Canon
Professional Services

MOSCA

MyOwnSmartCertificateAuthority

V2.16.0

Content

Introduction 3
Requirements 3
  General 3
  External Signing 3
  Subject Alternative Names 4
Base Configuration 5
  Credentials 7
The GUI 8
Create a new Certificate Authority (CA) 9
Manual creation of a device certificate 10
The Device List 11
  Add / Edit devices 12
Automation for Certificates 13
  Bulk Generation of Device Certificates 13
  Bulk Deployment of Device Certificates 14
  Bulk Deployment of CA Certificates 14
Automation for Certificates with CSR 15
  Request CSR’s from devices 15
  Sign CSR’s from devices 16
  Deploy Certificates from CSR’s 17
  Activate Certificates from CSR’s 17
  Invoke SCEP 18
Automation for Special Functions 20
  Control TLS 20
  Query Active Certificates 20
  Remove Unused Certificates 21
Quick OpenSSL Commands 22
Devices in uniFLOW Online 24
  Account without MFA 24
  Account with MFA 24
MOSCA Web API 28
    General API answers 28
    Request CSR 29
    Deploy Certificate from CSR 30
    Activate 802.1X Certificate 31
    Request Active Certificates 32
    Request State 34
List of supported devices 35

Used Ports and Protocols 36

Introduction

MOSCA is a tool to create a Certificate Authority and generate certificates for SSL and 802.1X connections using OpenSSL.
MOSCA is designed to work with Canon MFP and SFP devices. It can request Certificate Signing Requests (CSRs) from Canon devices. The CSRs can be signed local with OpenSSL or externally against a customer CA like the MS Windows Certificate Authority. Finally the generated or signed certificates can be deployed and activated on the Canon devices.
MOSCA works with single devices and also with a bulk of devices in parallel processing.

Requirements

General

MOSCA is portable and needs no installation. In most cases MOSCA needs to be started with the “Run as administrator” option for the base configuration. After that a normal start in a user context will be sufficient.
MOSCA needs the MS .Net Framework 3.5 (or higher) or its runtime to be installed. The software runs on Windows 10 or MS Windows Server 2008R2 or higher.
Since MOSCA uses the Remote User Interface (RUI) of the Canon devices, it requires an MS Internet Explorer to be installed. This IE has to have access to the Canon devices. If not, please check your settings for IE Enhanced Security, proxy and trusted zones.
Even if you do not use the OpenSSL components in MOSCA you need to create an own Certificate Authority (CA) because MOSCA always works in the created CA folder.

External Signing

If you want to let MOSCA sign certificates on your CA, then the Remote Procedure Call (RPC) service has to run on the CA server. In addition to that, the user who started MOSCA, has to have access to the servers RPC. When signing a certificate, a certificate template is used. The user who started MOSCA also has to have the access rights to use this template.

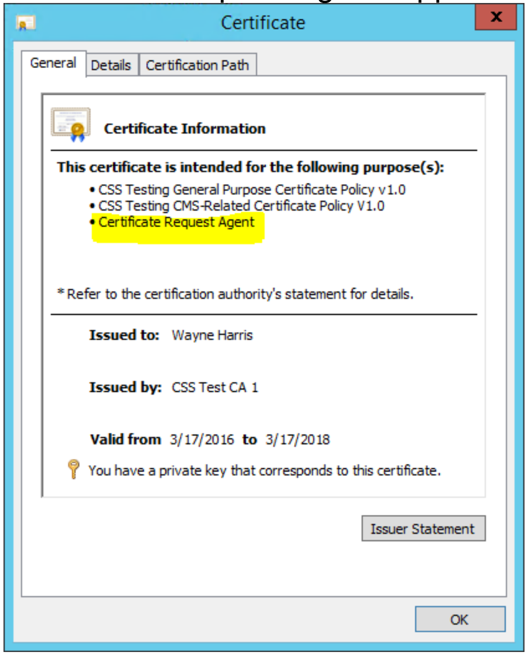
Subject Alternative Names

Since Canon devices are not able to set Subject Alternative Names (SANs) in their Certificate Signing Requests (CSRs), MOSCA is able to add SANs in the external signing process against an MS Certificate Authority.
There are two by MOSCA supported methods to that that:

Method 1 – During Signing:
To enable this method please refer to the section "Base Configuration".
The feature has to be enabled on the MS CA also. This can be done by setting the policies on the CA Server to allow adding SANs in the signing process with the following shell command:
certutil -setreg policy\EditFlags +EDITF_ATTRIBUTESUBJECTALTNAME2

You have to restart the MS Certificate Services to let the changes take effect, e.g. with these shell commands:
net stop certsvc
net start certsvc

Method 2 – Resigning CSR using EA certificate:
This method uses an enrollment agent certificate to resign the original certificate request and add the needed SANs.
You have to acquire an enrollment agent certificate. This is a certificate based off the Enrollment Agent default template. The resulting EA certificate must contain the "Certificate Request Agent" application policy extension. (1.3.6.1.4.1.311.20.2.1.)



