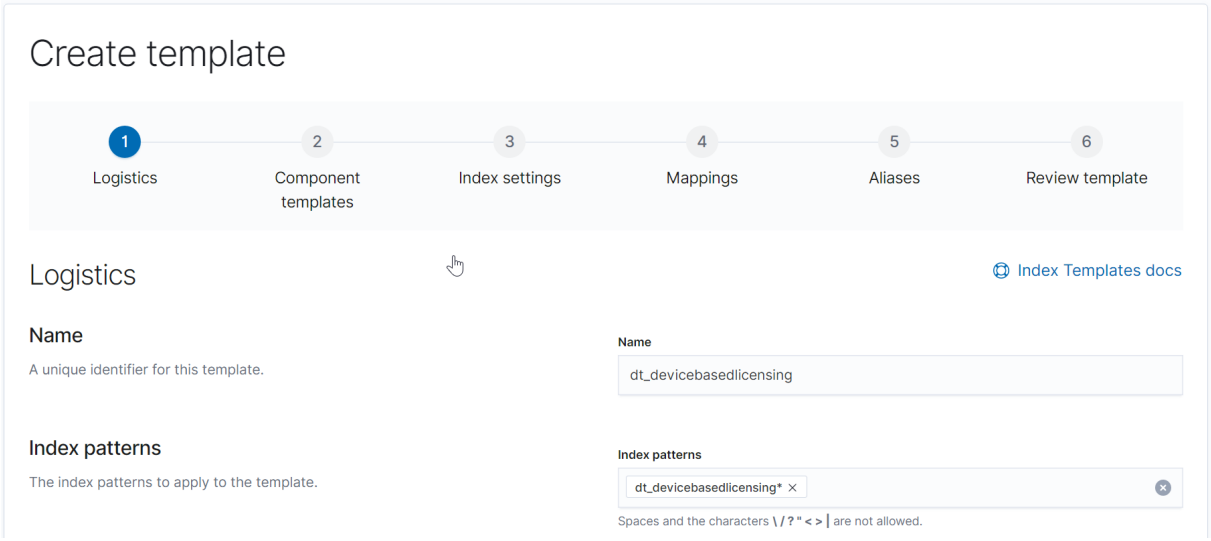
Step 2.1: Add Index Template to ELK Stack

The first step is to create an `Index Template`. Index Templates describe the data that is being sent to an index. They make sure that every date is treated according to its type.

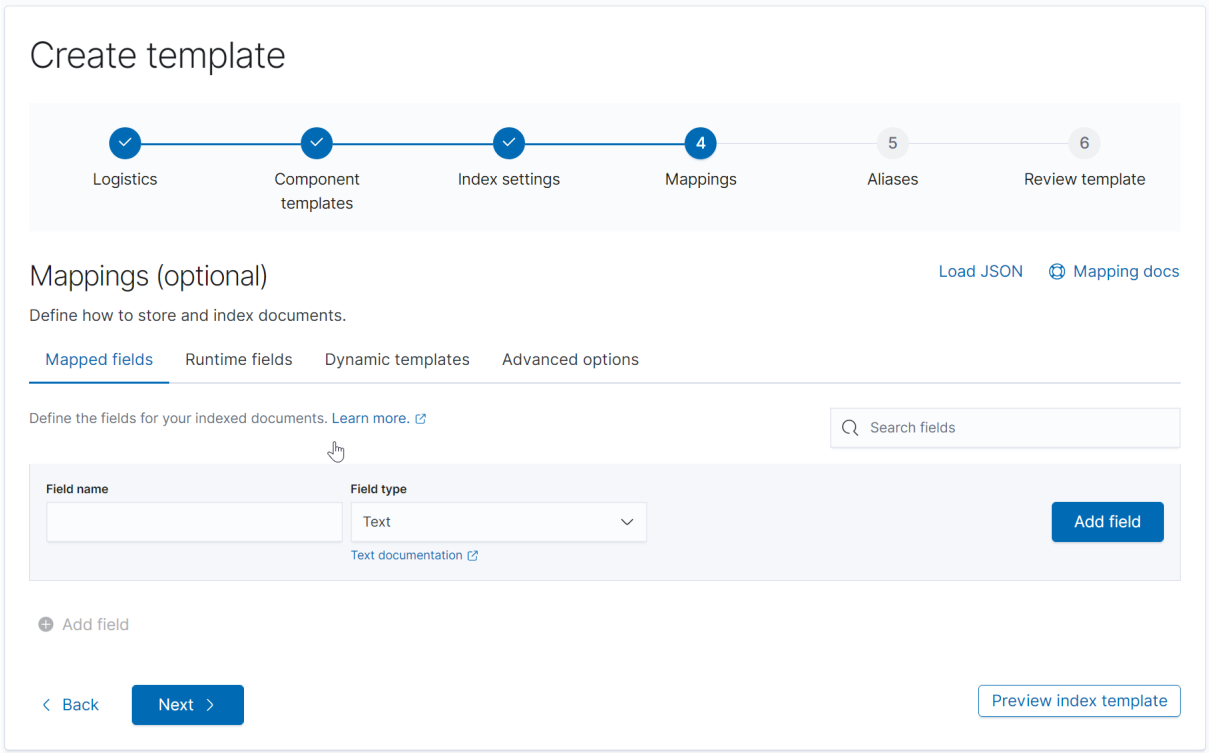
- Within the ELK Stack management console, navigate to `Menu\Stack Management\Index Management\Index Templates`.
- Click `Create Template`.



- Set `Name` to `dt_devicebasedlicensing`.
- Set `Index Patterns` to `dt_devicebasedlicensing*`.



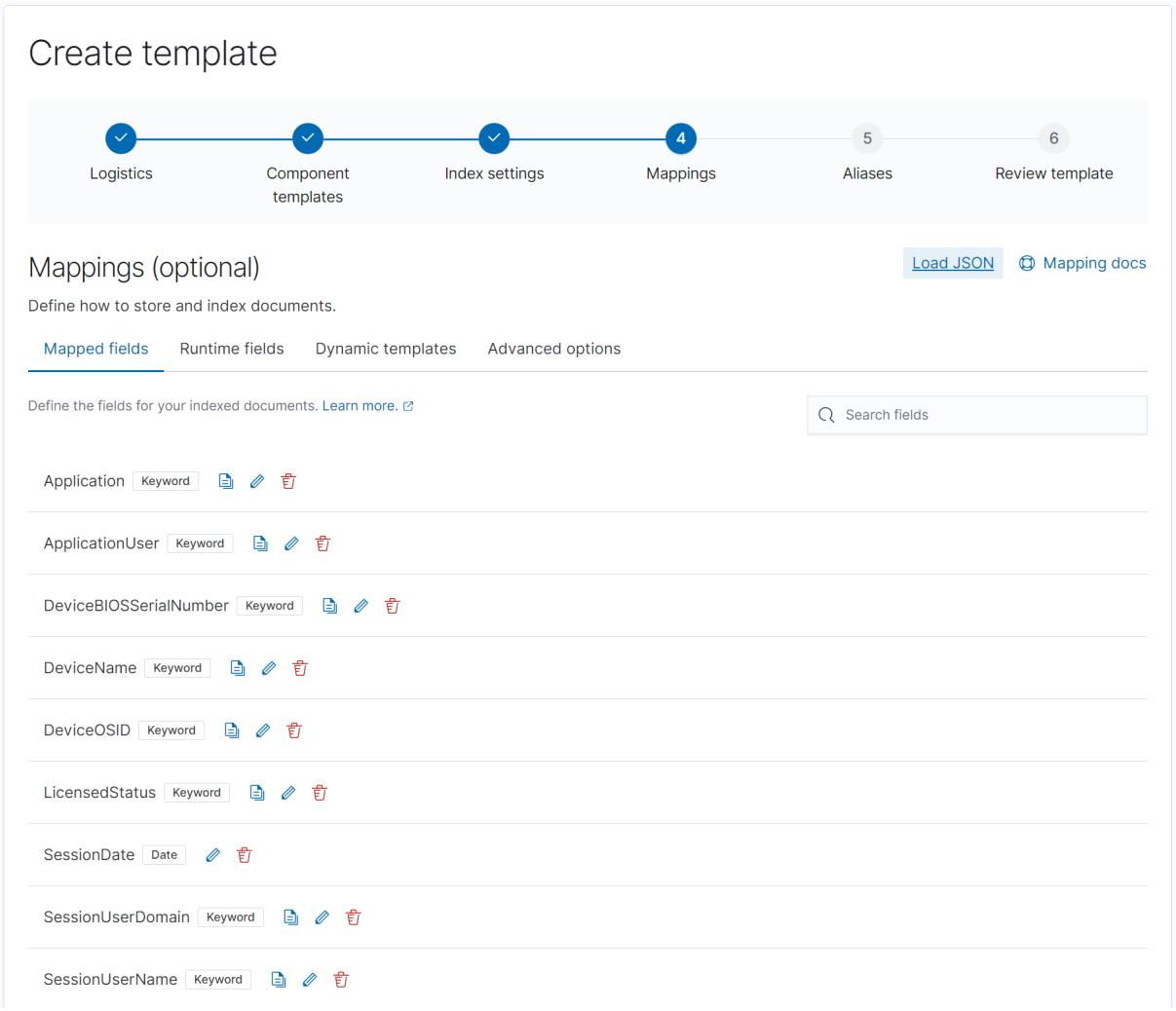
- Skip all options until `Mappings`.
- Select `Load Json` to import the file `elkstack-device-based-licensing-mappings.txt` that was downloaded in Step 1.1.



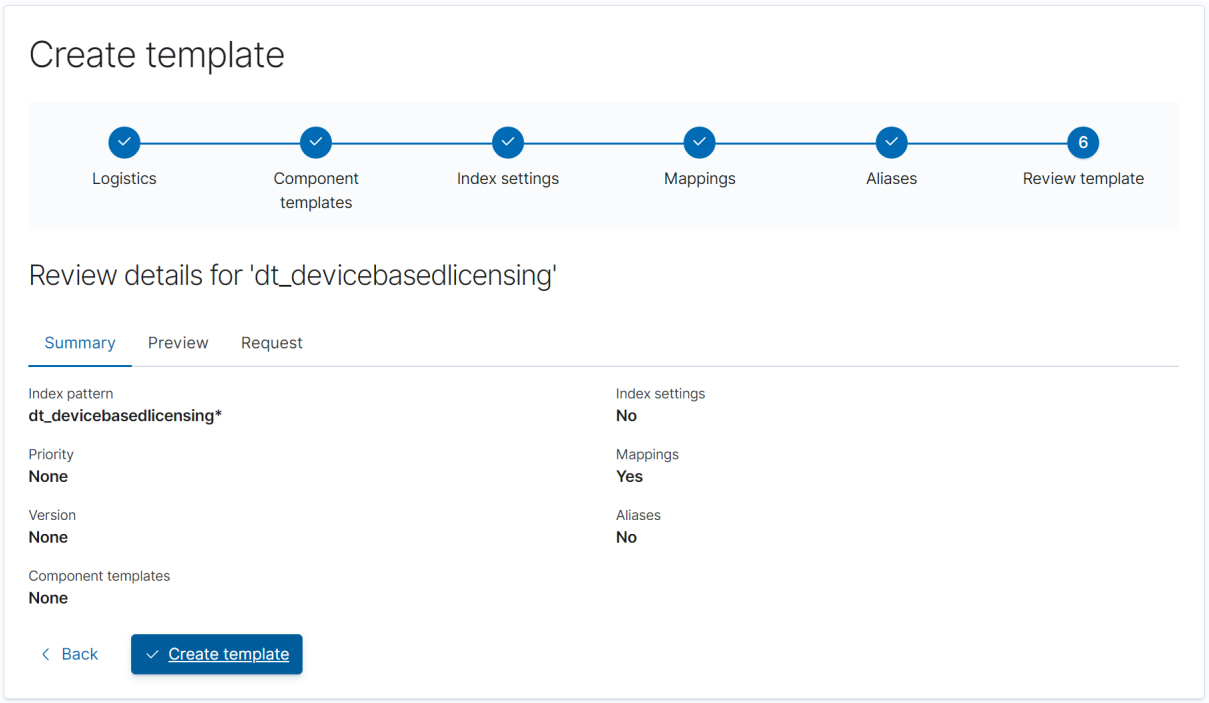
- Proceed with `Load and overwrite`.

The imported mappings are displayed. Please review them carefully.

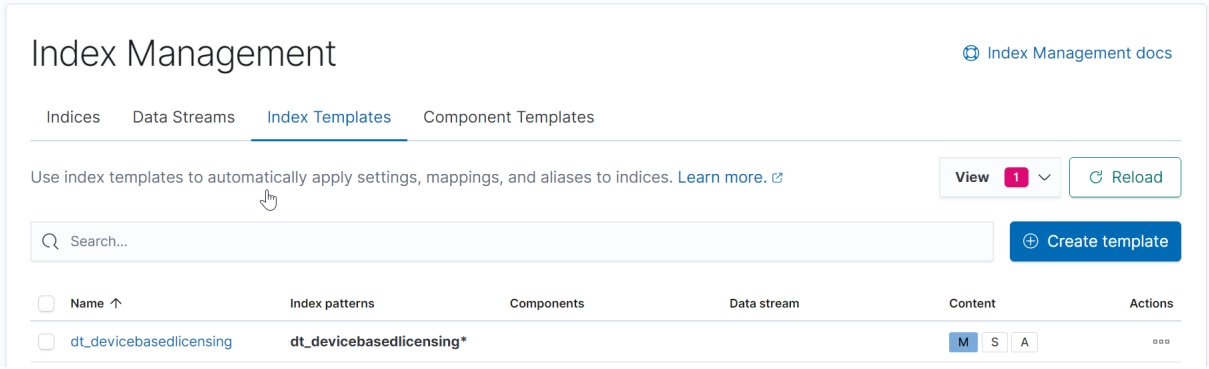
- `SessionDate` needs to be recognized as type `Date` for the report to function properly.



- Skip all options until `Review Template`.
- Generate the template with `Create Template`.



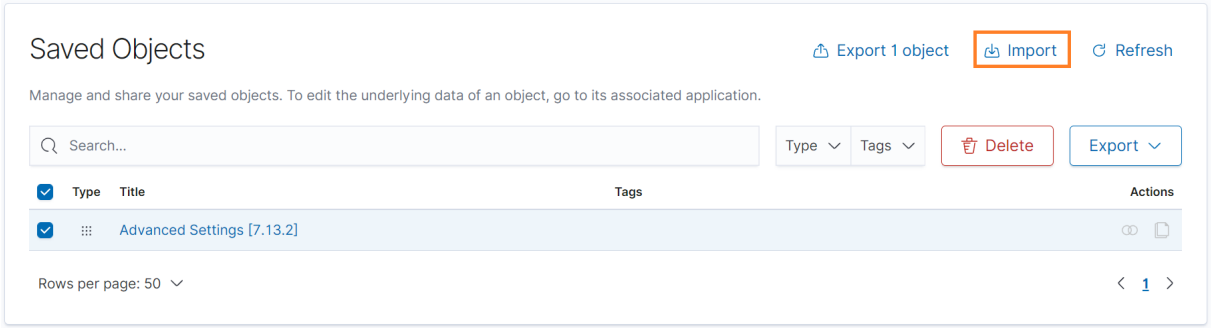
You'll be given an overview of the created template. A blue marked `M` in the `Content` section indicates that mappings are available.



