1.1.1.1.1

ESMIG U2A

Qualified Configurations

V1.3.1

Author	4CB
Version	1.3.1
Date	15/10/2021
Status	Final
Classification	Unclassified
Accessible	
Classified until	

History of releases

RELEASE	DATE	ISSUES	STATUS[1]
1.0	01/03/2021	First version. Applicable for TIPS	Draft
1.0.1	06/04/2021	Second version, clarifications on support for terminal servers	Draft
1.1	05/05/2021	Third version. Extension to CLM and RTGS GUIs	Final
1.2	26/07/2021	Added terminal server support for Ascertia client. Extension to ECMS. Ascertia client URLs changed. Minor clarifications on U2A configurations. Added section in the annex concerning the GSD multi-user solution	Final
1.3	20/08/2021	Minor integrations to GSD multi-user solution installations Notes and typos amended	Final
1.3.1	15/10/2021	Added notes about HSM based certificate usage. Added clarifications in the "GSD multi-user solution installation" Annex.	

[1] Status value : Draft, Open, Final, Dismiss

Table of contents

1 INTRODUCTION

1.2 Purpose and Objectives

This document describes the general configuration that ESMIG users shall be complaint with in order to access TIPS, ECMS, RTGS, CLM and CRDM GUI via the ESMIG web portal. A specific section is devoted to describe the technical framework needed to fully implement the non-repudiation of origin functionality (NRO). This solution will be implemented in TIPS via the Change Request TIPS-0034-SYS, when the applet technology will be decommissioned in favour of a browser's java plugin independent solution. In RTGS and CLM GUIs the same solution will be implemented according to the official plan.

1.2.1 Background remarks

The aim of the ESMIG qualified configurations is to provide ESMIG users with a specific configuration that is proved to be fully working.

As already mentioned, the NRO solution, based on the Ascertia Go>Sign Desktop application, will be the unique U2A NRO solution to be adopted for TARGET services, therefore only one version of the Go>Sign Desktop client will be used and distributed across the different services.

Important also to hightlight that Go>Sign Desktop client applications are already in use in TARGET2 for Internet Access and Contingency Network and 4CBs will guarantee that no different versions are needed by the relevant services using the client, before the go-live of CSLD project.

1.2.2 Qualified configurations

As already mentioned, the 4CB has qualified a specific subset of the NSPs compatibility matrix. These configurations have been extensively tested and support on them is guaranteed.

NSP	SWIFT	SIA-COLT
OS	Windows 10	
Browser	Google Chrome 88.0+, Firefox 78.0+	
Go>Sign Desktop	> 6.0.0.14	

These cryptographic key stores, used to access the signing keys, are supported:

- PKCS#11 for hardware-based tokens

- HSM based certificates (as per NSP specifications)

The 4CB will ask customers running a software version lower than that qualified to upgrade to a qualified version in order to proceed with problem investigation.

The 4CB will investigate issues experienced by customers while running a software version higher than that qualified: If the root cause is linked to the specific software version, then the 4CB will attempt to find a workaround (which may involve customers downgrading their software to a qualified version). The 4CB will evaluate whether a fix for the issue can be included in a future relevant TARGET Service GUI release.

Customers using totally or partially different system components or versions than those mentioned are then responsible to verify the full compatibility with the relevant TARGET Service GUI in the test environments and the system. The 4CB will in any case, provide support to the maximum extent possible for checking / testing alternative configurations in order to support troubleshooting process.

The local customer system set-up, i.e. adaption of the local firewall and security policies in order to enable the client installation, HTTPS transfer communication, access to the certificates on the USB tokens from the client machines (either physical or remote workstations) is under the sole responsibility of the end users (that may also need to involve their internal IT Dept. as well as external providers, in case of product specific issues).

1.3 Technical requirements and recommendations

1.3.1 Download mechanism

The client is available for download at the following URLs on the ESMIG portal:

EAC PORTAL

https://esmig-eac-portal.u2a.sianet.sia.eu/gosign/download/32bit_client

https://esmig-eac-portal.emip.swiftnet.sipn.swift.com/gosign/download/32bit_client

https://esmig-eac-portal.u2a.sianet.sia.eu/gosign/download/64bit_client

https://esmig-eac-portal.emip.swiftnet.sipn.swift.com/gosign/download/64bit_client

UTEST PORTAL

https://esmig-cert-portal.u2a.sianet.sia.eu/gosign/download/32bit_client

https://esmig-cert-portal.emip.swiftnet.sipn.swift.com/gosign/download/32bit_client

https://esmig-cert-portal.u2a.sianet.sia.eu/gosign/download/64bit_client

https://esmig-cert-portal.emip.swiftnet.sipn.swift.com/gosign/download/64bit_client

PROD PORTAL

https://esmig-portal.u2a.sianet.sia.eu/gosign/download/32bit_client

https://esmig-portal.emip.swiftnet.sipn.swift.com/gosign/download/32bit_client

https://esmig-portal.u2a.sianet.sia.eu/gosign/download/64bit_client

https://esmig-portal.emip.swiftnet.sipn.swift.com/gosign/download/64bit_client

(64bit version is recommended ; 32bit to be used in case of specific needs).

The full installation guide provided by Ascertia is distributed separately and it can be used as reference for specific needs (e.g. automated installations).

Downloading and installing the Go>Sign Desktop client is a mandatory step to sign U2A requests. Installation requires administrative privileges; local IT support must be involved to make sure the installation correctly ends.

Please make sure the correct version Go>Sign desktop is installed. To check this please right click on the go sign icon and choose "about". After that the following window appears:

Detailed installation steps and troubleshooting tips for Multi User environment are reported in the Annex.

1.3.2 Go>Sign Desktop Client Requirements

The client invocation on user side will be triggered by the web application (via Javascript) with the first attempt to sign an instruction (and each time the user needs to sign one) and is transparent to users.

ADSS Go>Sign Desktop relies on TLS communication only with the web application (port 8782). This communication is secured using a TLS server certificate having hostname:

client.go-sign-desktop.com.

Therefore, the local client machine must be able to resolve this FQDN (Fully Qualified Domain Name complete domain name for a specific computer, or host, on the internet) to itself.

In order to achieve this, the standard procedure foresees that the Go>Sign Desktop Installer automatically adds the entry :

127.0.0.1 client.go-sign-desktop.com

in the Operating System host file to register the client.go-sign-desktop.com as a local domain (Windows OS: C:\Windows\System32\Drivers\etc\hosts).

This will add the FQDN client.go-sign-desktop.com to resolve to IP address 127.0.0.1.

The default value client.go-sign-desktop.com must not be changed.

The TLS server certificate will be self-signed and different for each workstation where the client will be installed. Once loaded into Windows OS, it is expected to be found in the WINDOWS-ROOT CA keyring (i.e. and not in the personal certificate keyring).

The end users have to ensure that the security settings of their institutions, i.e. firewalls, allow for installation of the applet/desktop client as well as for code signing certificate revocation check, if not generally disabled. By default, User Account Control Settings (UAC) are enabled in windows. ADSS Go>Sign Desktop needs user permissions to make changes on the installing device. Windows always prompt a dialog to get the user permissions if user granted the permissions then ADSS Go>Sign Desktop would be installed on the device.

1.3.3 Other technical requirements

Here following a list of items to be checked before starting the test sessions or in case of exceptions that may block the testing. Internal IT support may be needed to perform these checks because security restrictions may be in place preventing the end users to complete them autonomously.

- As a general remark, please make sure that the configurations listed in the relevant NSPs documentation are applied (as a not exhaustive example, the mandatory changes on the pac file). For further details please refer to the "SWIFT's Solution for ESMIG U2A Setup Guide Step-by-Step" document and the " SIAnet.XS Connectivity Services for ESMIG U2A User Guide"
- In case of certificate exceptions in the browser during first interaction with new Ascertia infrastructure: add DSS host certificates in browsers keyring (e.g Chrome and Firefox). Host names following for information:

SIA TST	esmig-tst-dss.u2a.sianet.sia.eu
SIA CRT	esmig-cert-dss.u2a.sianet.sia.eu
SIA PRD	esmig-dss.u2a.sianet.sia.eu
SWIFT TST	esmig-tst-dss.emip.swiftnet.sipn.swift.com
SWIFT CRT	esmig-cert-dss.emip.swiftnet.sipn.swift.com

SWIFT PRD esmig-dss.emip.swiftnet.sipn.swift.com

The same above URL may need to be added to the browsers trusted sites.