Certificate Information

This certificate is intended for the following purpose(s):
- CSS Testing General Purpose Certificate Policy v1.0
- CSS Testing CMS-Related Certificate Policy V1.0
- Certificate Request Agent

* Refer to the certification authority's statement for details.

Issued to: Wayne Harris

Issued by: CSS Test CA 1

Valid from 3/17/2016 to 3/17/2018

You have a private key that corresponds to this certificate.

This certificate (and associated private key) should be located on the workstation of MOSCA. It is this certificate that MOSCA will use during the re-signing process to authorize itself against the CA.
To enable this method please refer to the section "Base Configuration".

Base Configuration

Please start the program once using the “Run as Administrator” option to register the external components.

All Settings of MOSCA are stored in the MOSCA.ini file in the program directory. Most of these settings are changed by using the Graphical User Interface (GUI) of MOSCA. Some of them are only editable in the MOSCA.ini file. These settings are described here:

```
[CertificateRequests]
UseFromINI=0
KeyAlg=RSA2048
SigAlg=SHA256
O=
OU=
L=
S=
C=
```

Normally the information for certificate requests comes from the chosen CA in MOSCA and from the “Certificate Request Settings” section.
If you want to override these settings of the GUI, set the “UseFromINI” setting to “1”. Please refer to the device RUI or the device manual to see which settings the device offers for the signature and key algorithms (SigAlg and KeyAlg). Please also set Organisation (O), Organisational Unit (OU), Location (L), State (S) and Country (C) as you need it.

```
[ExternalSigning]
SignExternalCommand=certreq -submit -binary -rpc -q -config "DC1\DC1-CA" -f -attrib
"CertificateTemplate:Computer802" "{FILENAMECSR}" "{FILENAMECERT}"

AppendSANs=1
```

If you want to sign certificates against an external Certificate Authority (CA), then you have to provide a shell command that will be used in this process. For an MS Windows CA a command “SignExternalCommand” is already in the MOSCA.ini file and can be changed for your requirements:

- “DC1\DC1-CA” represents the domain name of the server with the CA on it (before the backslash) and the name of the CA (after the backslash).
- “-binary” will enforce a binary certificate file which is most compatible to all Canon devices.
- “-rpc” will enforce the use of the “Remote Procedure Call” interface.
- “-q” runs the command silent without message boxes.
- The template to be used while signing is named after “CertificateTemplate:”.
- -“{FILENAMECSR}” and „{FILENAMECERT}” are placeholders and must not be changed.

With the “AppendSANs=1” setting MOSCA will try to append available Subject Alternative Names (SANs) to the request during the external signing (method 1). Please refer to the “Requirements / Subject Alternative Names / Method 1” to read more about the requirements for this feature.

```
AppendSANs=2

ReSignCSRCommand=certreq -policy -rpc -q -config "DC1\DC1-CA" "{FILENAMECSR}"
"{POLICY.INF}" "{TEMP.CSR}"
```

If you want to use method 2 to append SAN’s by resigning the original CSR you have to set the option AppendSANs to 2. You also have to provide a shell command like the above sample “ReSignCSRCommand=”. Here are the options that have to fit your environment:

- “DC1\DC1-CA” represents the domain name of the server with the CA on it (before the backslash) and the name of the CA (after the backslash).
- “{FILENAMECSR}”, „{POLICY.INF}” and „{TEMP.CSR}” are placeholders and must not be changed.

Please refer to the “Requirements / Subject Alternative Names / Method 2” to read more about the requirements for this feature.

Credentials

Credentials for accessing the devices web interfaces and to request device information via SNMP protocol have to be the same for all devices. The Following credentials could be set by pressing the “Credentials” button in MOSCA:

Username & **Password**
Will be used by iR-ADV devices with DepartmentID authentication and Lexmark devices with User Authentication. On Lexmark devices with a Password Authentication only the password will be used.

LMUser & **LMPass**
Will be used for all devices with a LoginManager authentication.

MfpRuiUser & **MfpRuiPass**
Will be used by iR-ADV devices with User authentication and LBP devices with User authentication.