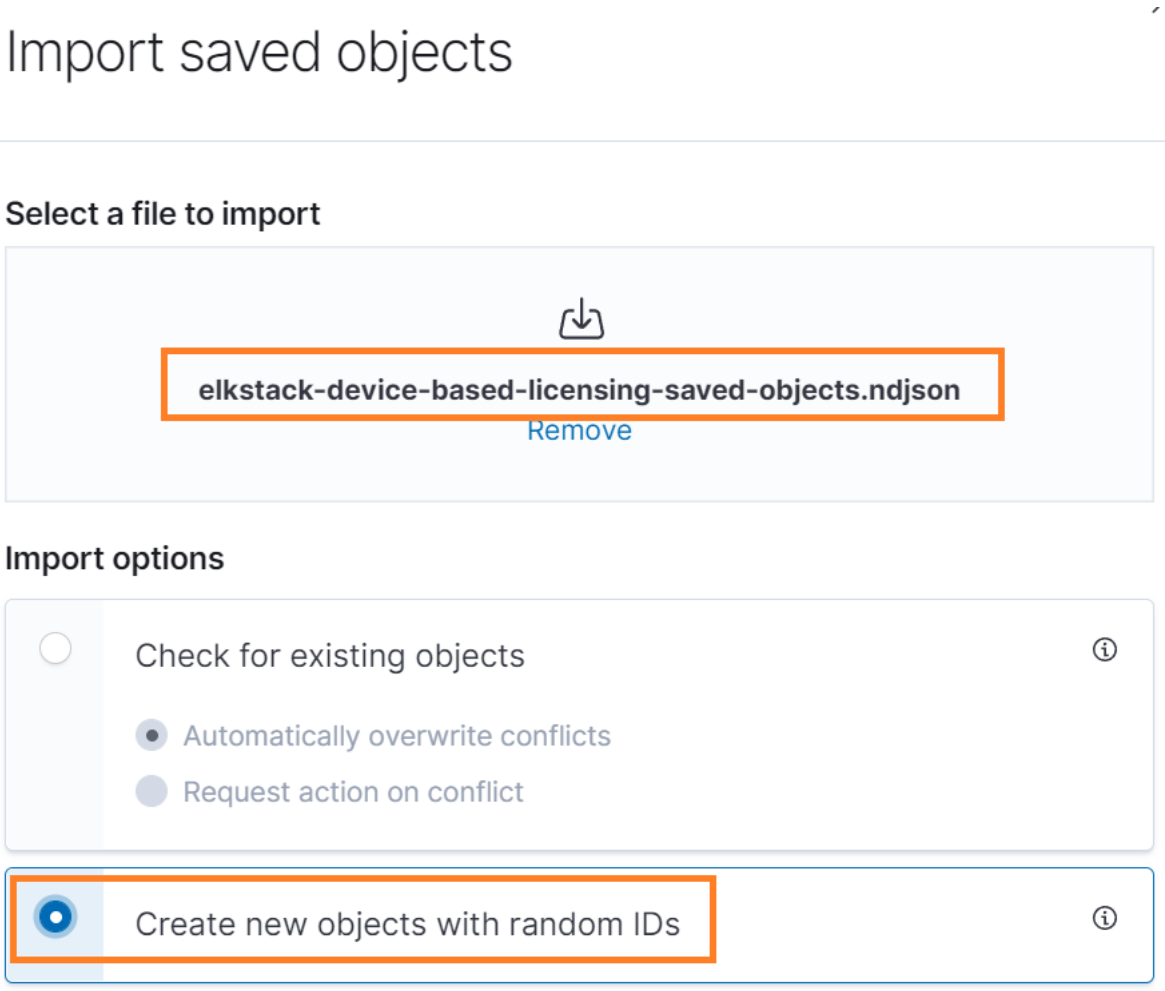
Step 2.2: Import Saved Objects to ELK Stack

All other parts of the report are to be imported as `Saved Objects`. The Saved Objects consist of Index Patterns with Scripted Fields, Visualizations and Dashboards.

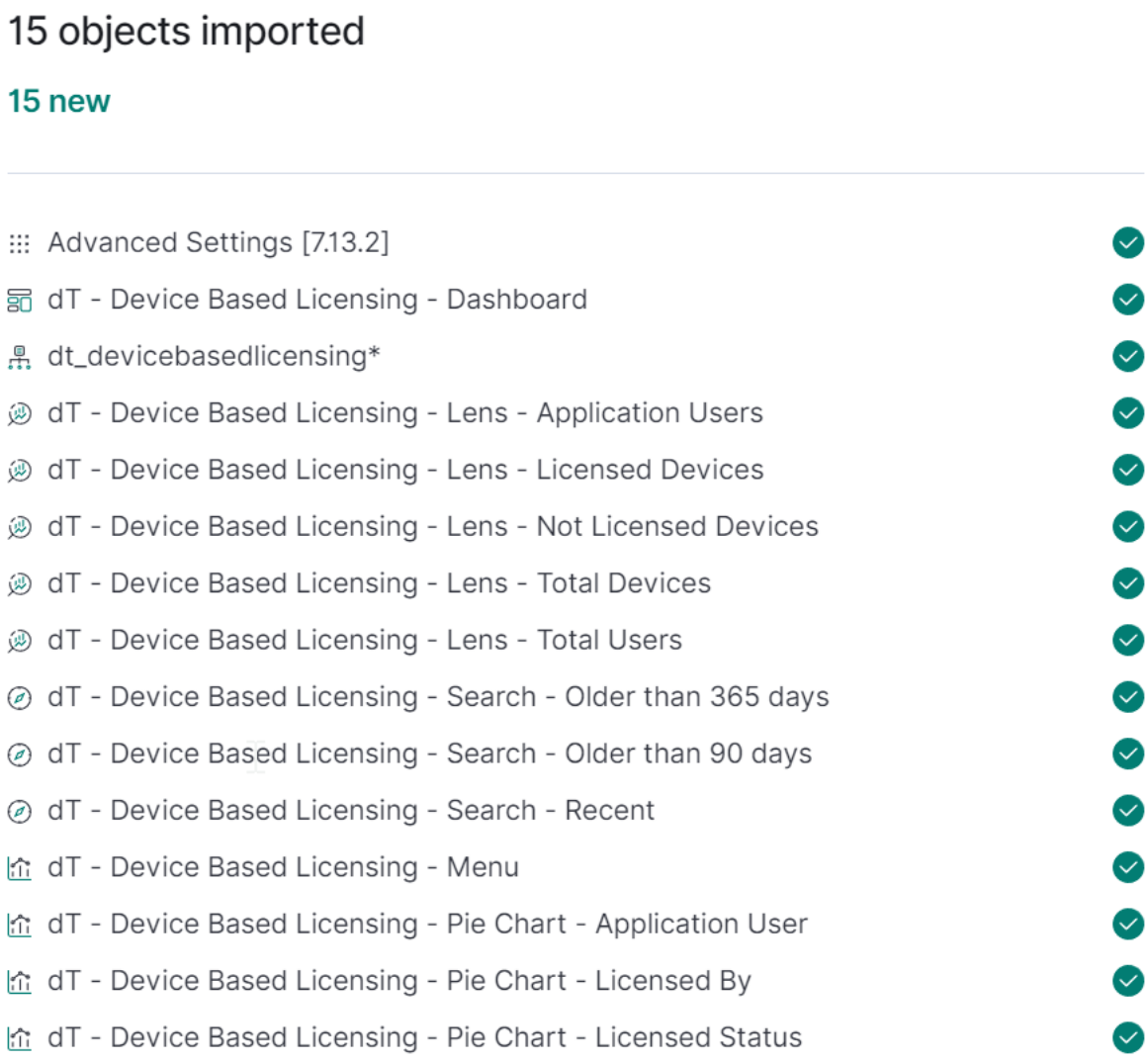
- Navigate to `Menu\Stack Management\Saved Objects` in your ELK Stack management console.
- Click `Import`.



- Select the file `elkstack-device-based-licensing-saved-objects.ndjson`.
- Check `Create new objects with random IDs` to make sure no existing objects are altered.



After importing, an overview of the imported objects will be displayed.

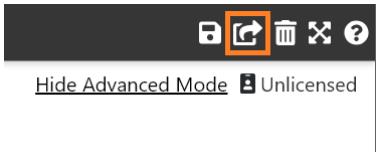


Step 2.3: Configuring deviceTRUST

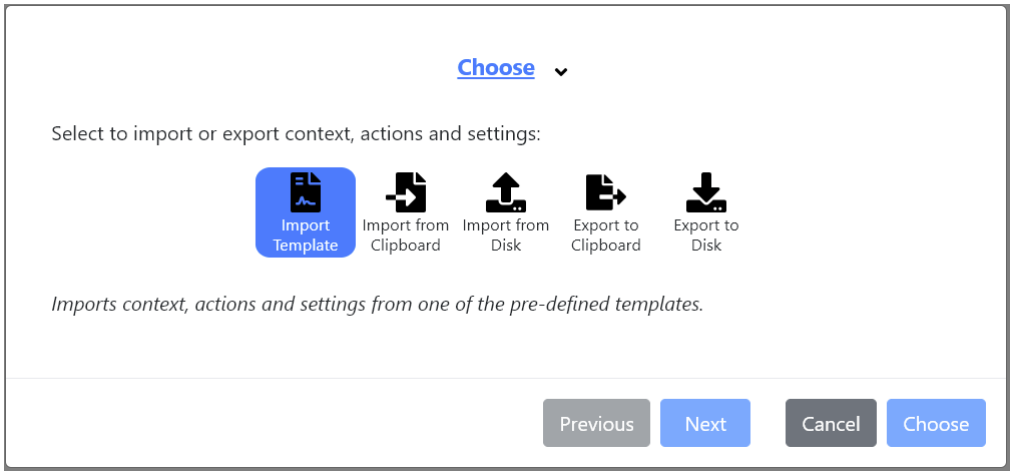
After the Index Mapping has been created and the Saved Objects are included, the ELK Stack is prepared for storing, sorting, and displaying your data.

The final step is to create the deviceTRUST configuration that will make sure all the required data is provided.

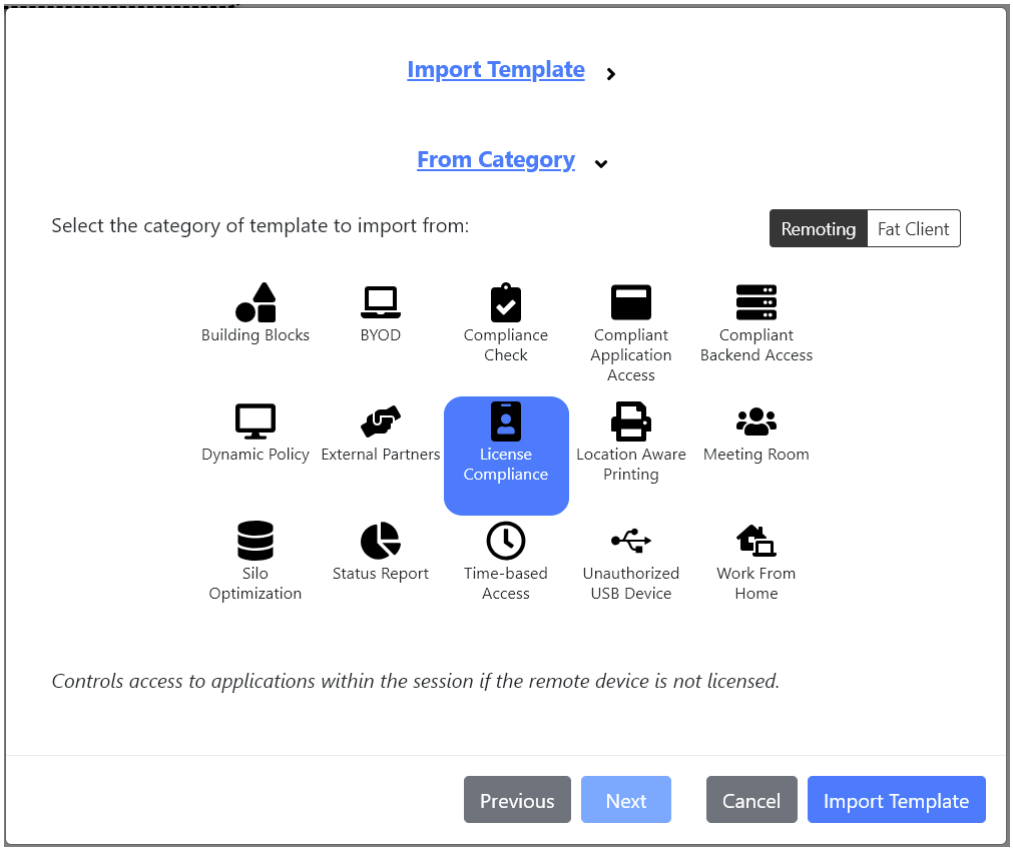
- Open the `deviceTRUST Console`.
- Click `Sharing` in the top right menu. You may need to click `Show Advanced View` if this button is not visible.



- Select `Import Template`.