- In case of Cross-Origin Resource Sharing (CORS) issue during first interaction with new Ascertia infrastructure, please temporarily disable CORS checks both in Chrome and/or FF

a. FF --> https://addons.mozilla.org/es/firefox/addon/access-control-allow-origin/ + Toggle ON

b. Chrome --> "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --disable-web-security --user data-dir="C:\.....\Chrome" (for single user environment)

c. Chrome --> "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe" --disable-web-security (for multi user environment)

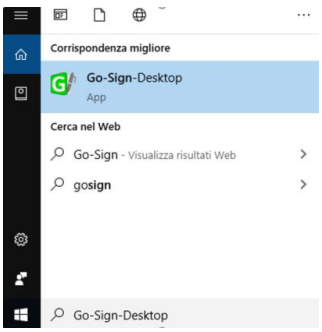
- Check windows host file for the definition 127.0.0.1 client.go-sign-desktop.com

- Check installation / install "client.go-sign-desktop.com" certificate in Chrome and Firefox explicitly (or check first if it is in all browsers keyring after GSD client installation). Without this exception error code 404 may be displayed. Also ensure that Firefox is allowed to check / read certificates from Windows keystore.

It is finally suggested to ensure that one token at time is connected to a workstation during signing operation.

1.4 Running the Application Go-Sign-Desktop

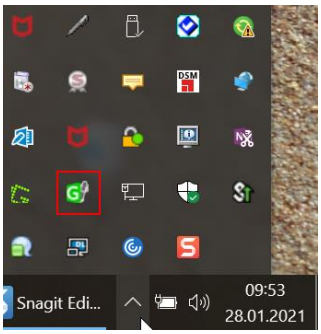
Once the application is installed, it is usually configured to run automatically when a Windows session is started. However, due to specific security settings, this might not happen, resulting signature attempts to end with a similar error "Go>Sign desktop not running/installed". In this case, it is necessary to run it manually before initiating a browsing session in ICM. It is possible to lookup for the Go>Sign via the Windows Search bar:



If the icon is not found, this can be probably linked to problems occurred during the installation. The local IT support needs to be involved in this case.

1.4.1 Verifying Go>Sign application running

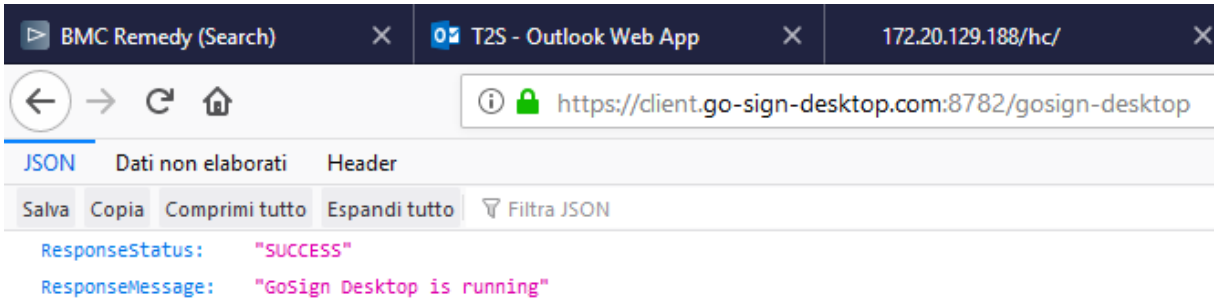
Ensure that the Go>Sign icon is featured in the system tray.



In addition, it is requested to verify that Go>Sign is running properly, by accessing the URL

https://client.go-sign-desktop.com:8782/gosign-desktop

The screenshot below is the expected result with Mozilla Firefox:



1.5 Troubleshooting information - Logging information

ADSS Go>Sign Desktop application has two log levels. First informational, which is for normal use, and second, debug, which should only be used when investigating performance issues, functionality problems, etc. For Windows OS and go>sign client configuration, users can view ADSS Go>Sign Desktop application logs at:

C:\Users\[User_Name]\AppData\Roaming\Ascertia\Go-Sign-Desktop\logs\

and should send the send the "GoSignDesktopLog.txt" when opening the incident to 4CB Service Desk.

1.5.1 Changing logging level

By default, ADSS Go>Sign Desktop logging level is set to INFO. To enable detailed debug logging, follow these instructions:

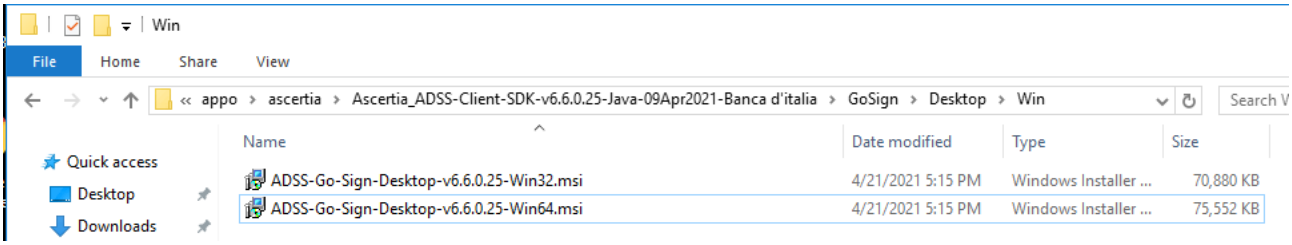
1. Go to ADSS Go>Sign Desktop installation path → C:\Program Files\Ascertia\Go-Sign-Desktop\app\conf\
2. Edit the gosign_desktop.properties file using a suitable text editor.
3. Change the value of the property GOSIGN_DESKTOP_LOG_LEVEL from INFO to DEBUG and save the file.
4. Stop ADSS Go>Sign Desktop application → right click ADSS Go>Sign Desktop application icon and select the option Quit.
5. Start ADSS Go>Sign Desktop application → Start Menu

2.1 GoSign Desktop (GSD) Client – Terminal server Installation Guide

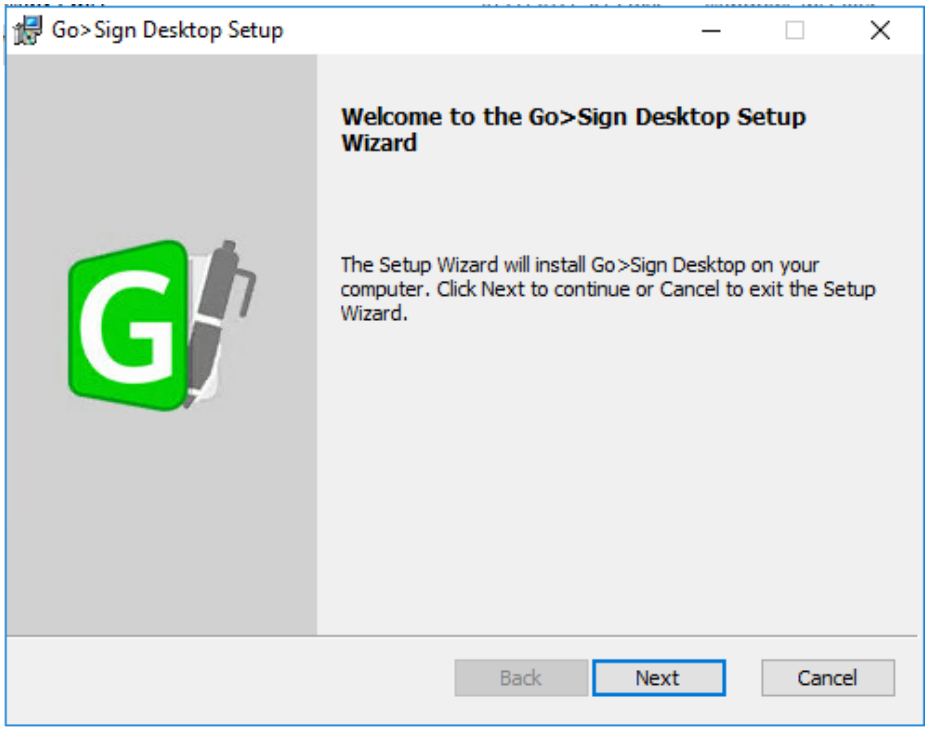
Installation steps are reported in the following paragraph; they may subject to further changes / improvements in order to simplify the overall process. Installation consists in a code upgrade of the standard GSD single user client plus some customizations into the terminal server environment.