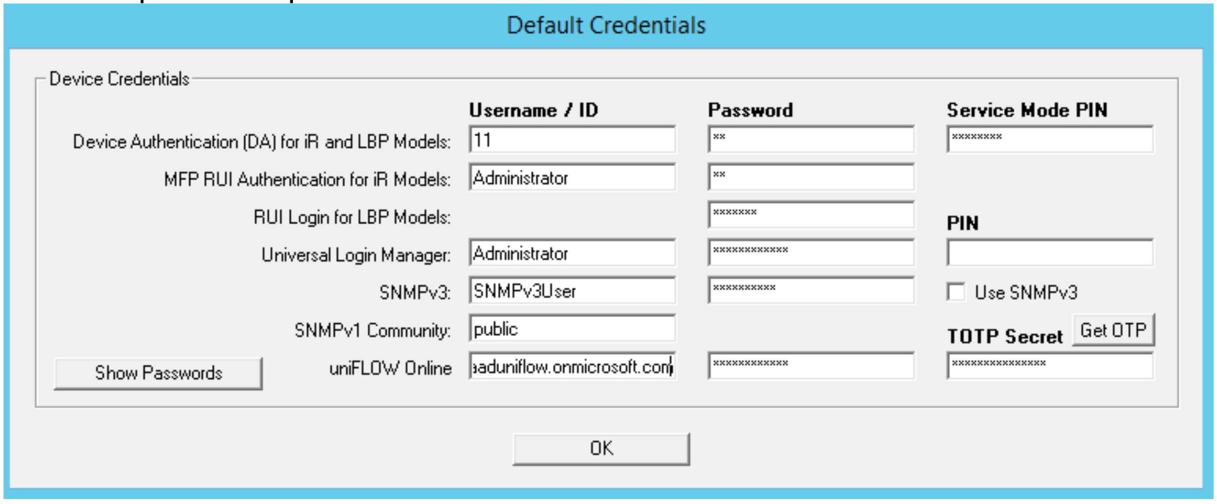
LbpRuiPass
Will be used by LBP devices with System Manger authentication

SNMPCommunity
Will be used to request device information via SNMPv1 protocol.

SNMPv3User & SNMPv3Password
Will be used to request device information via SNMPv3 protocol.

uniFLOW Online User, uniFLOW Online Password and TOTP Secret
Will be used to authenticate against devices that are connected to an uniFLOW online tenant.
Please refer to the section “Devices in uniFLOW Online” for more information.

All passwords are hidden by default. To show them, click “Show Passwords” and enter the protection password “mosca”.


Default Credentials

Device Credentials

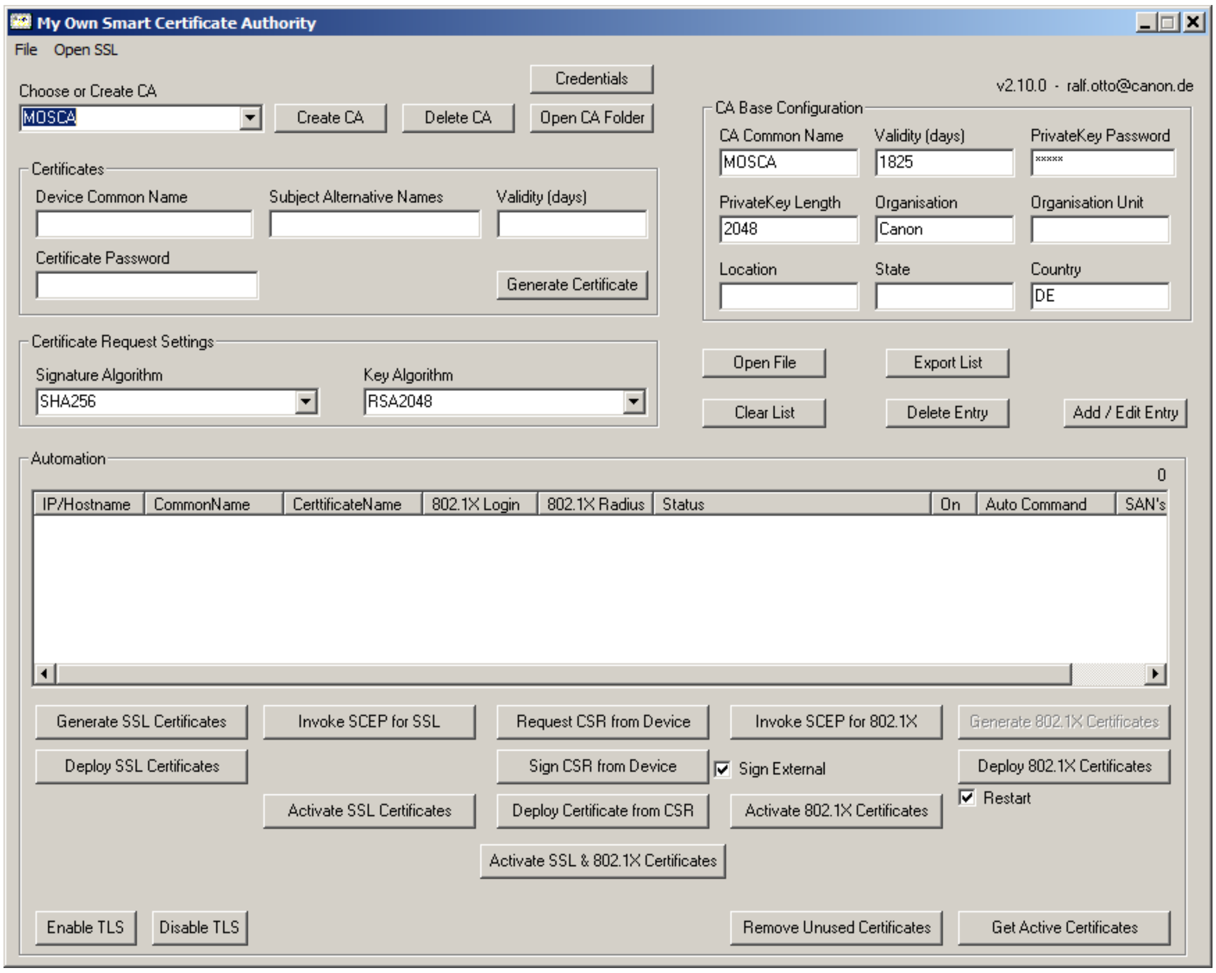
	Username / ID	**Password**	**Service Mode PIN**
Device Authentication (DA) for iR and LBP Models:	11	**	********
MFP RUI Authentication for iR Models:	Administrator	**	
RUI Login for LBP Models:		*******	**PIN**
Universal Login Manager:	Administrator	************	
SNMPv3:	SNMPv3User	**********	☐ Use SNMPv3
SNMPv1 Community:	public		**TOTP Secret** Get OTP
uniFLOW Online	aduniflow.onmicrosoft.com	************	***************