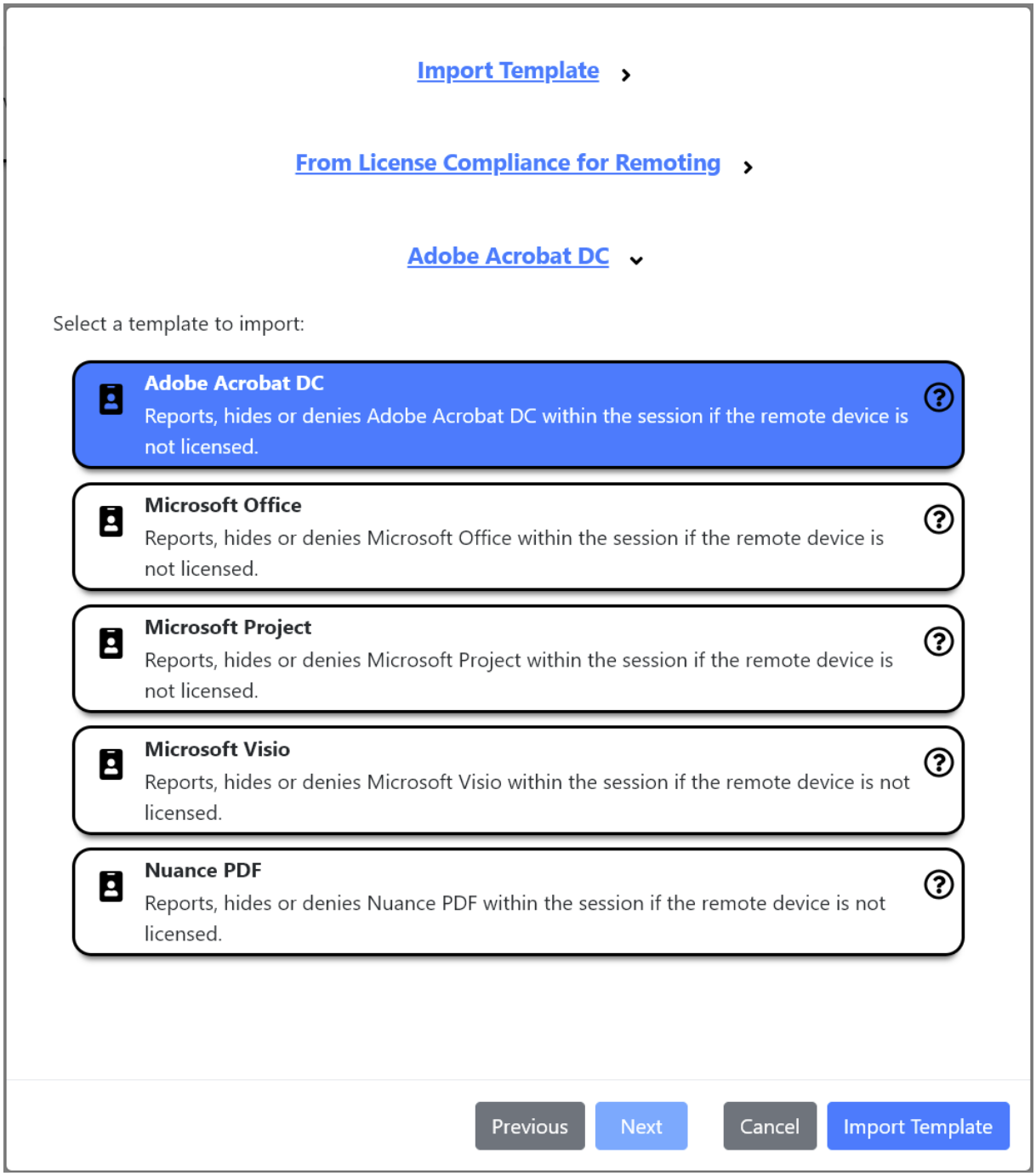


- For `Device-Based Licensing`, the template category `License Compliance` is used.



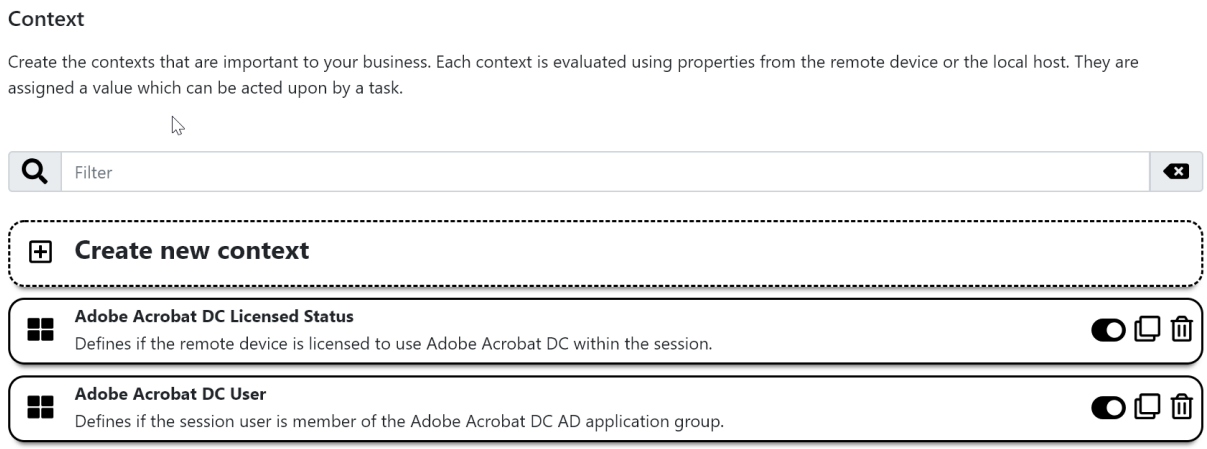
This category contains templates for several software products. This example uses `Acrobat DC`, but

can easily be customised for other applications.



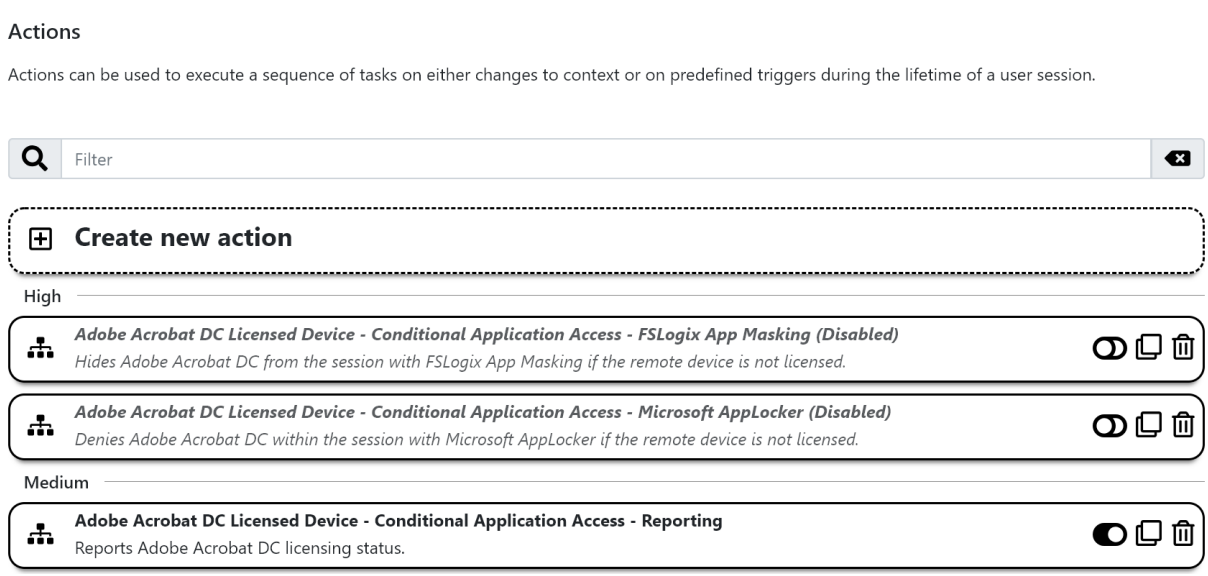
Two contexts are included:

- `Adobe Acrobat DC Licensed Status` to evaluate the device's license status.
- `Adobe Acrobar DC User` to define if the accessing user shall or shall not be using the software.

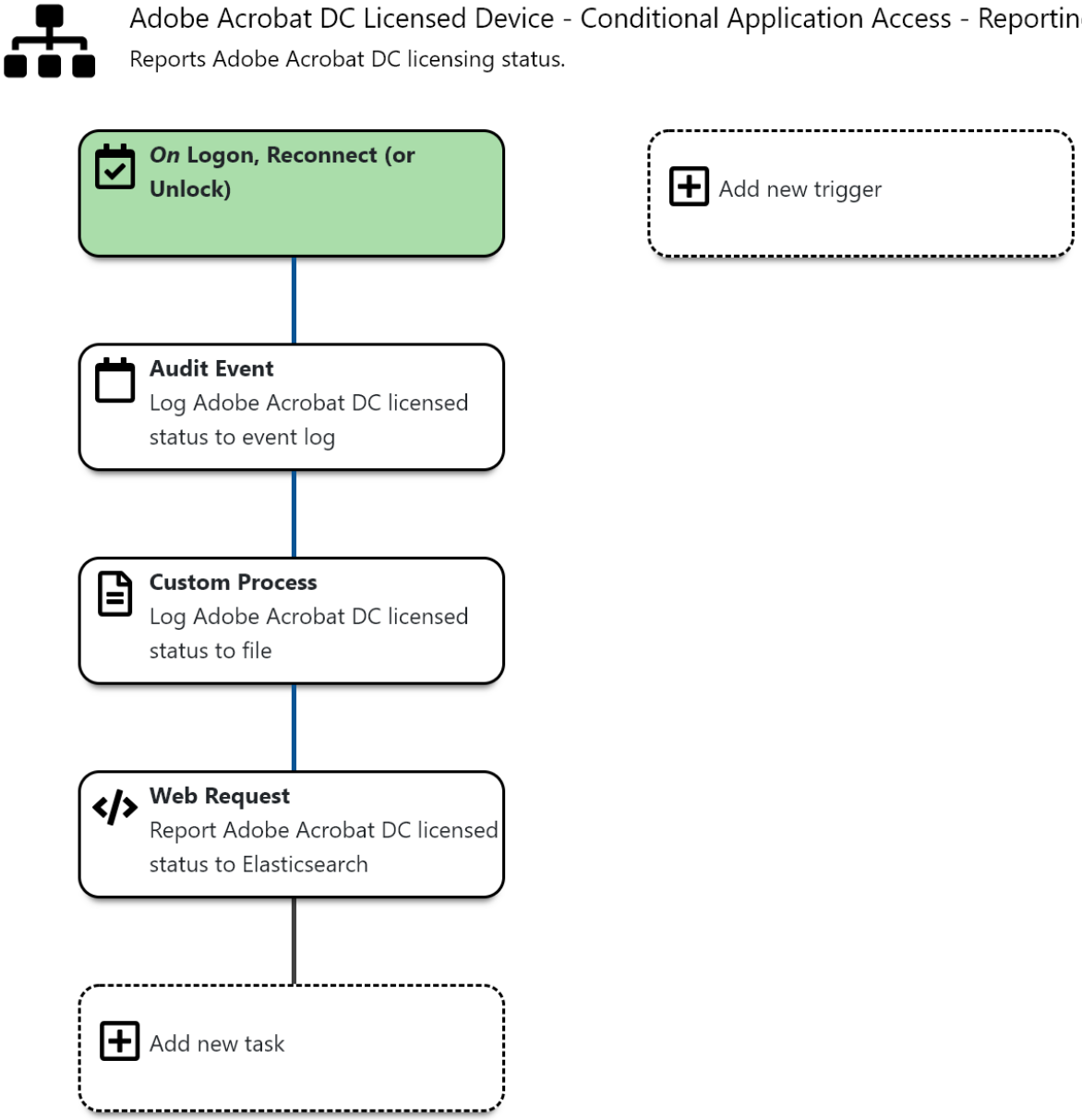


Three actions are included:

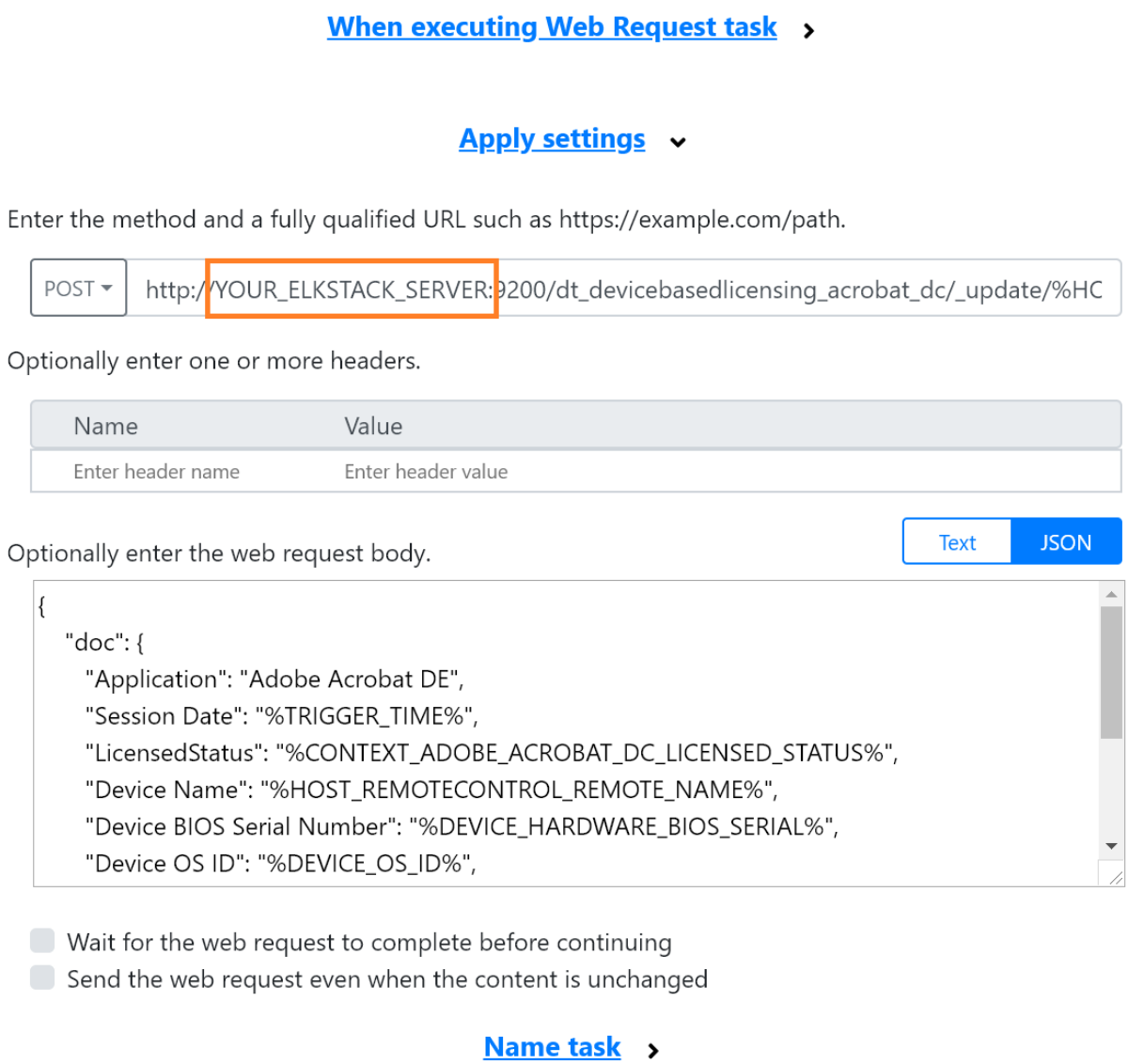
- `Adobe Acrobat DC Licensed Device - Conditional Application Access - FSLogix App Masking` is used for controlling access to the software using FSLogix App Masking. This action can be ignored or removed for now.
- `Adobe Acrobat DC Licensed Device - Conditional Application Access - Microsoft AppLocker` is used for controlling access to the software using Microsoft Applocker. This action can be ignored or removed for now.
- `Adobe Acrobat DC Licensed Device - Conditional Application Access - Reporting` is the only action required for reporting.



- The action contains multiple ways to store the data. Sending data to ELK Stack is configured by using a `Web Request` task. The `Audit Event`, `Custom Process`, as well as the `Web Request` Tasks for Splunk and Graylog can be deleted, as we are looking at ELK Stack here.