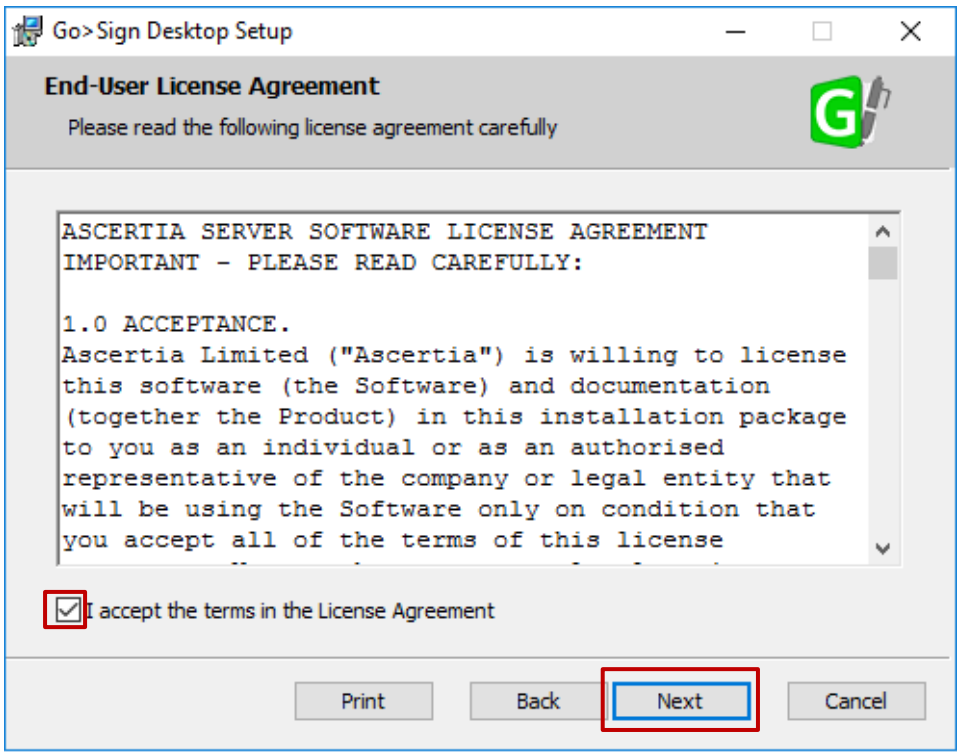
2.1.1 Setup GSD single user client

- Open Command prompt as Administrator
- Execute command: chgusr /install
- then run ADSS-Go-Sign-Desktop-v.6.6.0.xx-win64.msi installation package (currently distributed 6.6.0.14 can be used; 64bit version suggested)