Show Passwords

OK

All passwords are written encoded in the “WriteSettings.ini” file. The SNMP settings are written in the “MOSCA.ini” file.
If “Use SNMPv3” is checked the device information will only be requested via SNMPv3 protocol.

The GUI



Choose or Create CA: MOSCA

Device Common Name:
Subject Alternative Names:
Validity (days):
Certificate Password:

Signature Algorithm: SHA256
Key Algorithm: RSA2048

CA Base Configuration
CA Common Name: MOSCA
Validity (days): 1825
PrivateKey Password: *****
PrivateKey Length: 2048
Organisation: Canon
Organisation Unit:
Location:
State:
Country: DE

[✓] Sign External
[✓] Restart

In the upper left you can create and manage own Certificate Authorities (CAs) and set the credentials to access the devices RUI.

In the upper right you can manage the base configuration of your own CAs.

In the section “Certificates” you can set the information for a single certificate creation. The values for “Subject Alternative Names” have to be entered comma separated.

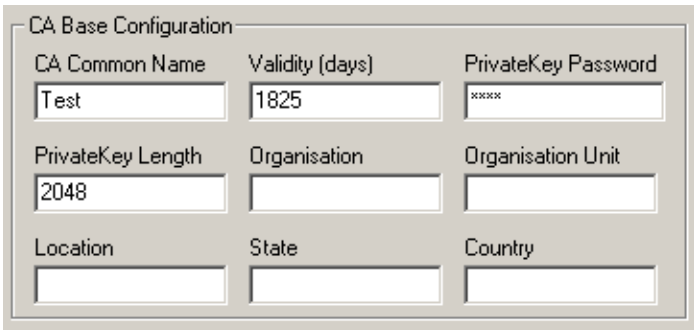
The section “Certificate Requests Settings” defines the base settings for device created Certificate Signing Requests (CSRs).

In the section “Automation” you will find the device list for bulk processing CSRs, signing certificates and certificate deployment and certificate activation.

With the buttons in the bottom, you can control the bulk processing and call some special functions for all devices in the device list.

Create a new Certificate Authority (CA)

Creating a CA in MOSCA is really simple. Just enter at least a CA Common Name, a Validity and a PrivateKey Password.