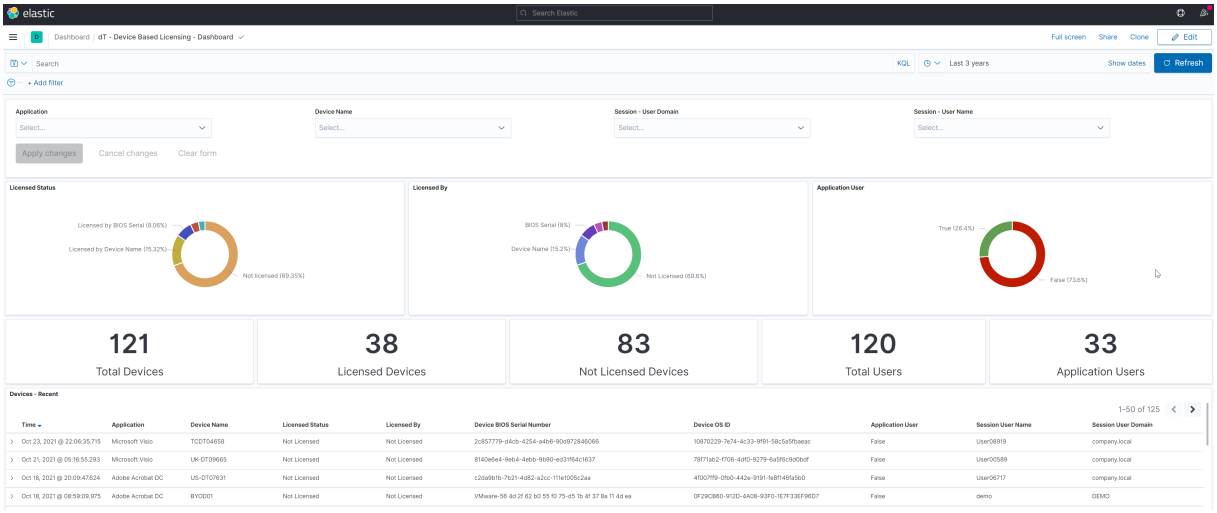


- The Web Request task must be edited to suit your environment. If you use a basic setup without SSL or authorization, adding `your server's fqdn` is the only required configuration change.



After the index template has been created, the saved objects are imported and the agent-side has been configured, the use case `Device-Based Licensing` has been implemented successfully.

deviceTRUST now sends status data about the application usage and the required hardware information to ELK Stack on every access to the remoting system. The data is presented in the created dashboards.



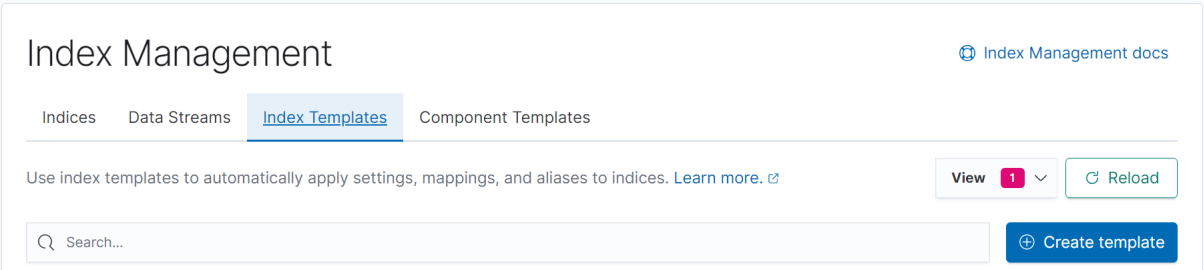
Step 3: The Status Report

This part of the guide relates to the `Status Report`. It lists all steps that are required to configure the use case on the agent-side, as well as on the ELK Stack side.

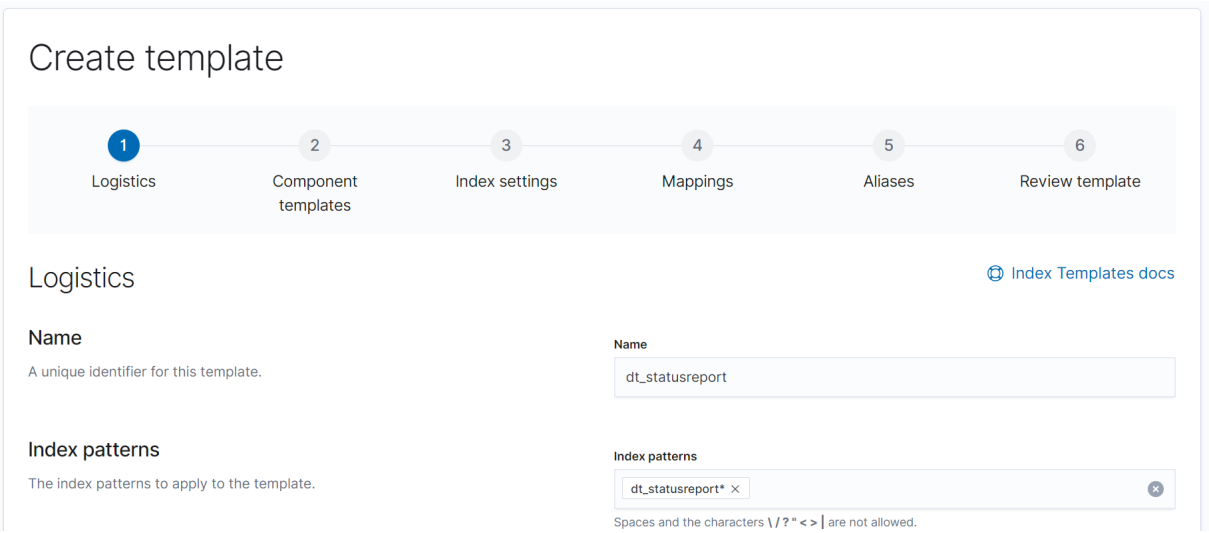
Step 3.1: Add Index Template to ELK Stack

The first step is to create an `Index Template`. Index Templates describe the data that is being sent to an index. They make sure, that every date is treated according to its type.

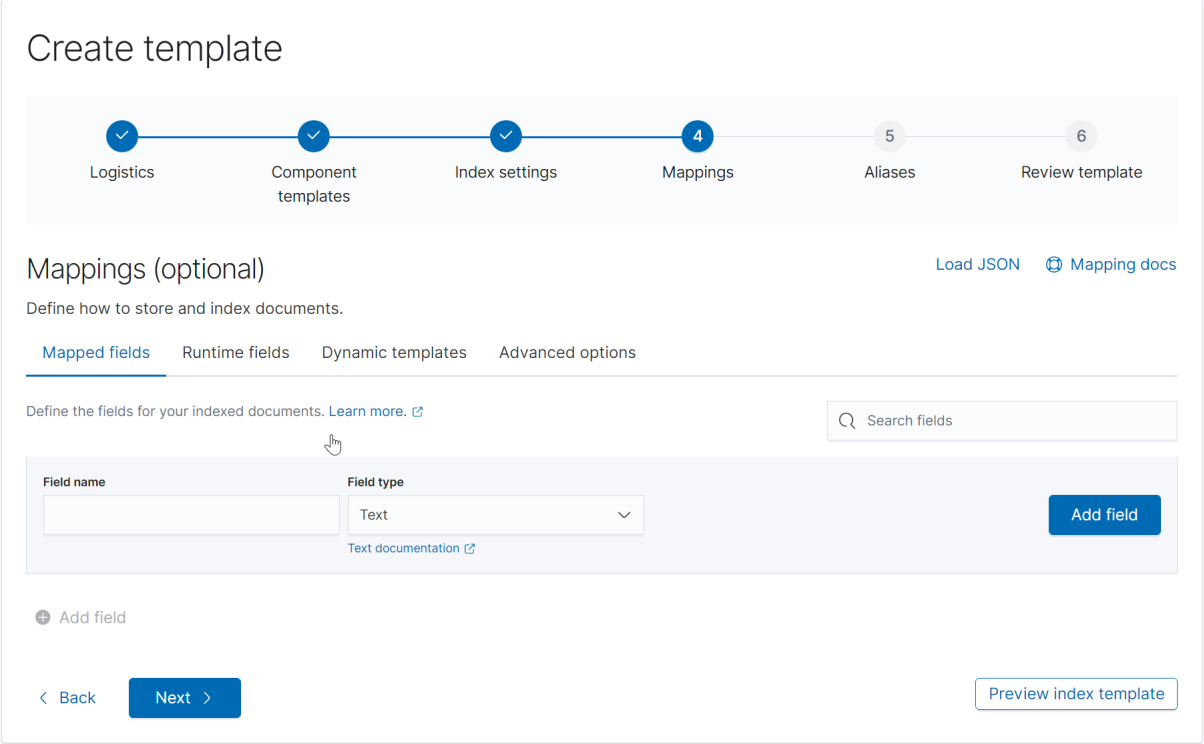
- Within the ELK Stack management console, navigate to `Menu\Stack Management\Index Management\Index Templates`.
- Click `Create Template`.



- Set `Name` to `dt_statusreport`.
- Set `Index Patterns` to `dt_statusreport*`.



- Skip all options until `Mappings`.
- Select `Load Json` to import the `elkstack-status-report-mappings.txt` that was downloaded in Step 1.1.



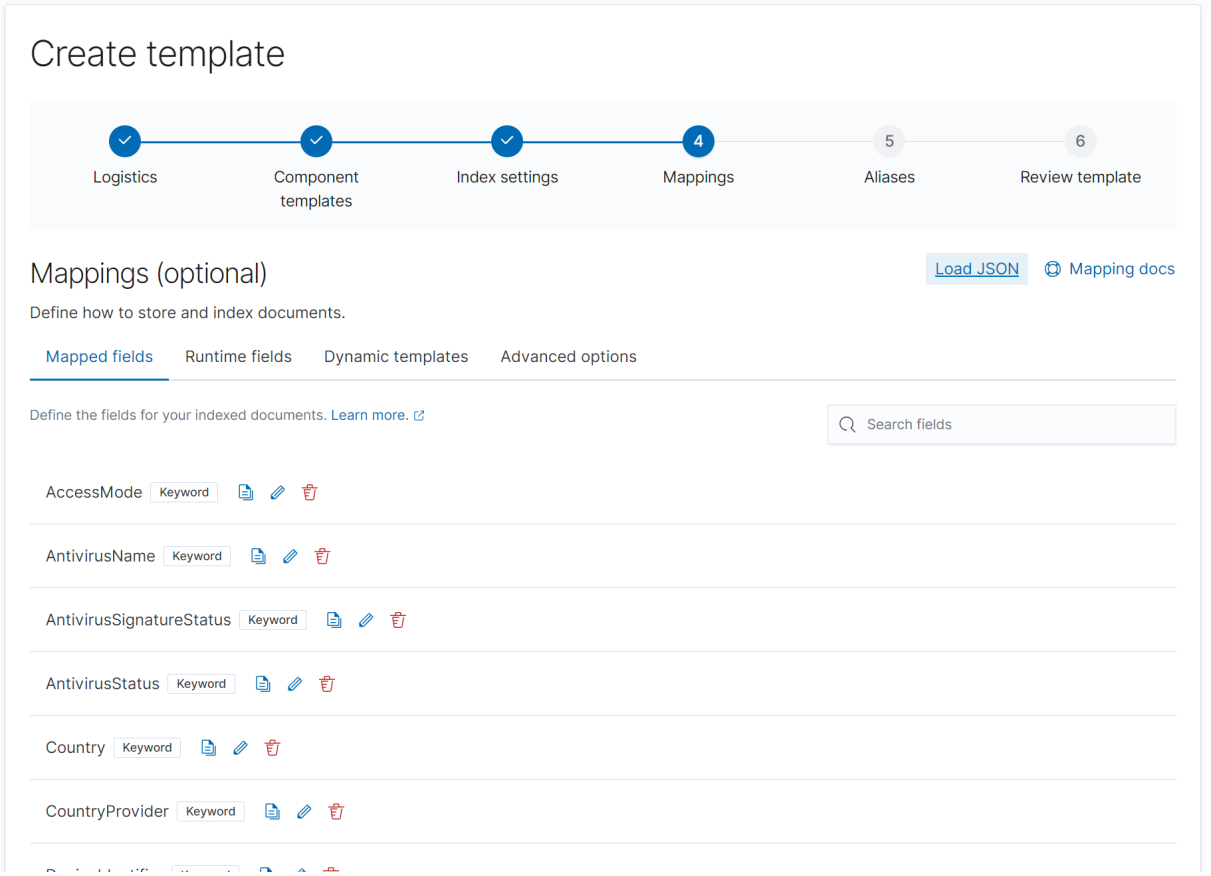
- Proceed with `Load and overwrite`.

Load JSON

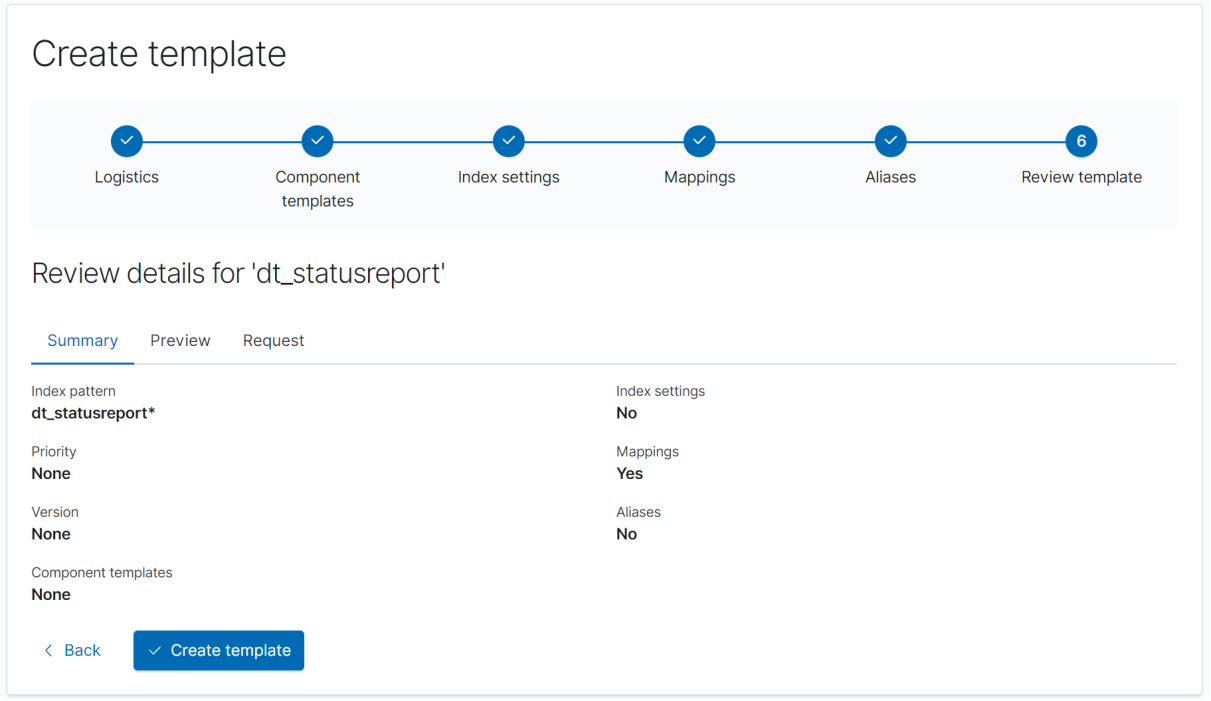
Provide a mappings object, for example, the object assigned to an index `mappings` property. This will overwrite existing mappings, dynamic templates, and options.