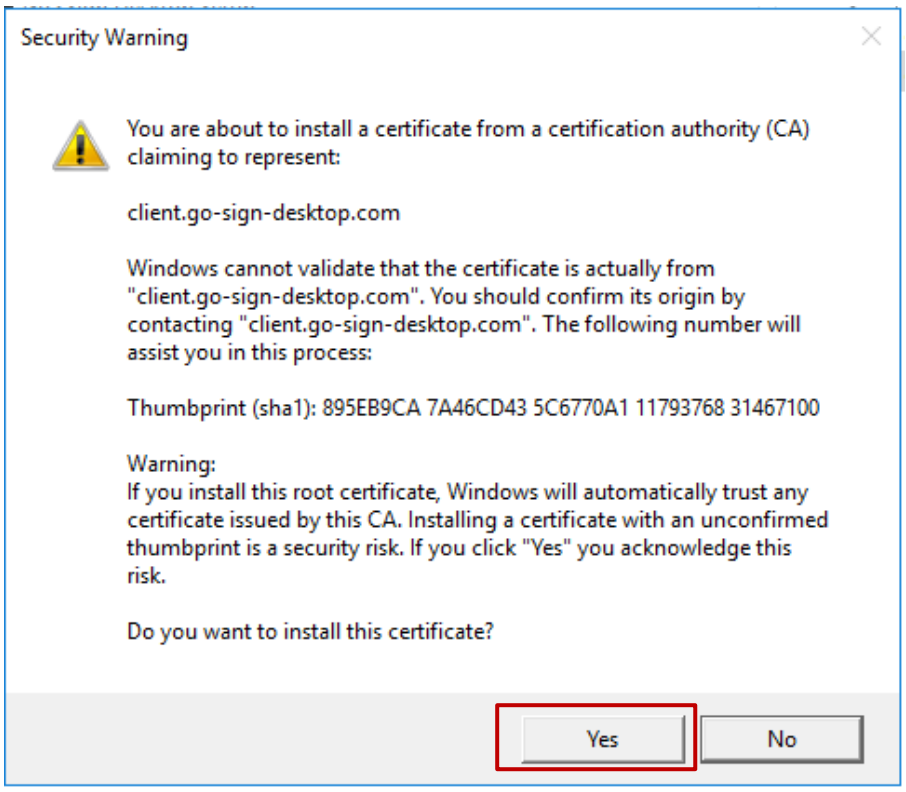


Click Next and accept End User License Agreement

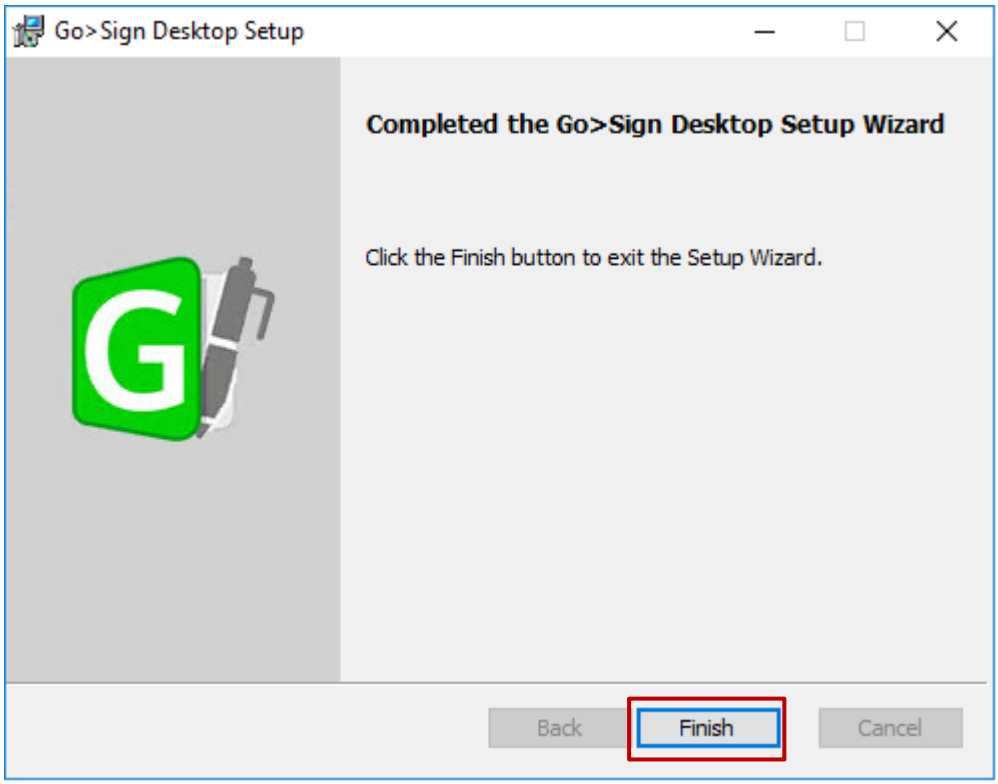




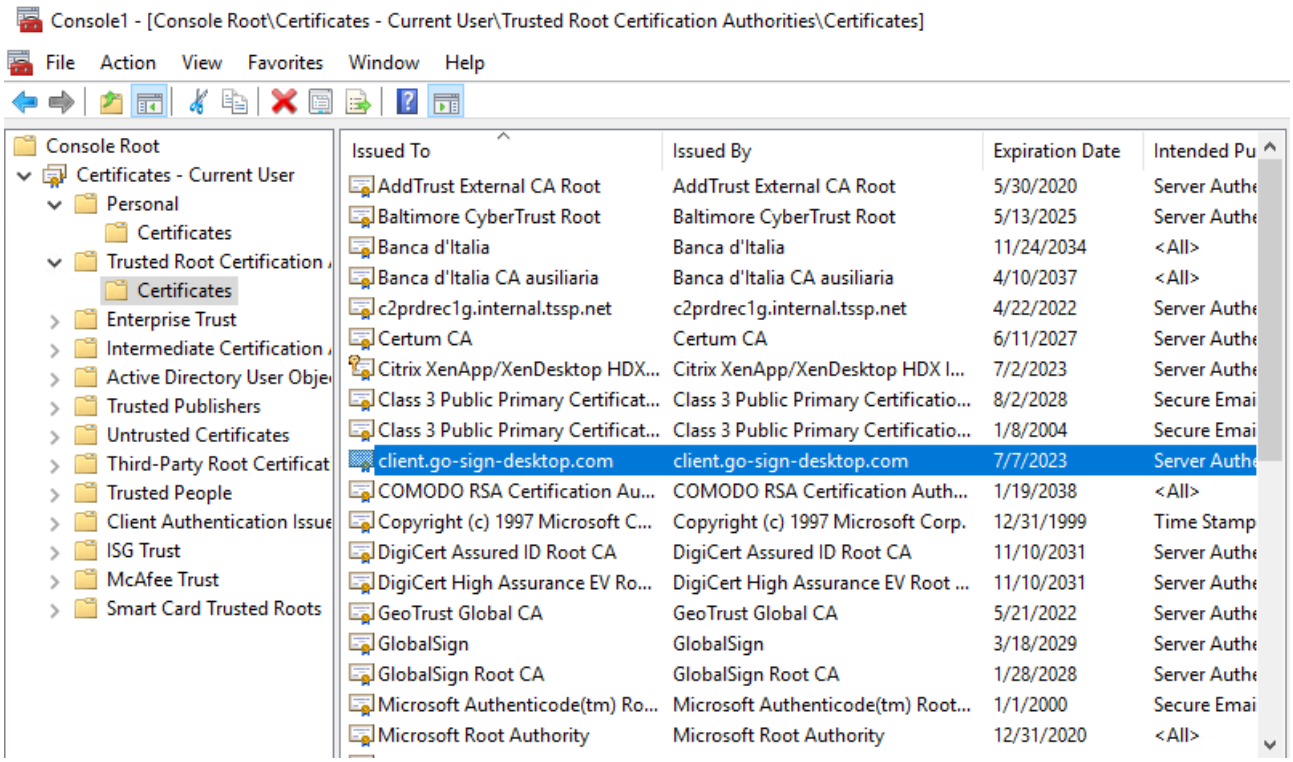
Accept to install certificate:



Select Finish:



Check that certificate client.go-sign-desktop.com is imported in the (Administrator) User Certificate store by running *certmgr.msc* tool:



2.1.2 Download and copy GSD multi user code into GSD client installation path

Please note that the following operations need administrative rights.

Download the updated code for terminal server environments from one of the following URLs:

EAC

https://esmig-eac-portal.u2a.sianet.sia.eu/gosign/download/gsdmu_client

https://esmig-eac-portal.emip.swiftnet.sipn.swift.com/gosign/download/gsdmu_client

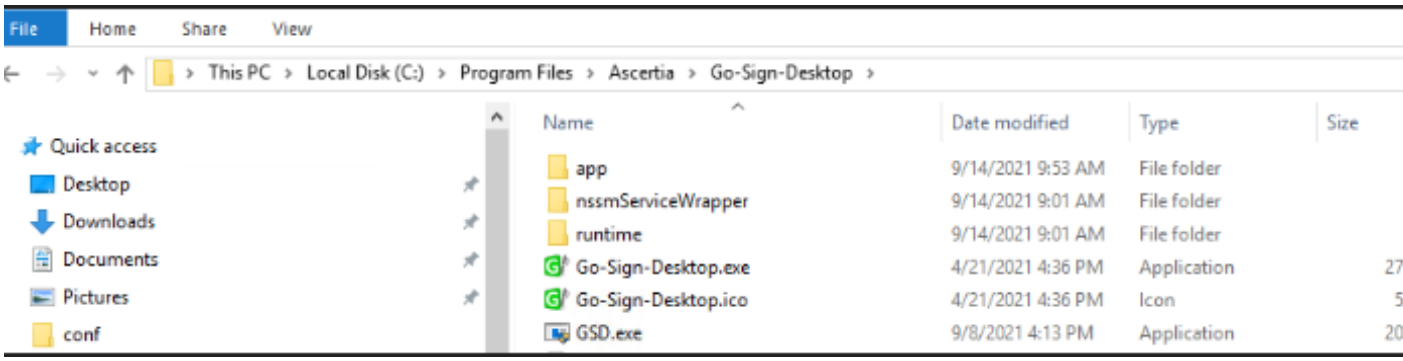
UTEST

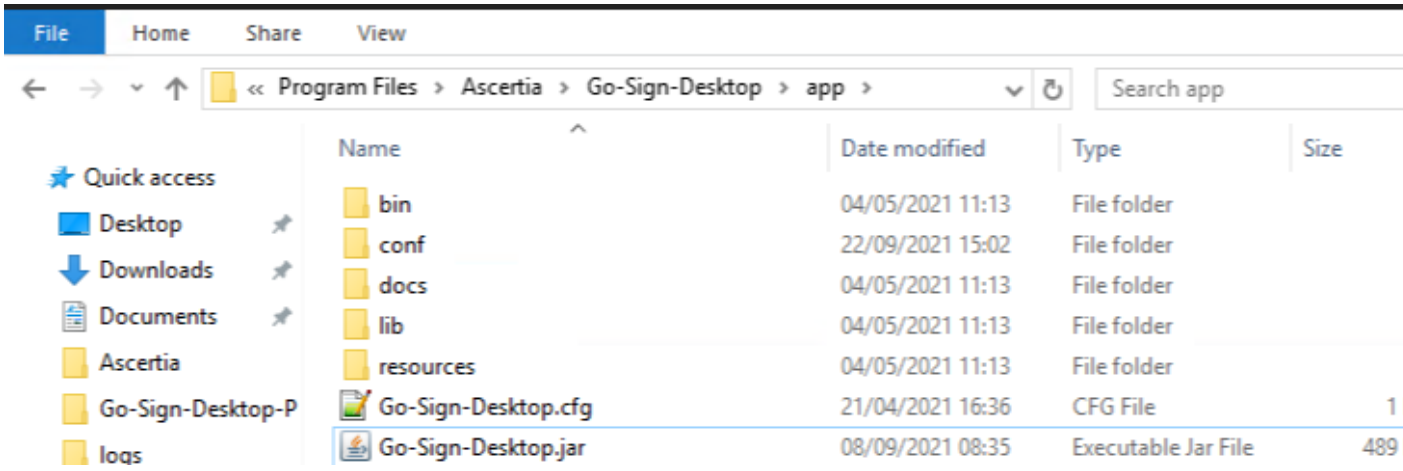
https://esmig-cert-portal.u2a.sianet.sia.eu/gosign/download/gsdmu_client

https://esmig-cert-portal.emip.swiftnet.sipn.swift.com/gosign/download/gsdmu_client

And

1) copy the GSD.exe file into GSD installation directory
2) replace the existing Go-Sign-Desktop.jar file with the one provided in the above package
3) see below screenshots as reference





2.1.3 Update go-sign-desktop.properties file

Check this line is already present:

GOSIGN_DESKTOP_HTTPS_PORT=8782

Following two lines have to be added to the go-sign-desktop.properties file (C:\Program Files\Ascertia\Go-Sign-Desktop\app\conf directory):

GOSIGN_DESKTOP_INSTALLATION_MODE=MULTI_USER

GOSIGN_DESKTOP_LOG_MODE=info

Second property can be used to modify user log level for troubleshooting purpose (possible values: info/debug).