CA Base Configuration

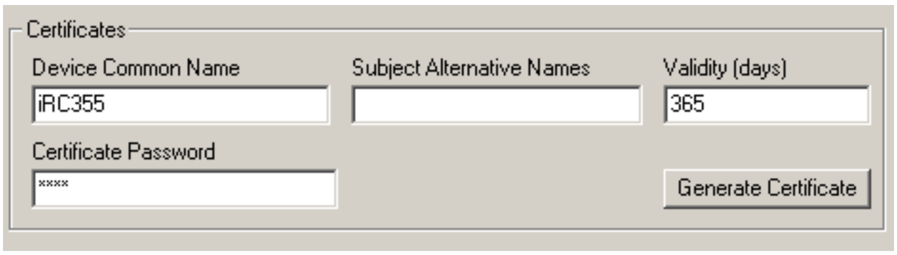
CA Common Name	Validity (days)	PrivateKey Password
Test	1825	****
PrivateKey Length	Organisation	Organisation Unit
2048		
Location	State	Country

Then just click “Create CA” and MOSCA is generating the private and public CA keys. All keys and certificates from the created CA will be stored in a subdirectory named “CA_NameOfYourCA” in the MOSCA directory. In our example the subdirectory got the name “CA_Test”. For a fast access to this folder you can press the button “Open CA Folder” in the top of the GUI.

Manual creation of a device certificate

After you created or selected a CA you can create password protected device certificates containing a new generated private key. These certificates (in PKCS#12 or *.pfx format) can be imported e.g. in Canon printing devices.

Enter at least the certificate details “Device Common Name”, “Validity” and “Certificate Password”. You may also add comma separated “Subject Alternative Names” if your certificate should match more than the Common Name.


Certificates

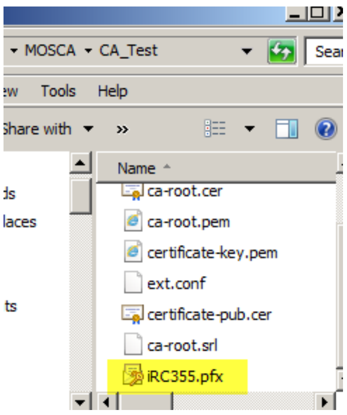
Device Common Name: iRC355

Subject Alternative Names:

Validity (days): 365

Certificate Password: ****

Then press “Generate Certificate”. The created certificate will be named like your entered Common Name with the suffix “*.pfx” and can be found in the CA subdirectory.



ca-root.cer
ca-root.pem
certificate-key.pem
ext.conf
certificate-pub.cer
ca-root.srl
iRC355.pfx

The Device List

If you want to use the bulk functions of MOSCA you need to create or load a device list with all necessary information:

IP/Hostname
IP address or the hostname of the device. In the most cases this is the same as the CommonName.

CommonName
The Common Name (CN) of the device with the full domain name (FQDN).

CertificateName
The name of the certificate file to be stored or to be loaded. The file suffix may change during the signing process (*.csr > *.cer) automatically.

802.1X Login
The login user information with which a device tries to authenticate during the 802.1X connection process. This information is only needed when activating EAP-TLS for 802.1X.

802.1X Radius
The name of the RADIUS server used in the 802.1X connection process. This information is only needed when activating EAP-TLS for 802.1X.

Status
The status of the currently running process for this device.

On
All devices in the list are periodically checked via SNMP for their online state. An “X” in this field indicates that the device is online.

AutoCommand
Shows the actual command MOSCA is trying to perform on this device.

SAN’s (comma sep.)
Shows all Subject Alternative Names (SANs) to be used. The values are comma separated.

CertPass
The certificate password if passwords per device are required for the certificate generation.

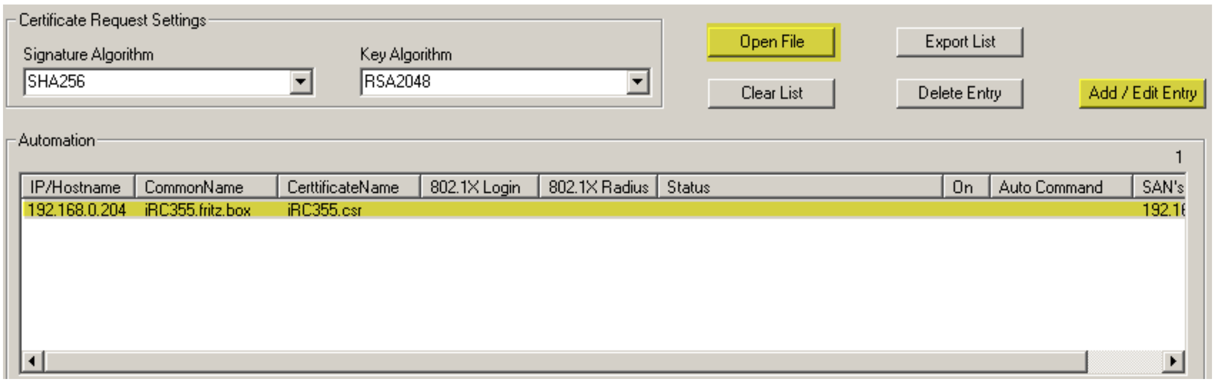
Add / Edit devices

To add devices to the device list you can import a CSV file containing all wanted information in the same order as the columns in the device list.
Please refer to the sample files in the folder “Sample CSV Files”.