Mappings object

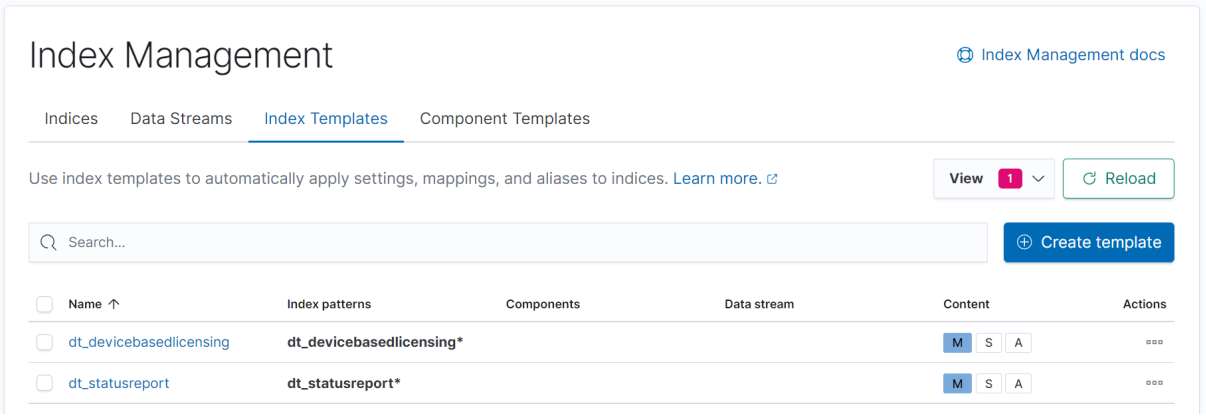
- The imported mappings are displayed. Please review them carefully. `Session Date`, `Anti-Virus Timestamp` and `Hardware BIOS Release Date` need to be recognized as type `Date` for the report to function properly.



- Skip all options until `Review Template`.
- Generate the template with `Create Template`.



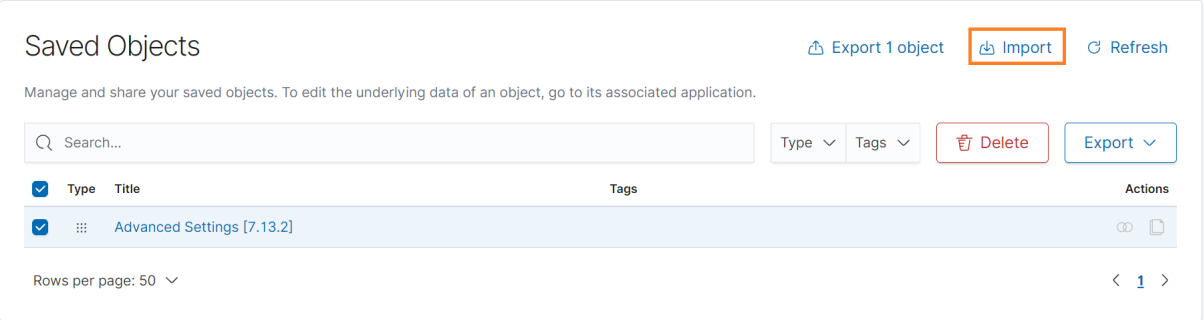
- You'll be given an overview of the created template. A blue marked `M` in the `Content` section indicates, that mappings are available.



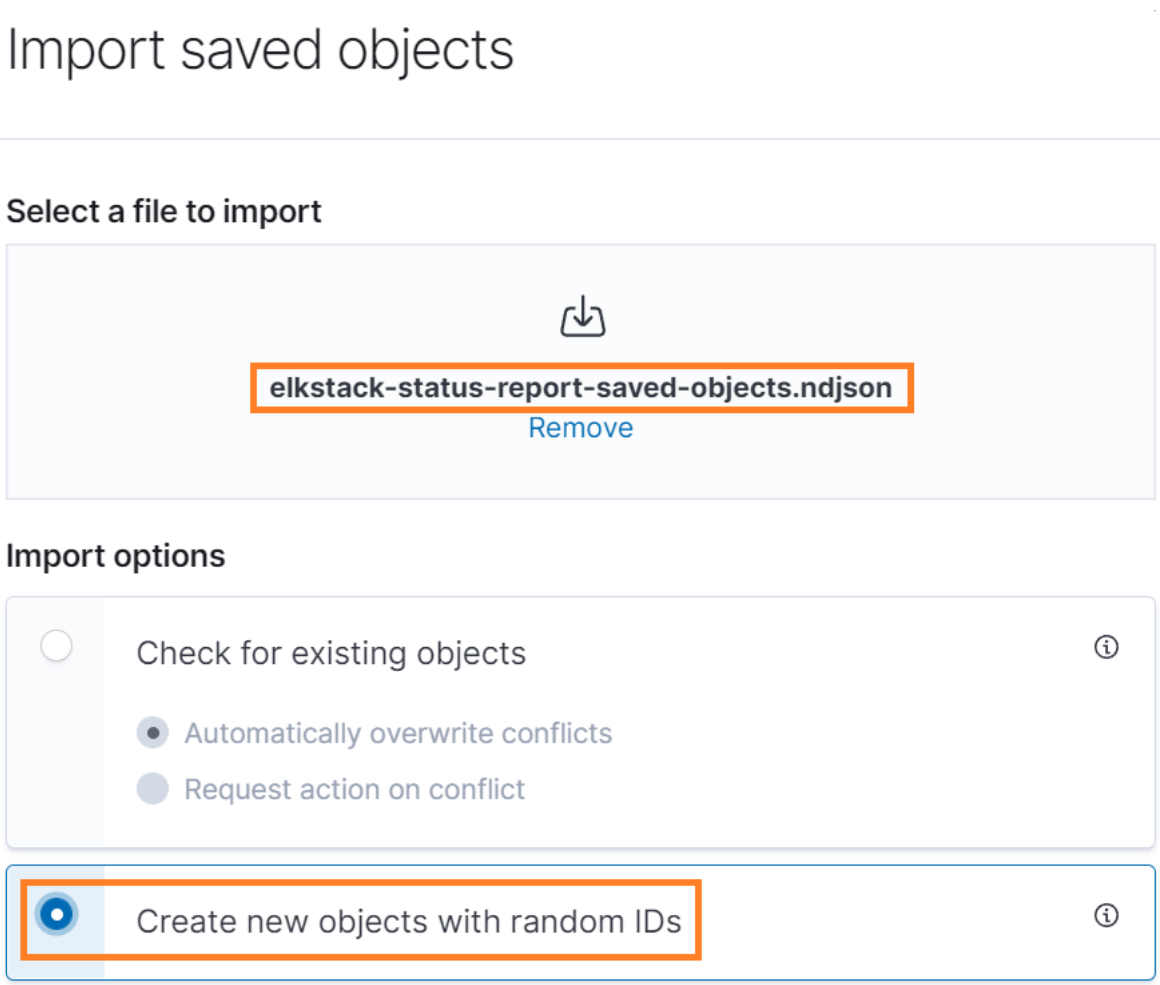
Step 3.2: Import Saved Objects to ELK Stack

All other parts of the report are to be imported as `Saved Objects`. The Saved Objects consist of Index Patterns with Scripted Fields, Visualizations and Dashboards.

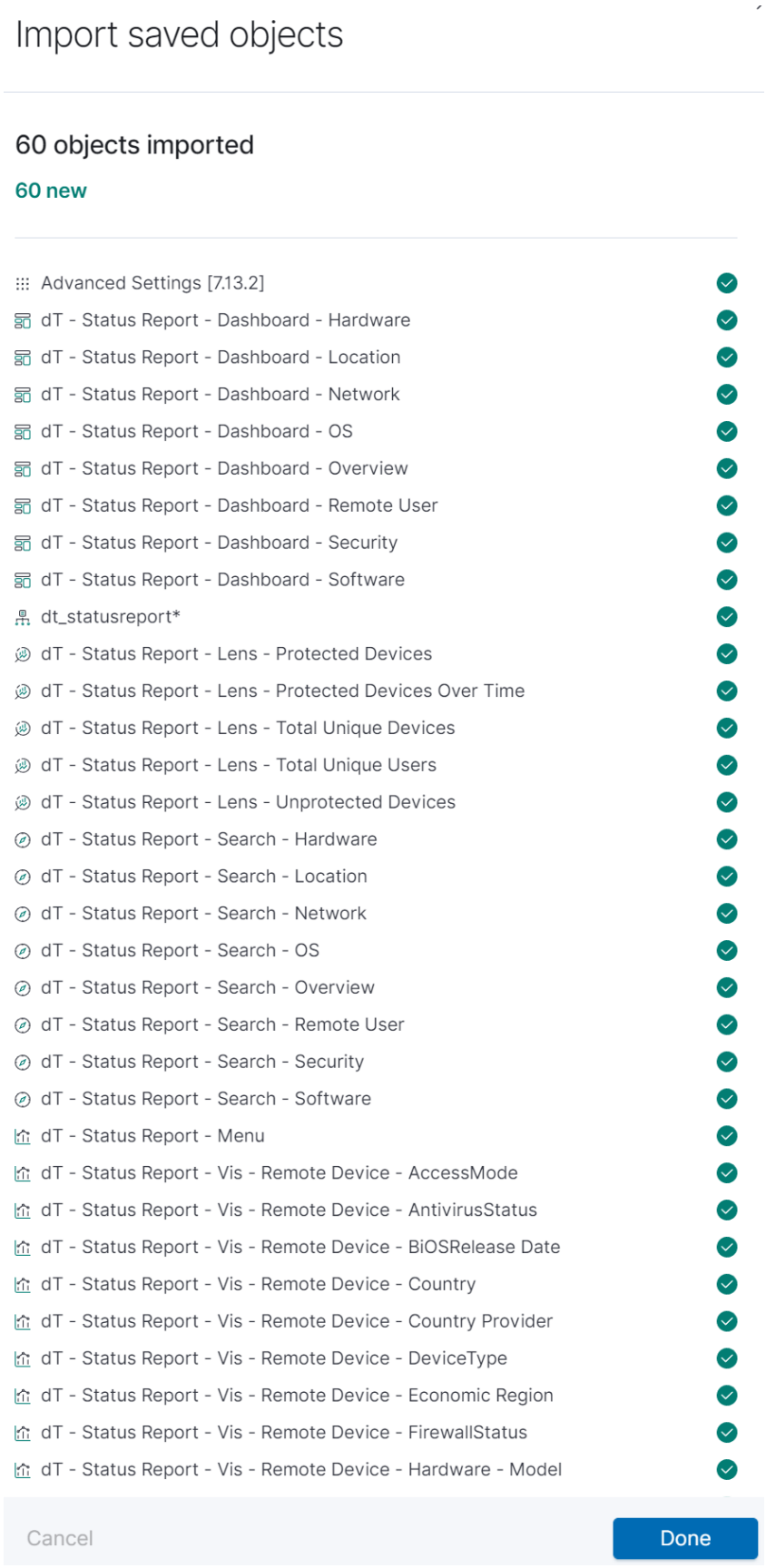
- Navigate to `Menu\Stack Management\Saved Objects` in your ELK Stack management console.
- Click `Import`.



- Select the file `elkstack-status-report-saved-objects.ndjson`.
- Check `Create **new** objects with random IDs` to make sure no existing objects are altered.



After importing, an overview of the imported objects will be displayed.

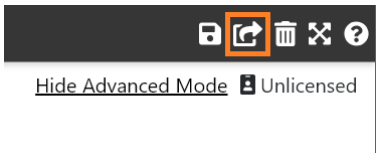


Step 3.3: Configuring deviceTRUST

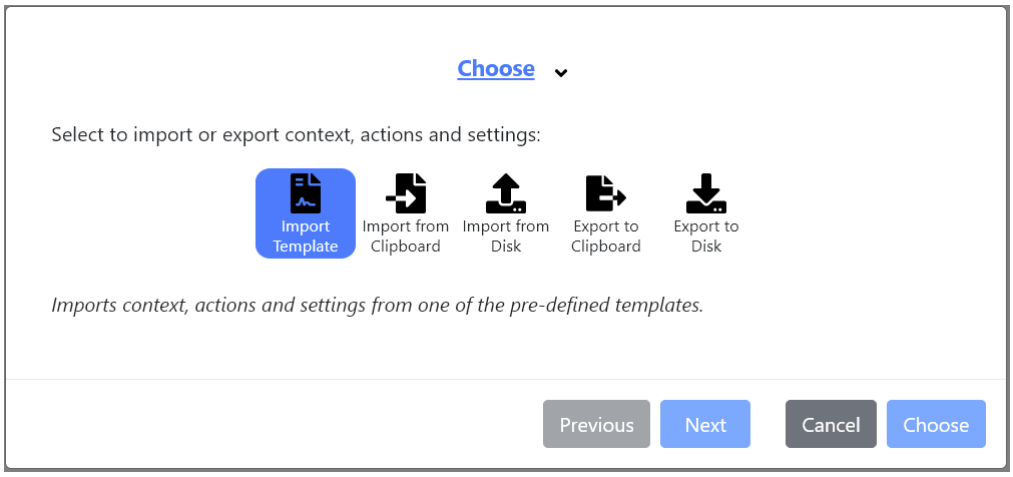
After the Index Mapping has been created, the Saved Objects are included the index has been edited and the agent-side has been configured, the ELK Stack is prepared for storing, sorting, and displaying your data.

The last step is to create the deviceTRUST configuration, that will make sure all required data is provided.