2.1.4 Create user.info.properties file

Following file has to be created into C:\Program Files\Ascertia\Go-Sign-Desktop\app\conf directory:

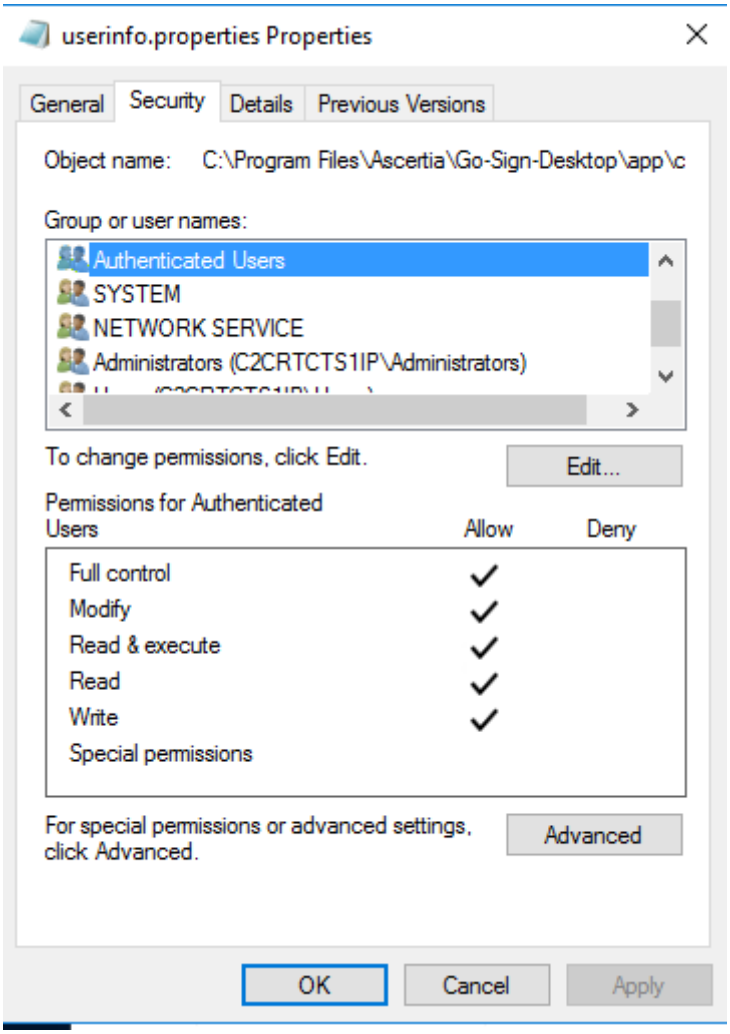
userinfo.properties

The file can be created empty and it will be populated by GSD application during NRO-activities performed by the user.

The file is expected to have with following access permissions in order for the NRO process to work correctly:

NETWORK SERVICE user needs full grant

Add Authenticated Users – Full Control if Everyone is not present



2.1.5 Configure Go-sign-desktop as Windows Service

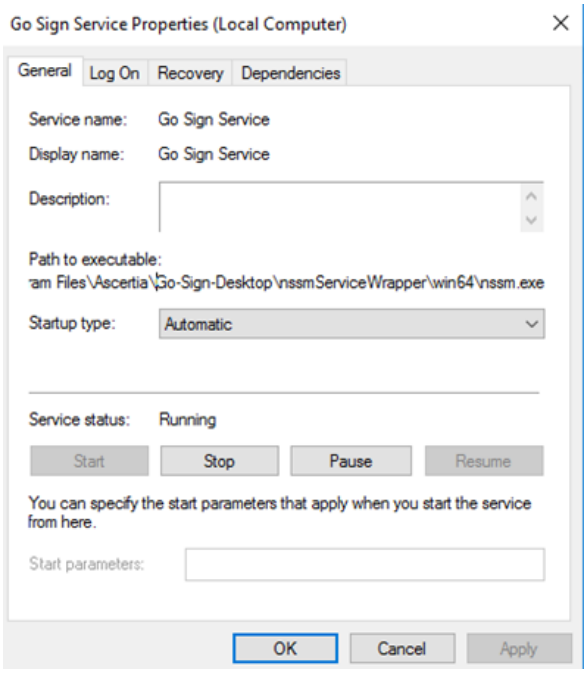
Current implementation foresees to have Go>Sign Desktop running as a Windows Service via a Service Wrapper which wraps the Go>Sign Desktop executable binary into a Windows Service via the NSSM opens source software available at https://nssm.cc/.

To configure service:

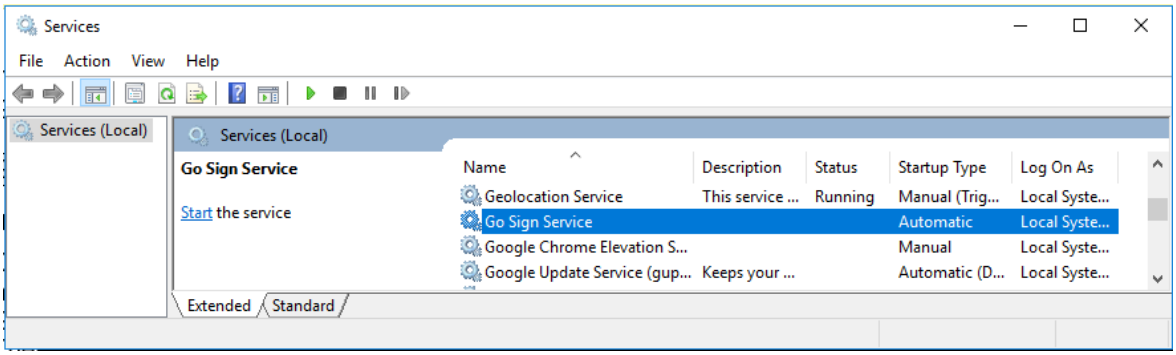
1) Extract archive content "nssmServiceWrapper.zip" to the C:\Program Files\Ascertia\Go-Sign-Desktop\ nssmServiceWrapper (64bit version suggested)

2) Run Command Prompt as Administrator and enter the following line with respect to the real location of Go>Sign Desktop:

```
"C:\Program Files\Ascertia\Go-Sign-Desktop\nssmServiceWrapper\win64\nssm.exe" install "Go Sign Service" "C:\Program Files\Ascertia\Go-Sign-Desktop\Go-Sign-Desktop.exe"
```



3) Open Services.msc and locate the Go Sign Service just installed:



- Right-click Go Sign Service and choose Properties.
- Ensure "Startup type" is set to Automatic.
- Open the Log On tab and change "Log on as" to this account, then specify the account NETWORK SERVICE and leave the password blank

Service can be started once installation procedure finalized. (Once service correctly started, IT Administrator should see process listening on port 8782).

2.1.6 Import client.gosign certificate into Network Service user keystore

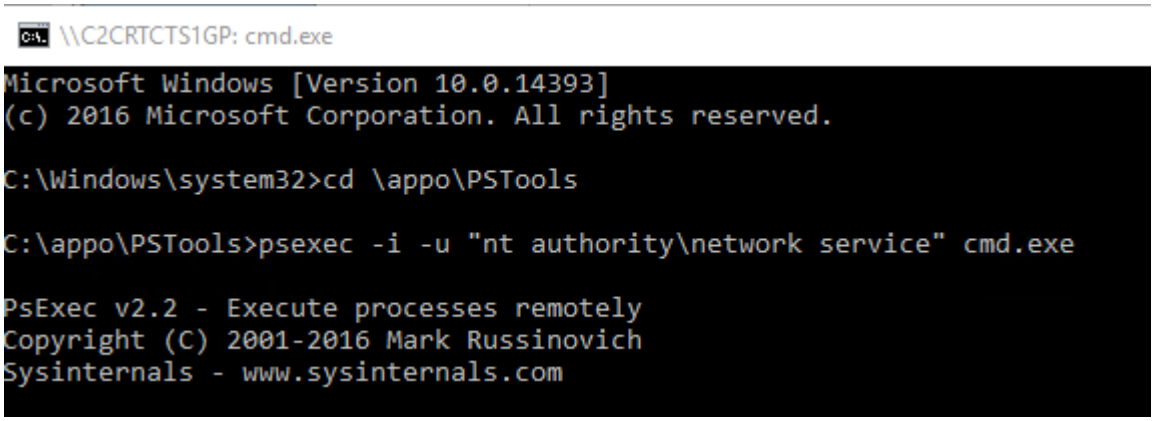
When GSD starts it check if there is the client.gosign Desktop certificate in user certificate store (see above) so to start service with user "Network Service" this certificate must be imported in that user keystore.