IP/Hostname;CommonName;Filename

At least the first 3 columns have to be filled.

You can Drag&Drop the file on the device list or press the “Open File” button to choose a file.



Certificate Request Settings

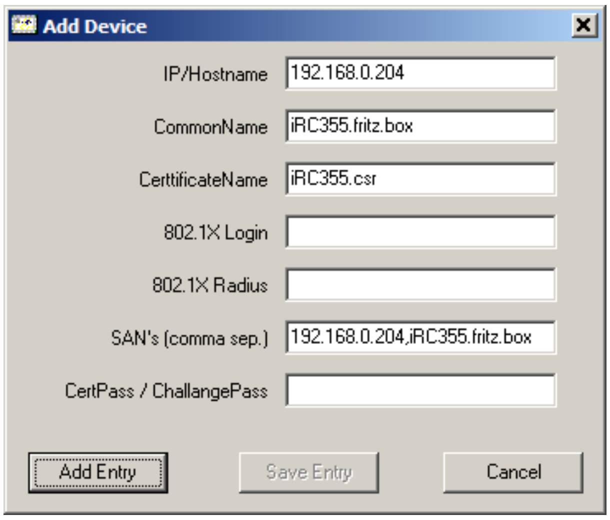
Signature Algorithm: SHA256

Key Algorithm: RSA2048

IP/Hostname	CommonName	CertificateName	802.1X Login	802.1X Radius	Status	On	Auto Command	SAN's
192.168.0.204	iRC355.fritz.box	iRC355.csr						192.16

“Clear List” empties the list.
“Export List” will let you save your list to a file.
“Delete Entry” will remove the selected entry from the list.

You can also press “Add / Edit Entry” to create or edit an entry in the device list.



Add Device

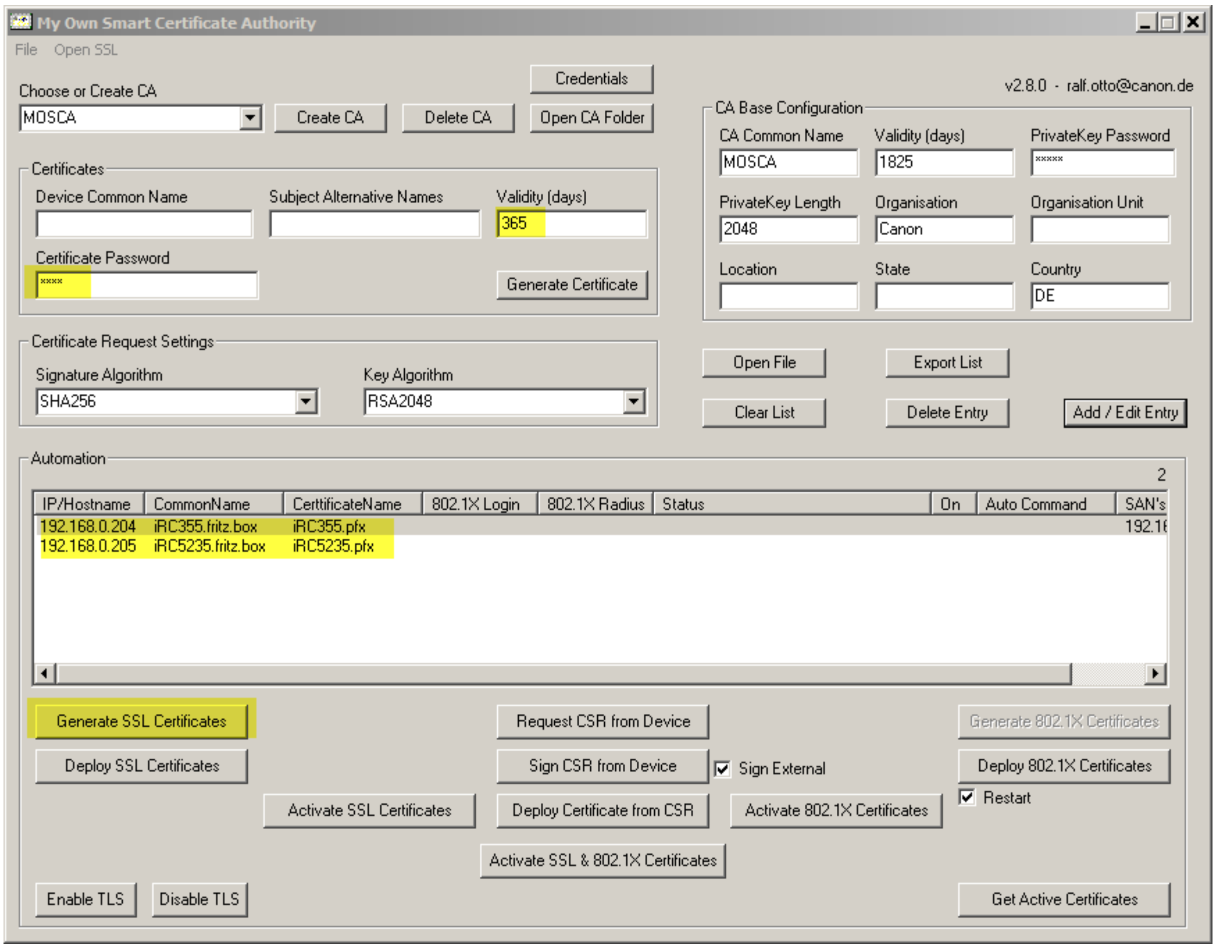
IP/Hostname	192.168.0.204
CommonName	iRC355.fritz.box
CerttificateName	iRC355.csr
802.1X Login	
802.1X Radius	
SAN's (comma sep.)	192.168.0.204,iRC355.fritz.box
CertPass / ChallangePass	