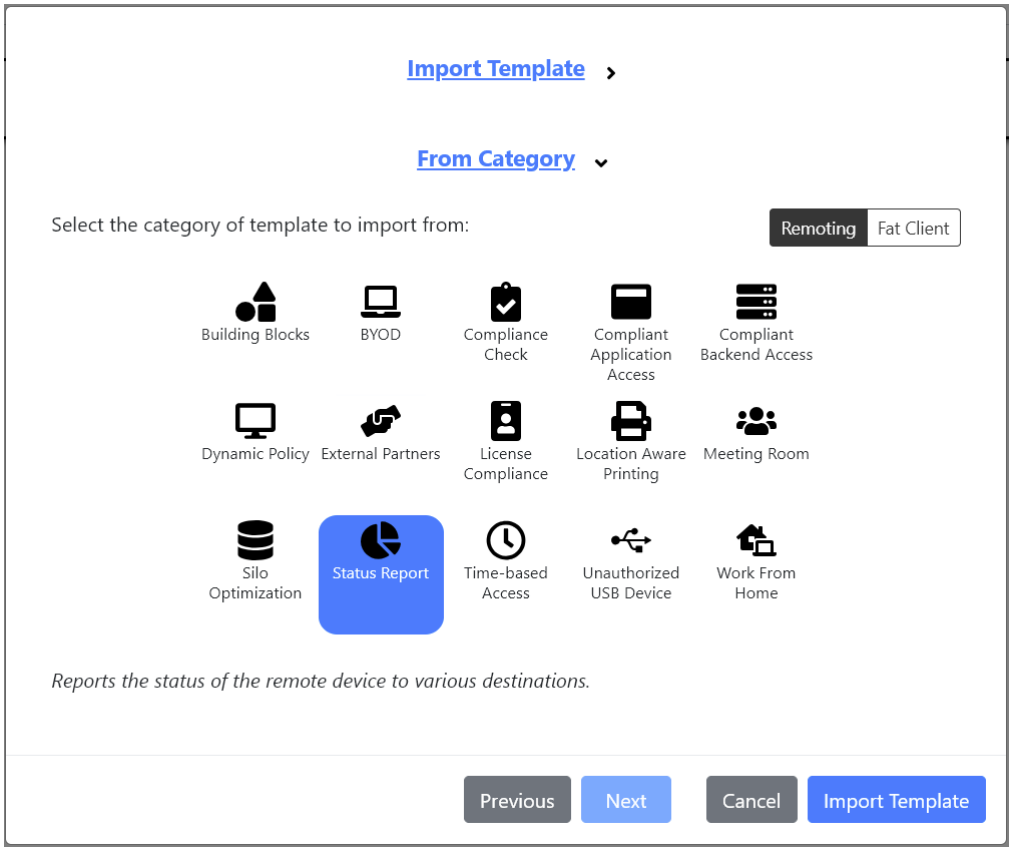
- Open the `deviceTRUST Console`.
- Click `Sharing` in the top right menu. You may need to click `Show Advanced View` if this button is not visible.



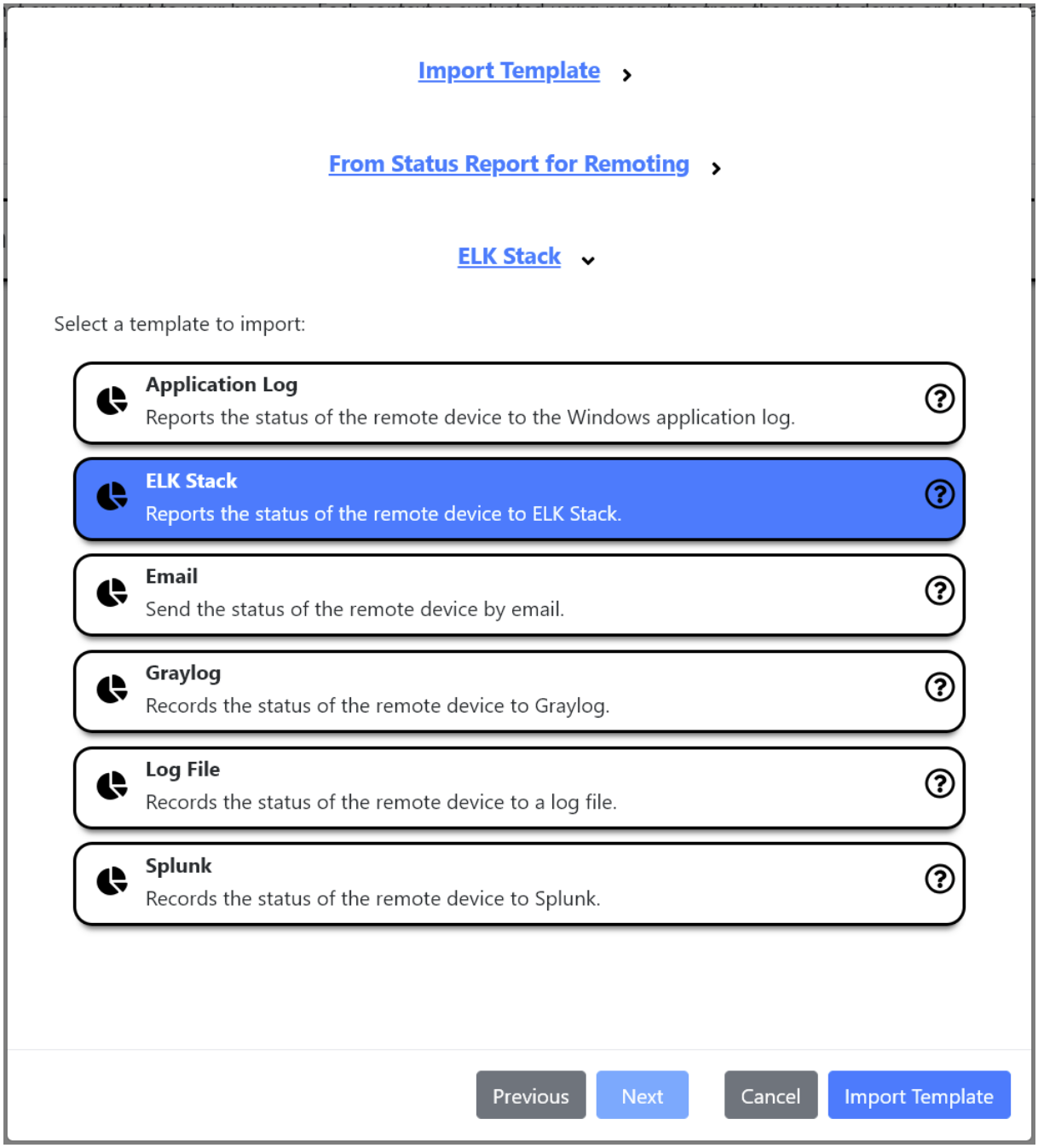
- Select `Import Template`.



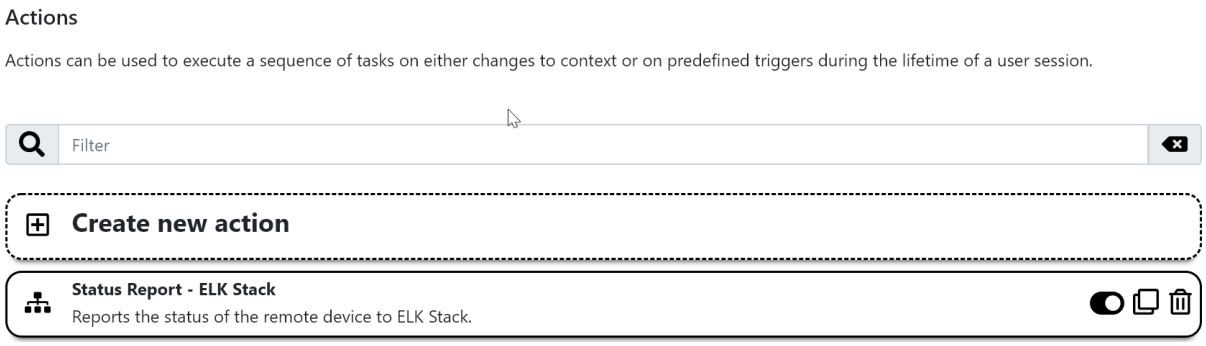
- For `Status Report`, the template category `Status Report` is used.



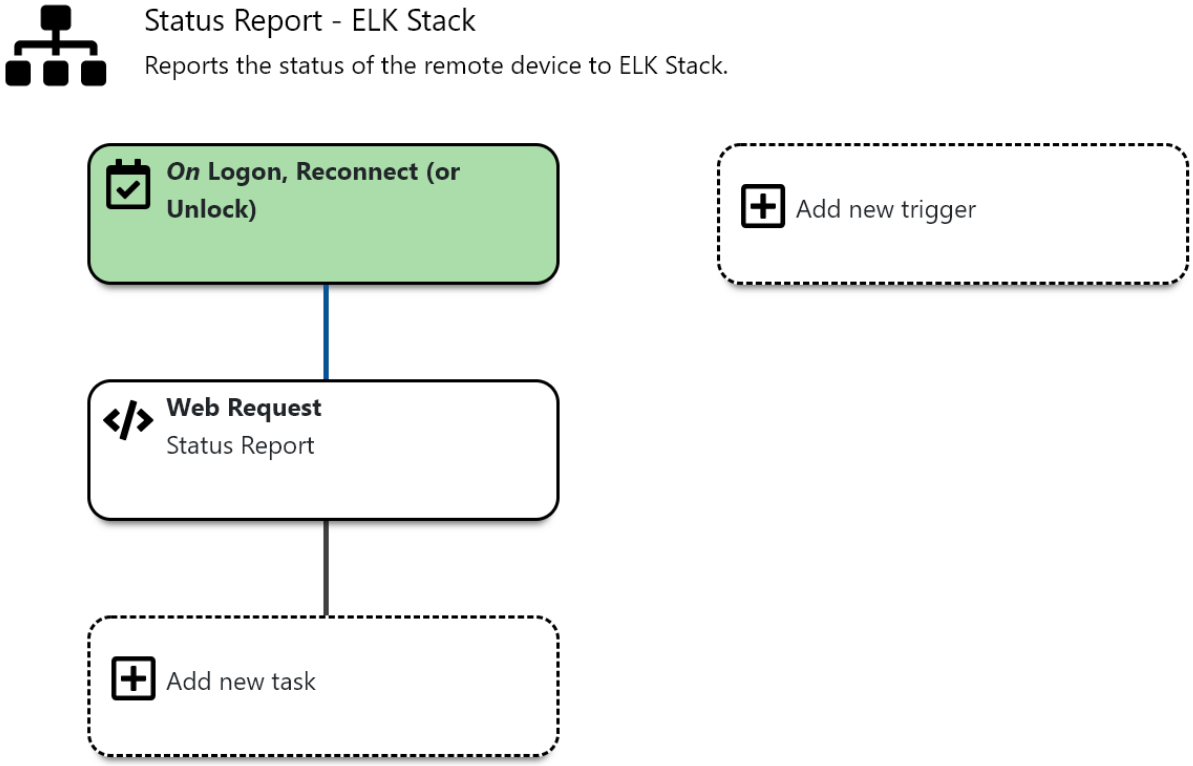
- This category contains templates for several ways of storing data. Choose `ELK Stack`.



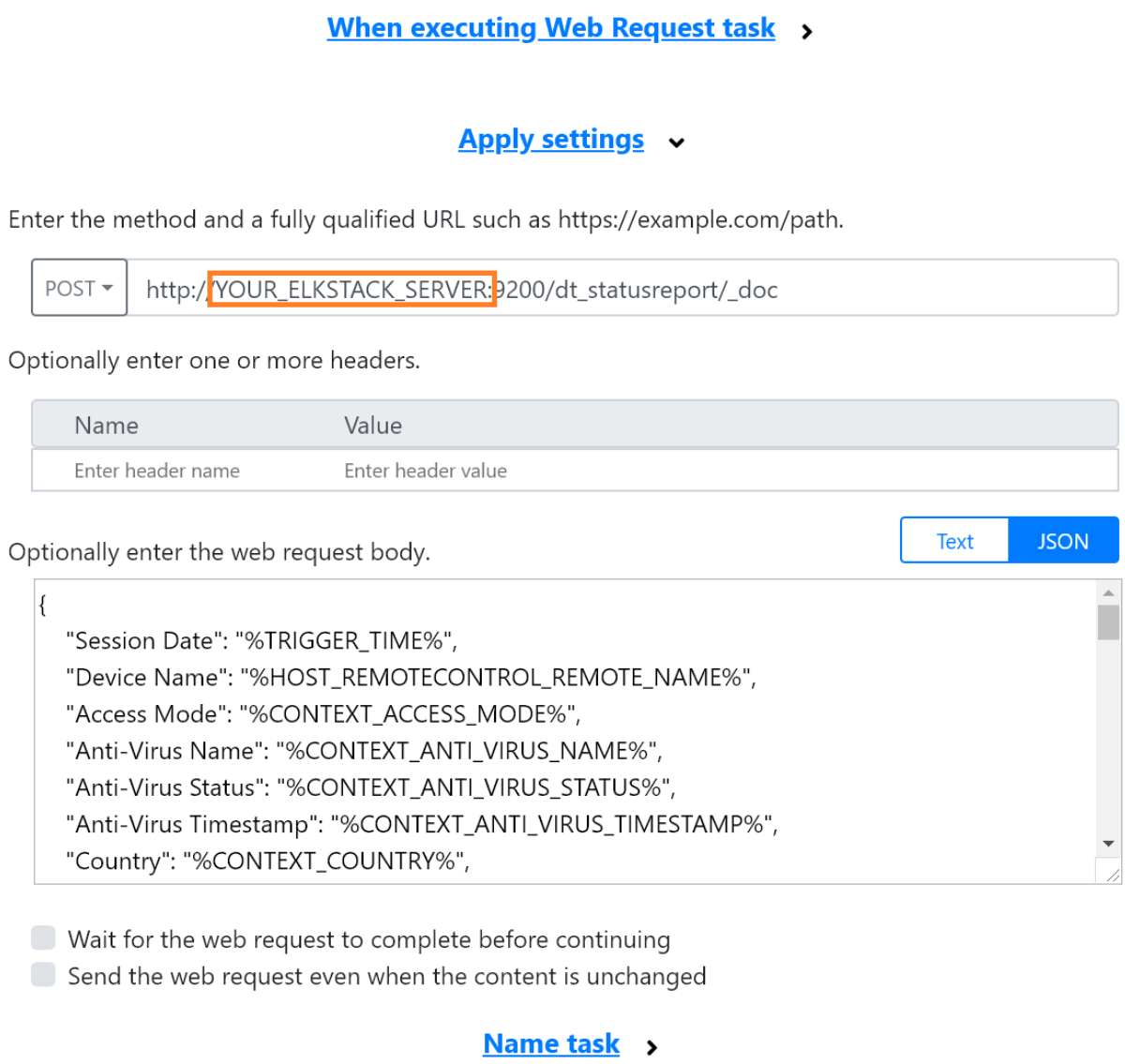
- The imported template consists of 50 contexts and one action.
- The Action `Status Report – ELK Stack` collects all relevant data and sends them over to the ELK Stack.



- Sending data to ELK Stack is configured by using a `Web Request` task.

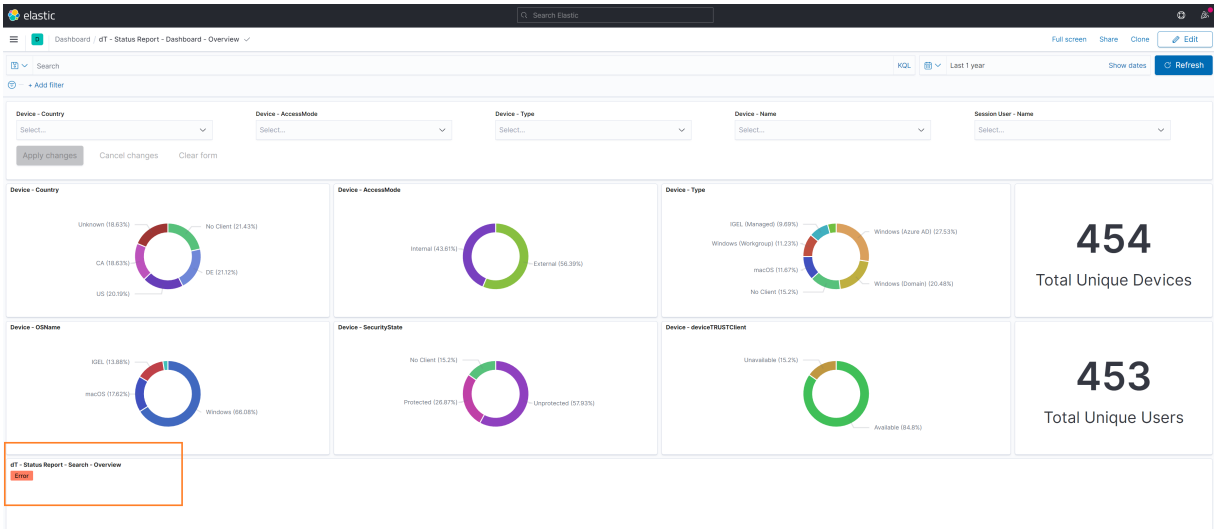


- The Web Request task must be edited to suit your environment. If you use a basic setup without SSL or authorization, simply adding your `server's fqdn` will do.