It can be done by following below steps:

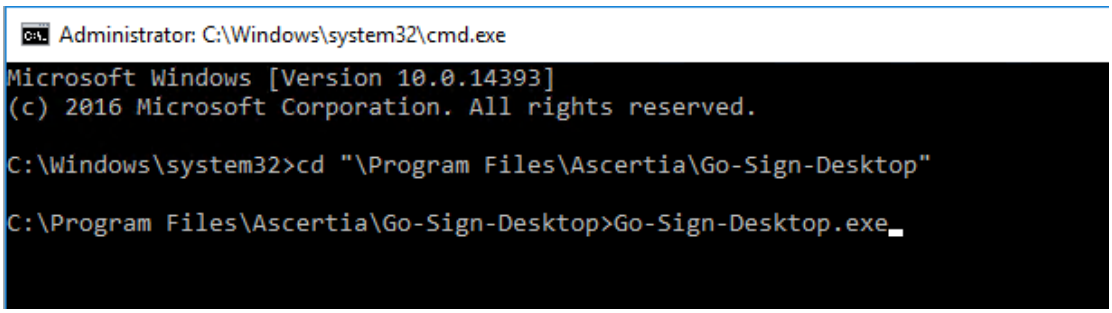
Using psexec (download link https://docs.microsoft.com/en-us/sysinternals/downloads/psexec)

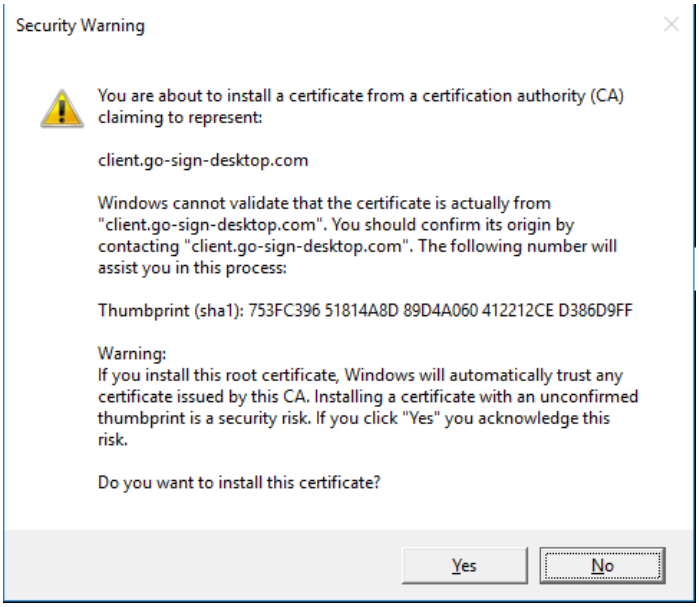
In administrative Command prompt run

```
psexec -i -u "nt authority\network service" cmd.exe
```



then you have a Command prompt opened in "Network Service" user environment so you can start Go-signDesktop.exe and it will import the client-go-sign certificate





In case of issues with above method please try starting command prompt for "Network Service" user environment and

- run *certmgr.msc* tool
- import directly the client certificate exported from the Current User Store.

2.1.7 Import client.gosign certificate into java « cacerts » keystore

Update 'cacerts' file located in

'C:|Program Files|Ascertia|Go-Sign-Desktop|runtime|jre|lib|security'

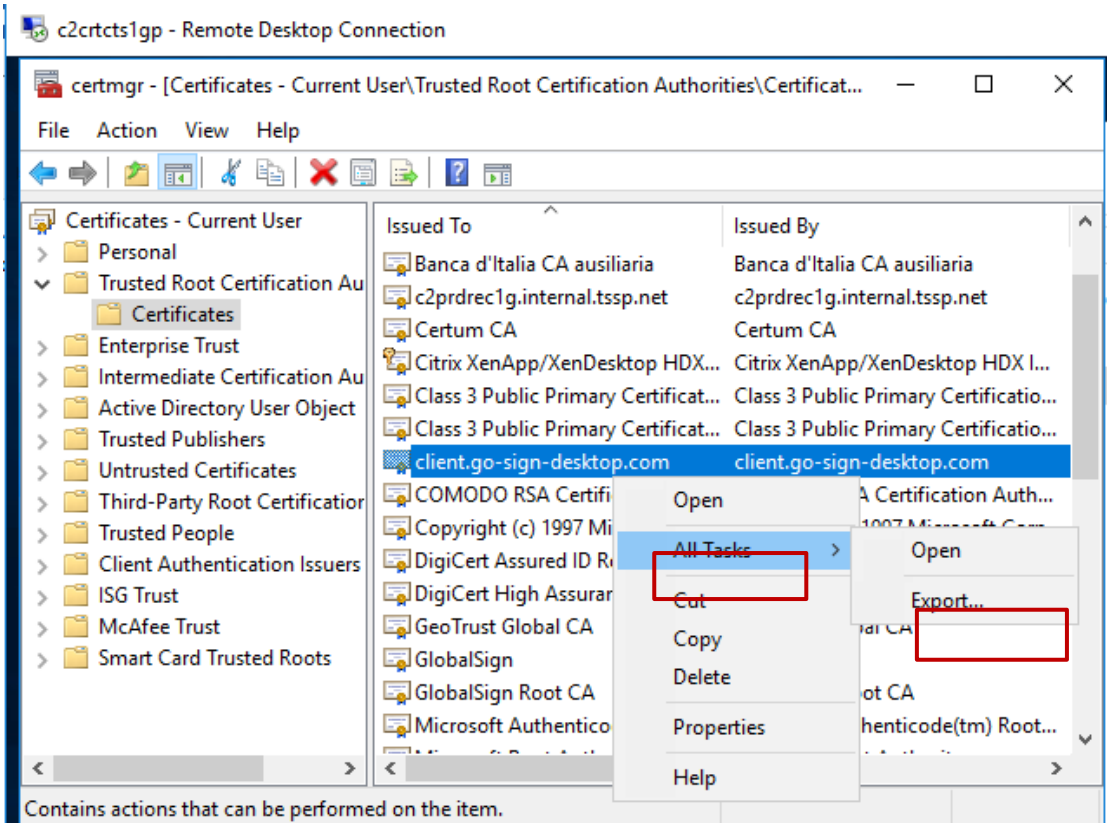
by loading the "gosign" certificate that Go>Sign Desktop app trusted when it ran for the first time.

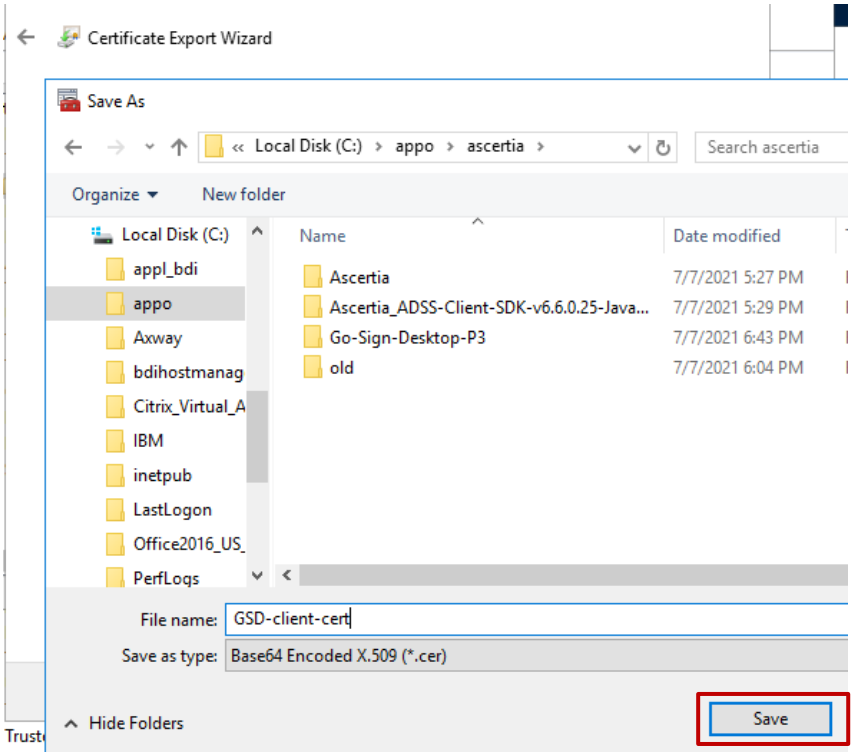
The following keytool command to be used:

keytool -import -alias gosign -file exported-cert-der.cer -keystore "C:\Program Files\Ascertia\Go-Sign-Desktop\runtime\jre\lib\security\cacerts"

Copy C:\Program Files\Ascertia\Go-Sign-Desktop\runtime\jre\lib\security\cacerts from the first server to the other terminal servers (if present).