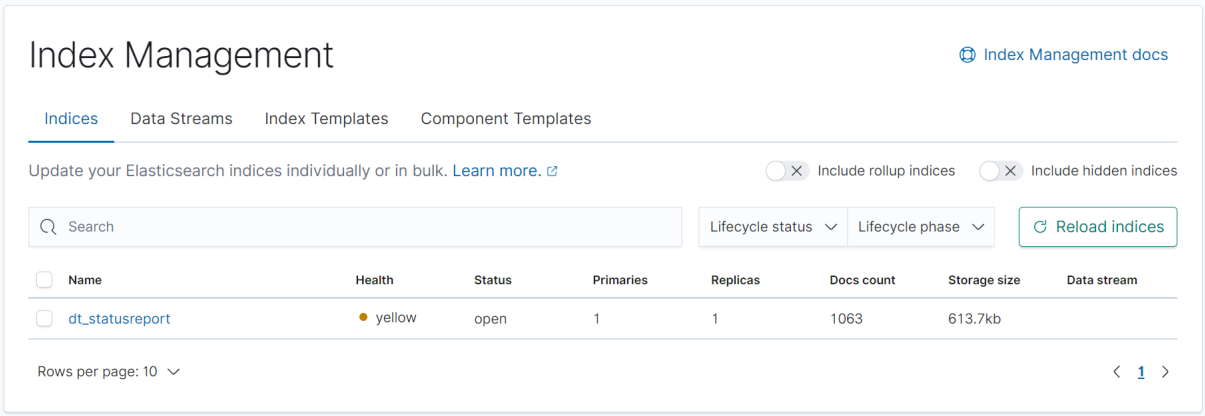
Step 3.4: Edit index settings

For the Status Report Dashboards to work properly, a configuration needs to be made at the index level: In a basic setting, ELK Stack allows to use 25 "calculated fields"per index. For the Status Report Dashboard, 48 calculated fields are used. Thus, the `allowed number of calculated fields` needs to be set to a higher value.

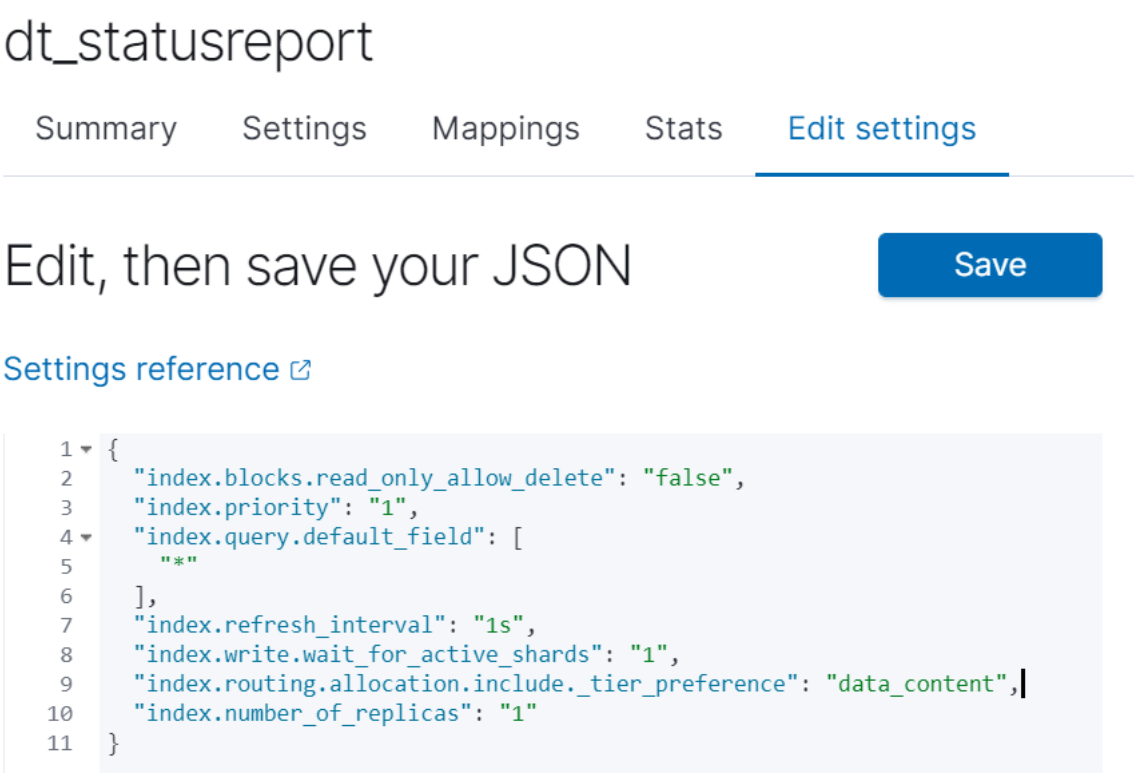


You need to send data to ELK Stack first. Sending data will create the index with basic settings. It can then be edited.

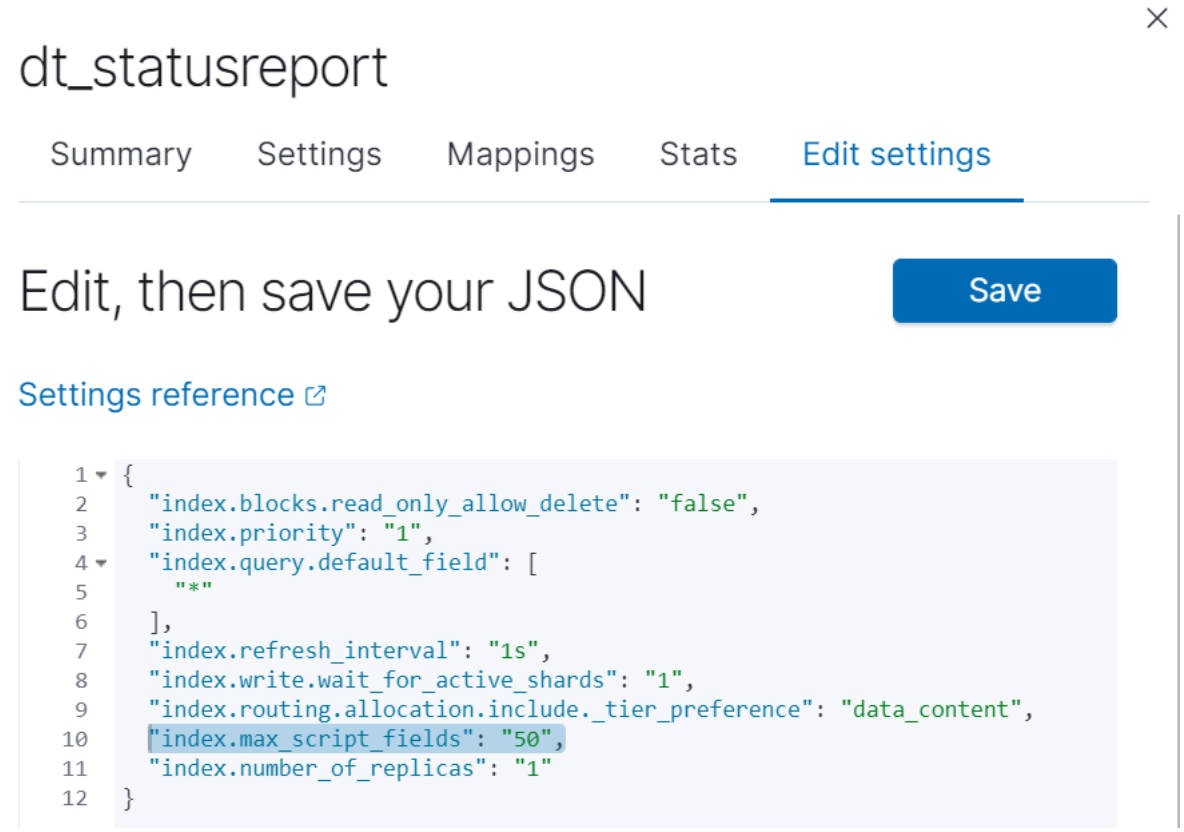
- After sending your first data, you will find the index `dt_statusreport` has been created in the Index Management Menu.



- Select the Index and chose `Edit Settings`. You'll be presented a json configuration view.

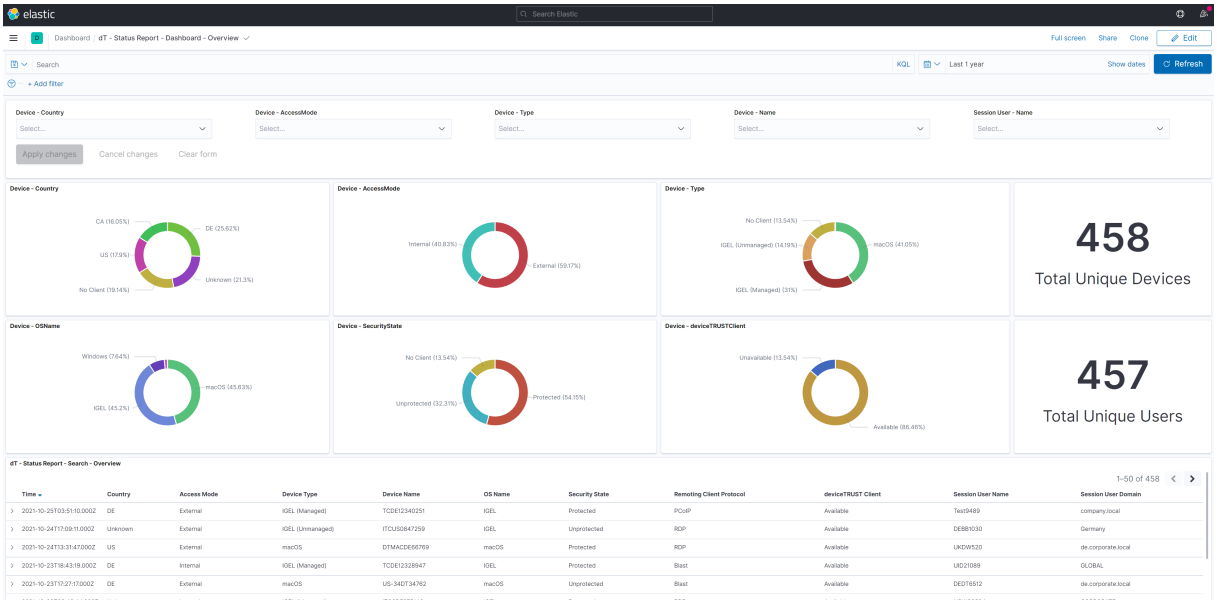


- Add `"index.max_script_fields": "50"` as a new line, making sure to keep the correct json formatting.
- Save your changes.



Your Dashboard will now be displayed without errors.

After the index template has been created and the saved objects are imported, the use case `Status Report` has been implemented successfully.
deviceTRUST now sends status data to ELK Stack on every access to the remoting system. The data is presented in the created dashboards.



Reference

TABLE OF CONTENTS

Properties

- Access Point Properties - Describes the available Wi-Fi access points.
- Browser Properties - Describes installed internet browsers.
- Cellular Properties - Describes the active cellular capabilities of an endpoint.
- Certificate Properties - Describes the private certificates available within the users certificate store.
- ChromeOS Properties - Provides properties unique to a Google ChromeOS device.
- Custom Properties - Provides a condition that can operate on any property, including custom properties created using 'dtcmd.exe'.
- deviceTRUST Properties - Provides information about whether a connection to a deviceTRUST Client Extension has been established, the version of the deviceTRUST software and the status of the deviceTRUST license.
- Display Properties - Describes the displays available to the user session.
- Domain Properties - Describes the domain membership of the endpoint.
- eLux Properties - Provides properties unique to a Unicon eLux device.
- Hardware Properties - Describes the hardware and its capabilities.
- IGEL Properties - Provides properties unique to an IGEL device.
- Input Properties - Describes the input devices available to the user session.
- iOS Properties - Provides properties unique to an Apple iOS endpoint.
- Location Properties - Describes the geographical location of the endpoint.
- Logical Disk Properties - Describes the logical disks available to the user.
- macOS Properties - Describes properties unique to an Apple macOS endpoint.
- macOS Firewall Properties - Provides real-time properties describing the state of the macOS Firewall.
- macOS Update Properties - Describes the status of macOS Software Update settings and updates.
- Mapped Drive Properties - Describes the mapped drives available within a user session.
- Multihop Properties - Describes the number of hops taken by the user over deviceTRUST connected sessions.
- Name Properties - Identifies the endpoint.

- Network Properties - Describes the network adapters and their bound network addresses.
- OS Properties - Provides information about the operating system installed on the endpoint.
- Password Policy Properties - Describes the password policy of the logged in user.
- Performance Properties - Describes the performance of the remoting protocol.
- Power Properties - Describes the power profile of the endpoint.
- Printer Properties - Describes the printers available to the user session.
- Region Properties - Describes the regional information of the user session.
- Remote Control Properties - Determines whether the user session is being remote controlled and provides information about the remote controlling endpoint.
- Remoting Client Properties - Provides properties about the remoting client used to remote control the user session.
- Screen Saver Properties - Describes the screen saver applied to the user session.
- Security Product Properties - Provides real-time properties describing the state of the installed Antivirus, Antispyware and Firewall security products.
- Session Properties - Provides information describing the user's logon session.
- Smartcard Reader Properties - Describes the connected smart-card readers available to the user.
- User Properties - Identifies the logged in user.
- WHOIS Properties - Provides the results of a WHOIS lookup of the endpoint.
- Windows Properties - Provides real-time properties unique to a Microsoft Windows device.
- Windows Defender Properties - Provides real-time properties describing the state of Microsoft Windows Defender Antivirus.
- Windows Firewall Properties - Provides real-time properties describing the state of the Microsoft Windows Firewall.
- Windows Registry Properties - Provides access to Windows Registry entries.
- Windows Update Properties - Describes the status of Microsoft

Agent Reference

TABLE OF CONTENTS

- Product Events - Product Events

Console Reference

TABLE OF CONTENTS

- Actions - Actions Reference

- Settings - Settings Reference

Troubleshooting

If your deviceTRUST installation or configuration does not work as expected, then the Knowledge Base is a useful resource for common problems and resolutions. The following sections detail some useful knowledge base articles depending upon your deployment scenario:

Scenario: Remote

In remote scenarios, deviceTRUST® transports the context information from the user's remote device to the virtual session where the configuration is enforced. Please check the following knowledge base articles:

- Step 1: Make sure that you have a valid license
- Step 2: Check that your contextual security policy has been saved and deployed
- Step 3: Check that the user is managed by deviceTRUST
- Step 4: Check that the deviceTRUST Client Extension is installed on the remote device
- Step 5: Check that Citrix Virtual Channel Security is configured (Citrix Only)
- Step 6: Check that your contexts are correctly defined
- Step 7: Exclude specific users from the deviceTRUST policy
- Step 8: Check that the deviceTRUST Agent service is running
- Step 9: Check that you are using the latest deviceTRUST version

Scenario: Local

In local scenarios, deviceTRUST collects context information and executes actions locally. Please check the following knowledge base articles:

- Step 1: Make sure that you have a valid license
- Step 2: Check that your contextual security policy has been saved and deployed
- Step 3: Check that the user is managed by deviceTRUST
- Step 4: Check that your contexts are correctly defined
- Step 5: Exclude specific users from the deviceTRUST policy
- Step 6: Check that the deviceTRUST Agent service is running
- Step 7: Check that you are using the latest deviceTRUST version

Open a support ticket with us

Additional articles may be found by searching the Knowledge Base. However, if you are still experiencing difficulties please raise a ticket with the Citrix Support Portal at https://support.citrix.com.