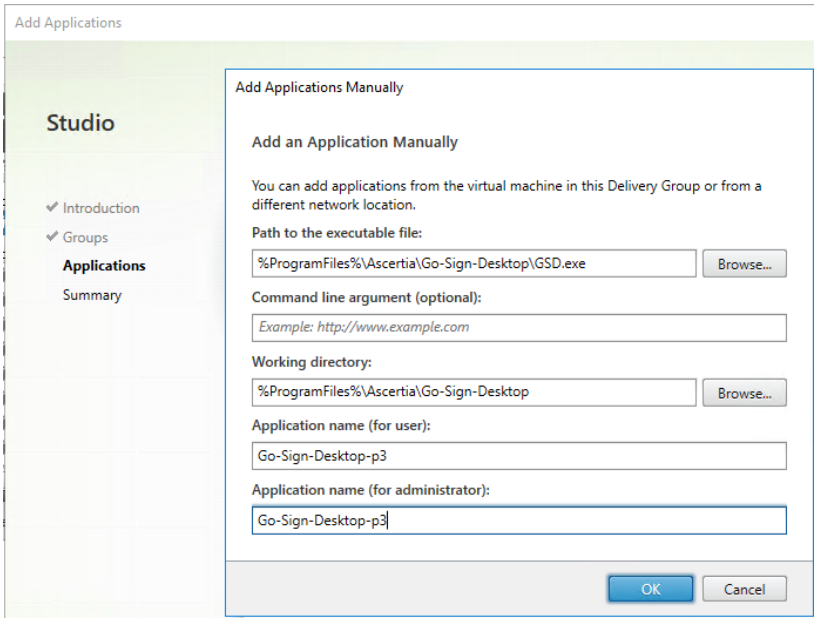
Gosign certificate can be exported via certificate manager as follows:

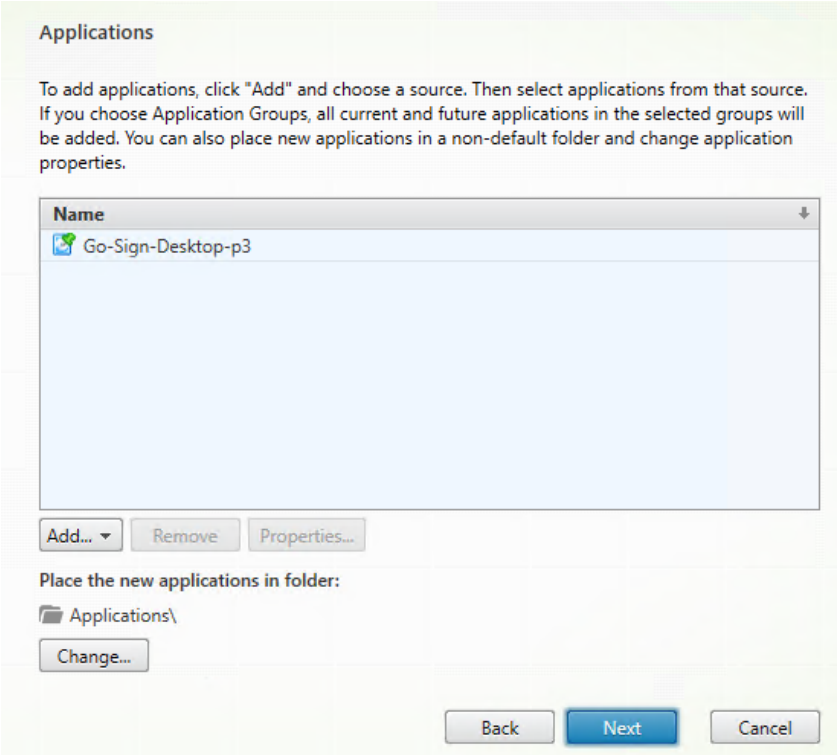




2.1.8 Publish applications GSD.exe and Chrome in Citrix Farm

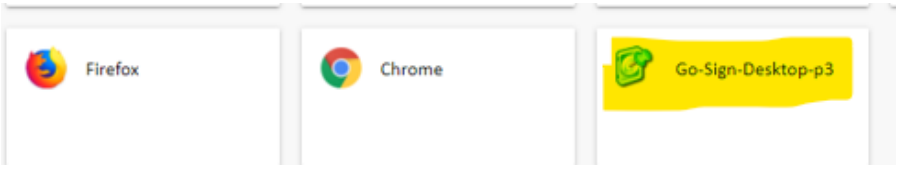
For Citrix Terminal Server environment, please use Citrix Studio console to add a new Application, assign Delivery Group and "add an application manually" as described:





To be repeated for all the relevant users.

Before using GSD each user must launch GSD.exe (one time only) application in order to add the right registry keys in user profile and save the user configuration.



Then open browser and start NRO task.

In case of RDP connections to the server, user should connect to the server and launch the GSD.exe application (only once).

2.2 GSD CLIENT TS installation– user actions

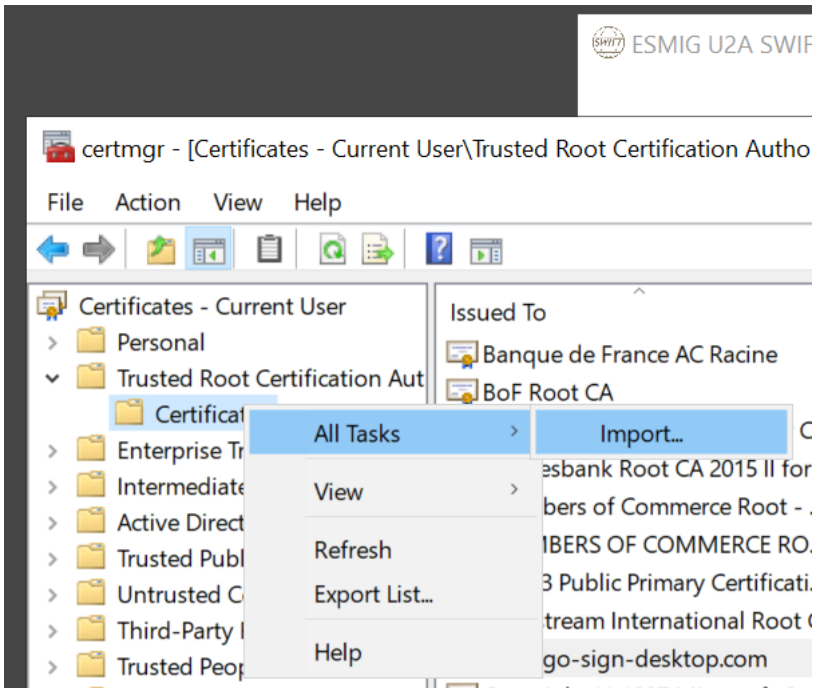
2.2.1 Import client.gosign certificate into WINDOWS-ROOT user keystore

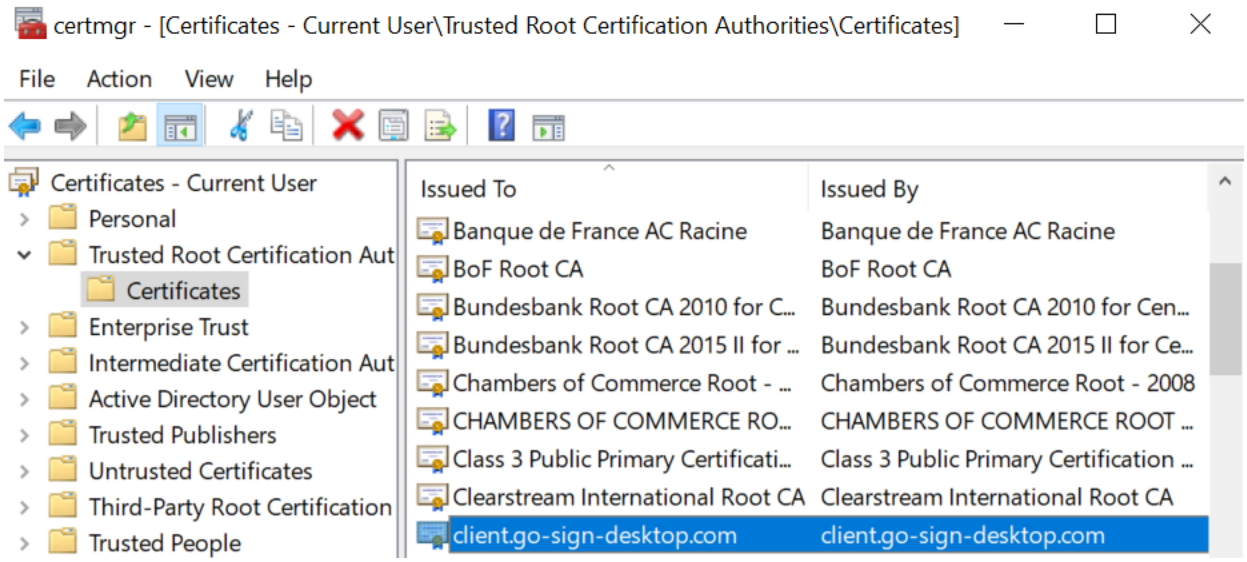
Either via GPO or via certmgr.msc tool (from user RDP session), the client.go-sign certificate should be imported in the user trust store. Please ensure that certificate will have "gosign" alias/friendly name after import.

Following steps could be followed for option #2:

1) log into the terminal server via remote desktop connection;

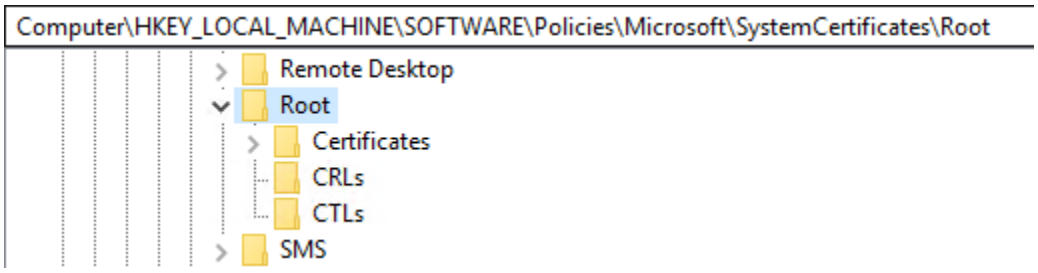
2) Open Certificate list under the respective user and import the certificate:



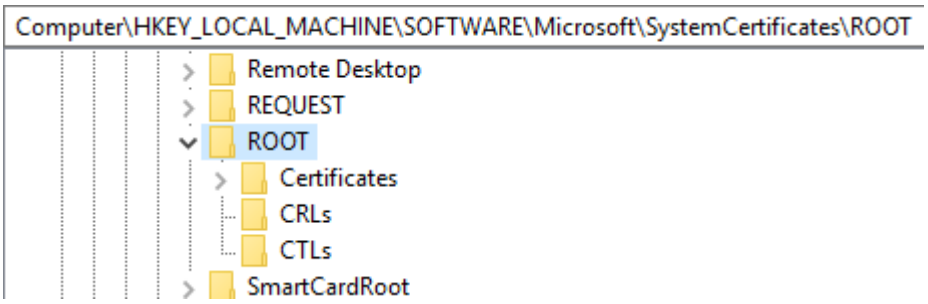


In order to ensure user is able to correctly import the gosign certificate into its trust store please double check that the following two keys do not exist:

1) HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\SystemCertificates\Root\ProtectecRoots



2) HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\SystemCertificates\Root\ProtectecRoots



and that the following key has the permission listed below:

3) HKEY_CURRENT_USER\SOFTWARE\Microsoft\SystemCertificates\Root\ProtectecRoots

APPLICATION PACKAGE AUTHORITY\Software and hardware certificates or a smart card - Allow - ReadKey

Administrator - Allow - FullControl

CryptSvc - Allow – FullControl

2.2.2 Start NRO task

Once all the above steps will be completed and GSD service running with Network Service user, business user could start an NRO task.

Right before signing, users are expected to allow execution of GSD user instance in order to allow start of a child GSD child application that will communicate with the GSD service/parent instance in order to properly perform NRO task.

GSD child instances will listen on greater ports than the GSD service/parent one and will start dedicated Go-Sign-Desktop.exe application.

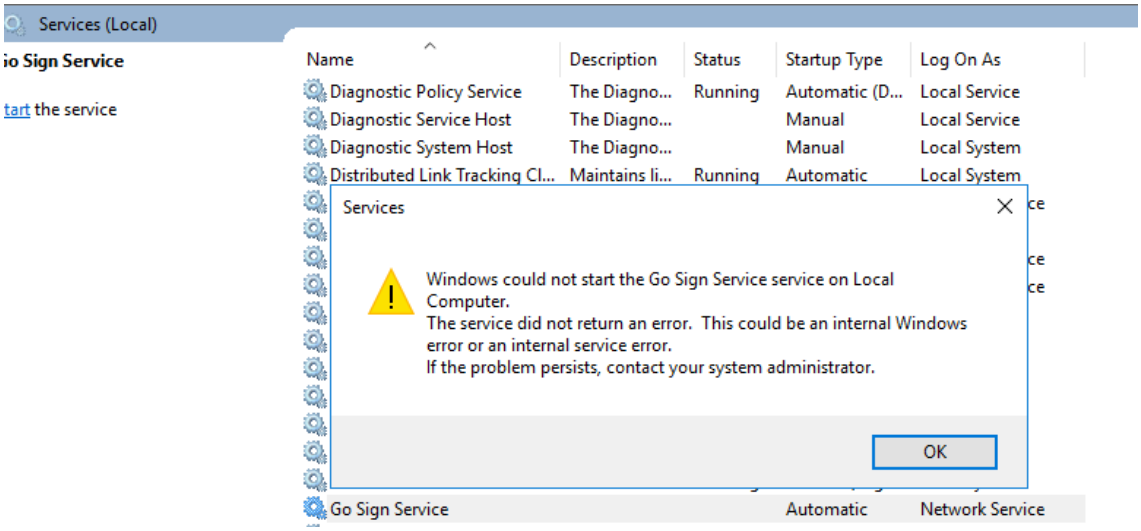
2.3 Issues

2.3.1 Service do not start

In case of issues with service start, please check and share Ascertia service logs from the following path:

C:\Windows\ServiceProfiles\NetworkService\AppData\Roaming\Ascertia\Go-Sign-Desktop\logs

See also notes in « 1.3. configure Go-sign-desktop as Windows Service « .



When service correctly started, IT Administrator should see process listening on port 8782.

2.3.2 Failed to load keystore issue during NRO task

In case user is displayed ”failed to load keystore” issue during NRO task, please double check and ensure that gosoign_app.properties file is present in Ascertia user folder and that user is allowed to properly write in this file (i.e. no GPO or User Account Control restricing access to it).

NRO will not work properly in case GSD application launched by the user during NRO task can not create and/or update the gosign_app.properties file.

Please ensure that certificate will have “gosign” alias/friendly name in all the keystore where it has been imported (network service, cacerts and user trust stores).

2.3.3 Other exceptions

In case of other exceptions:

- users are requested to collect and share browser console log and screenshots plus GSD user/child instance log file

- IT Administrator should provide GSD service/parent log