Releases

TABLE OF CONTENTS

TABLE OF CONTENTS

June 20, 2025

Ubuntu Client Extension 21.1.300

This release includes enhancements and bug fixes to the deviceTRUST® Client Extension for Ubuntu Desktop.

Support for VMware Horizon View

The deviceTRUST Ubuntu Client now includes support for VMware Horizon View using the Blast, PCoIP or RDP protocols. The full list of supported technologies can be found at Operating System Compatibility

Custom Properties for Linux devices

The Custom Properties Settings now supports remote Linux devices. This allows a custom script to be executed on the remote Linux device to return one or more properties beginning with REMOTE_CUSTOM by writing e.g. `REMOTE_CUSTOM_NAME=VALUE` to the output.

Minor enhancements

- Added VPN and Adapter properties to the WHOIS results.

Bug Fixes

- Fixed a potential crash during the logoff or disconnect from a session.

Compatibility

There are no compatibility concerns with this release of the deviceTRUST Ubuntu Client.

macOS Client Extension 21.1.300

This release includes enhancements and bug fixes to the deviceTRUST® Client Extension for macOS.

Native support for Apple silicon

We've integrated native support for Apple silicon processors, such as the Apple M1 and M2, alongside our existing support for Intel processors into a single universal installation package. This removes the previous requirement for Rosetta 2 translations. A remoting client that is also built with native Apple silicon support is required, and is therefore currently supported with the following clients:

- Citrix Workspace App for macOS v22.11.0.12 or later.
- FreeRDP v2.8.2 or later.
- VMware Horizon View v8.7.0 or later.

Bug Fixes

- Fixed an issue where Location properties were unavailable on macOS Ventura.
- Fixed an issue where Display properties contained unavailable Adapter and Monitor products and vendor properties on Apple silicon processors on macOS Ventura.
- Fixed a potential crash during shutdown of the remoting client.

Compatibility

There are no compatibility concerns with this release of the deviceTRUST macOS Client.

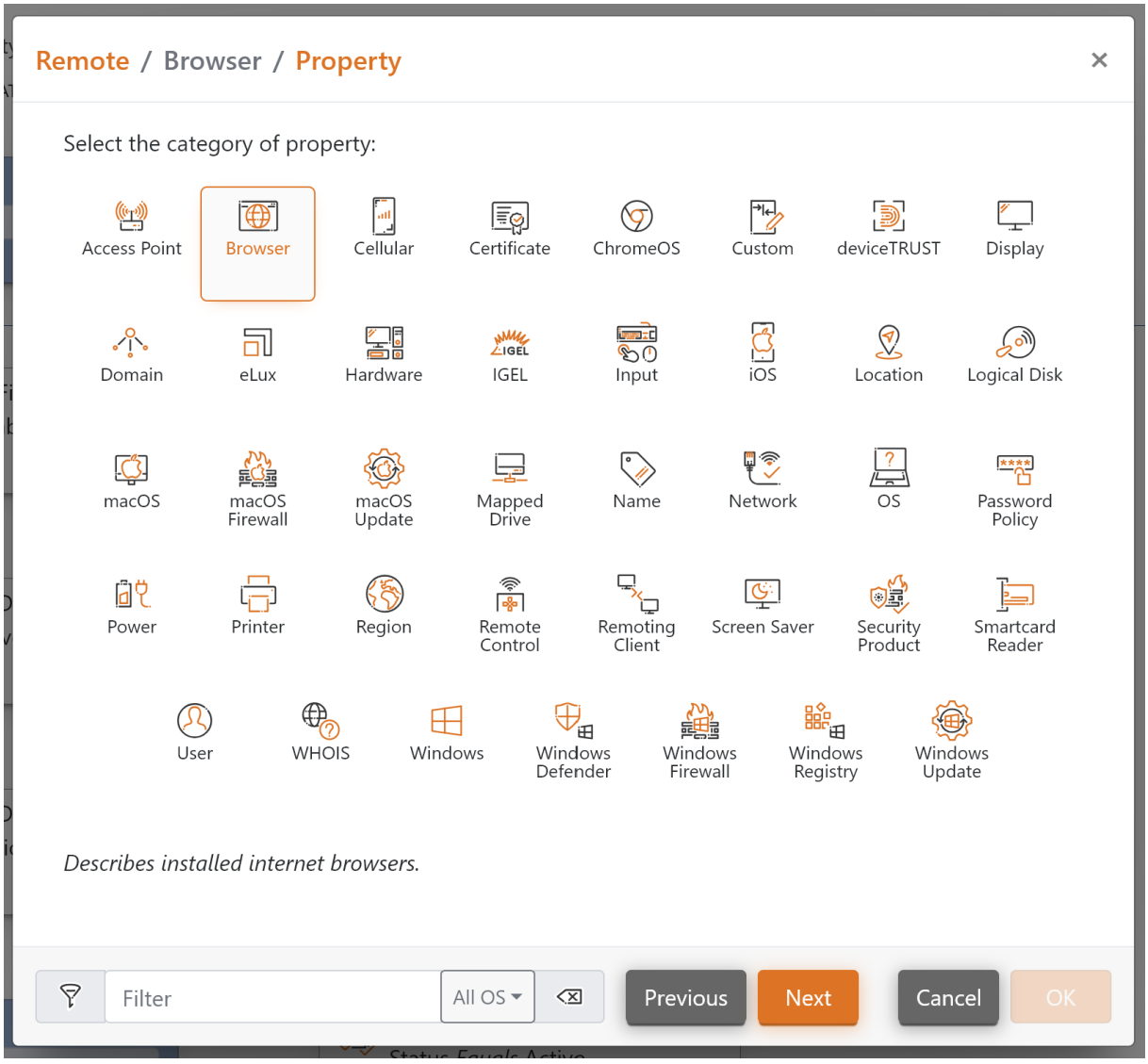
deviceTRUST 21.1 SP2 (version 21.1.310)

This service pack release includes minor features and bug fixes to the deviceTRUST Console, Agent and Client Extension for Microsoft Windows. See deviceTRUST 21.1.210 release notes for full details of the changes introduced in 21.1.210. Please refer to Compatibility for changes that may impact users upgrading from previous releases.

The deviceTRUST 21.1.310 patch includes bugfixes and minor enhancements to the deviceTRUST Console, Agent and Client Extension.

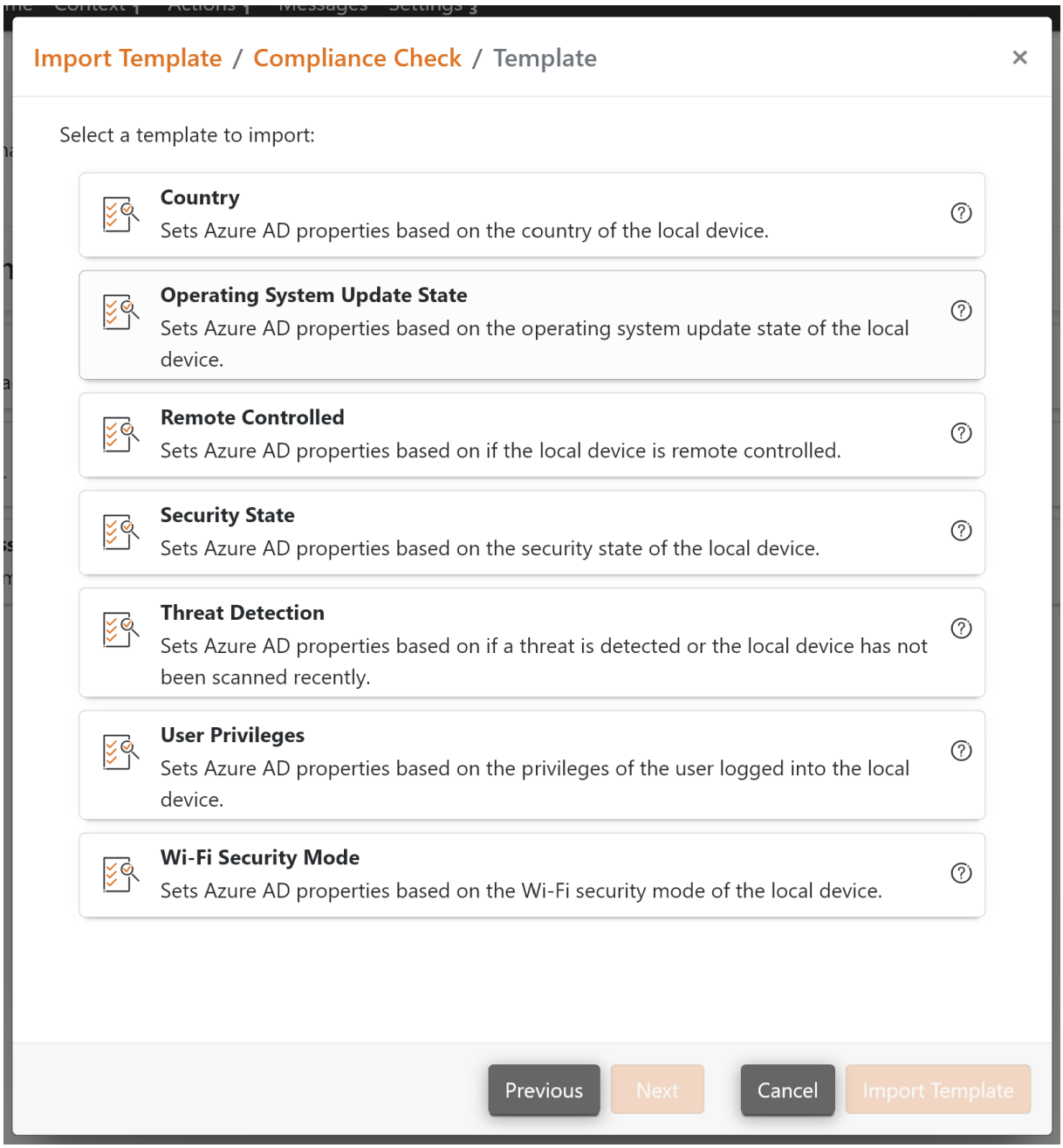
User interface refinement

We've refined the branding within the deviceTRUST® Console, and also enhanced the layout of many of our pages. This includes the context condition picker, which now offers a filter allowing faster navigation of properties either by name or by their supported operating system.



New Cloud Templates

A new collection of templates targeting the cloud and our Azure AD integration has been added. More information can be found cloud.



Other changes

- The Custom Properties Settings now supports remote Linux and IGEL devices. The forthcoming deviceTRUST Client Extension 21.1.300 or later for IGEL and Linux devices is required.
- The Terminate App task now groups together processes with matching names and parent / child relationships.
- When configured with mobile integration, the deviceTRUST Agent no longer calls to the server on Citrix® platforms when Citrix reports that the remote device is Windows, Linux or macOS.

Bug fixes in 21.1.300

- Fixed an issue where the Please Wait screen would appear on Local console sessions even when the `Block user session during Logon and Reconnect (or Unlock)` advanced setting was enabled.
- Fixed an issue where multiple resizes of a full screen Citrix session would occur during an Unlock on a Local device running the deviceTRUST Agent.
- Fixed an issue where Remote Custom Properties for Windows would not work when using Local Custom Properties on the same device.
- Fixed an issue where the WHOIS country could be cached even when it was empty.
- Fixed an issue where the deletion of properties would fail when the remote device was running the deviceTRUST Client Extension 20.2 or earlier.
- Fixed an issue where the Terminate App task, when matching processes by process name, could be triggered when the context filter no longer matches.

Bug fixes in 21.1.310

- Fixed a bug where the deviceTRUST Console would very rarely save or export encrypted deviceTRUST Policy files with a truncated random key, preventing the policy from being loaded.
- Fixed issue where Event ID's 201, 202 and 203 from remote custom properties execution would have a 1970 trigger time.

Minor enhancements to 21.1.310

- Added support for session idle time of remote Windows devices.
- Added support for the Popup Task to execute a named trigger when the default OK button is clicked.
- Updated the deviceTRUST Console's supported remote device properties available on Linux, IGEL and iOS platforms.

Compatibility

Please refer to our general approach to compatibility which can be found on the compatibility page.

If upgrading from a version older than deviceTRUST 21.1.210, be sure to refer to the deviceTRUST 21.1.210 Compatibility notes.

IGEL Client Extension 21.1.310

This release includes new features and bug fixes, and is natively integrated into IGEL OS 11.08.200 and later.

Custom Properties for IGEL OS devices

The Custom Properties Settings now supports remote IGEL OS devices. This allows a custom script to be executed on the remote IGEL device by writing `REMOTE_CUSTOM_NAME=VALUE` to the output.

Native IGEL OS integration

IGEL OS 11.08.200 now supports native integration of the deviceTRUST Client Extension for Amazon WorkSpaces, Citrix Workspace App, Microsoft Azure Virtual Desktop and Free RDP. Additionally, VMware Horizon View is supported in IGEL OS 11.08.230. For more information, see Supported IGEL Operating Systems for Client Extension.

Bug fixes

- Fixed an issue where the Third Party Geolocation provider could fail to determine a position.

Known issues

- We don't have the necessary permissions to determine the *REMOTE_HARDWARE_BIOS_SERIAL* property on an IGEL device. Until we have a permanent solution, this can be solved by running `dmidecode -s system-serial-number > /wfs/bios_serial_number` within the IGEL configuration at System -> Firmware Customization -> Custom Commands -> Base -> Final initialization command. Please note that the location of this file has changed in this release.

Compatibility

Please see the Known issues for a change to the default location of the `bios_serial_number` file.

There are no other compatibility concerns with this release of the deviceTRUST® IGEL Client Extension.