Press “Add Entry” to create a new entry or to clone an existing entry.
“Save Entry” will apply the changes to the device list. Please regard that all changes to the list will not be stored to an imported file until you use the button “Export List”.
“Cancel” will discard the changes.

Automation for Certificates

Bulk Generation of Device Certificates

To generate more device certificates in MOSCA at once, you have to fill the device list with the needed information. You have to provide the certificate validity and a certificate password. If you want to use a certificate password per device, you have to fill these passwords in the device list.



v2.8.0 - ralf.otto@canon.de

Choose or Create CA: MOSCA

Certificates
- Device Common Name:
- Subject Alternative Names:
- Validity (days): 365
- Certificate Password: ****

CA Base Configuration
- CA Common Name: MOSCA
- Validity (days): 1825
- PrivateKey Password: *****
- PrivateKey Length: 2048
- Organisation: Canon
- Organisation Unit:
- Location:
- State:
- Country: DE

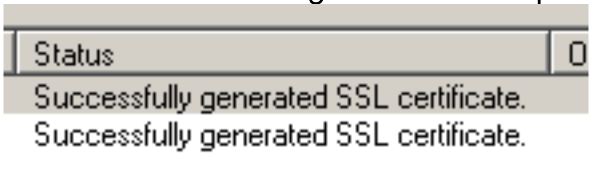
Certificate Request Settings
- Signature Algorithm: SHA256
- Key Algorithm: RSA2048

Automation 2

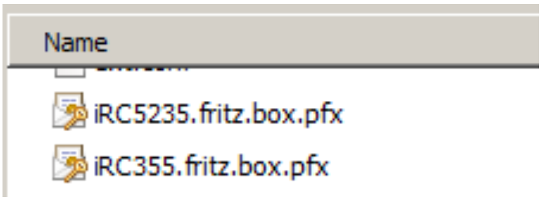
IP/Hostname	CommonName	CertificateName	802.1X Login	802.1X Radius	Status	On	Auto Command	SAN's
192.168.0.204	iRC355.fritz.box	iRC355.pfx						192.16
192.168.0.205	iRC5235.fritz.box	iRC5235.pfx						

[✓] Sign External
[✓] Restart

Then press “Generate SSL Certificate”.
All certificates will be generated via OpenSSL and stored in the CA folder:



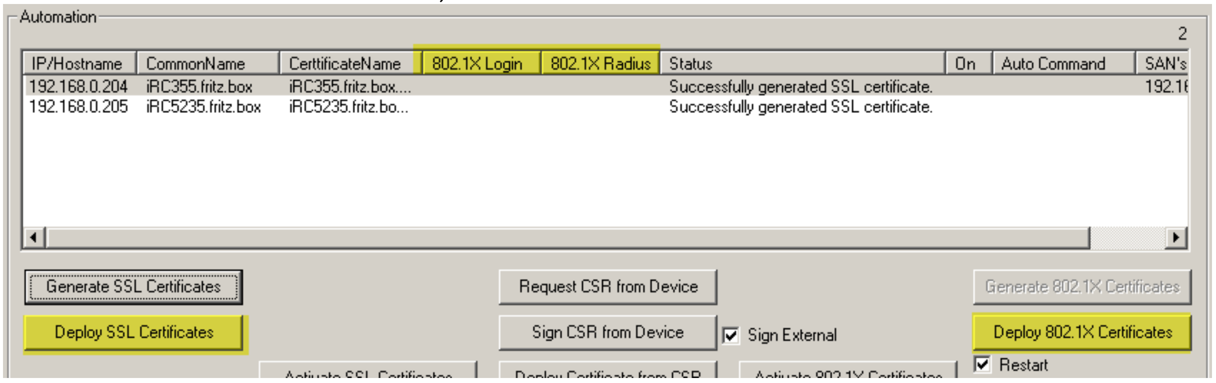
Status	O
Successfully generated SSL certificate.	
Successfully generated SSL certificate.	



Name
iRC5235.fritz.box.pfx
iRC355.fritz.box.pfx

Bulk Deployment of Device Certificates

If you have generated certificates in MOSCA or externally generated certificates in PKCS#12 (or *.pfx) format, you can deploy and activate these certificates on the Canon devices. Please ensure, that the certificate files are in the CA folder.



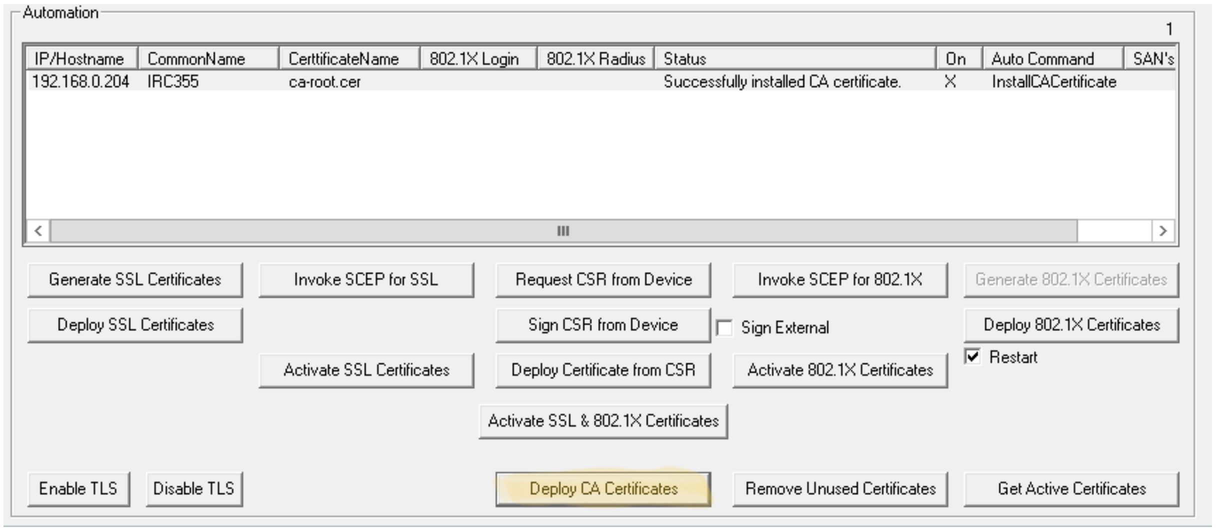
IP/Hostname	CommonName	CertificateName	802.1X Login	802.1X Radius	Status	On	Auto Command	SAN's
192.168.0.204	iRC355.fritz.box	iRC355.fritz.box....			Successfully generated SSL certificate.			192.16
192.168.0.205	iRC5235.fritz.box	iRC5235.fritz.bo...			Successfully generated SSL certificate.			

If you want to deploy certificates for TLS / SSL press “Deploy SSL Certificate”.
The certificates will be deployed, activated and SSL will be turned on on the devices if it is not alredy active. Finally the devices will be restarted.

If you want to deploy certificates for 802.1X and you want EAP-TLS to be activated on the devices you have to provide the additional information for “802.1X Login” and “802.1X Radius” in the device list. Then press “Deploy 802.1X Certificates”. The certificates will be deployed, activated and EAP-TLS for 802.1X will be turned on on the devices if it is not alredy active. If the checkbox “Restart” is checked, the devices will be finally restarted.

Bulk Deployment of CA Certificates

If you have want to deploy root certificates of a Certificate Authority, you can deploy these certificates on the Canon devices. Please ensure, that the certificate files are in the CA folder.

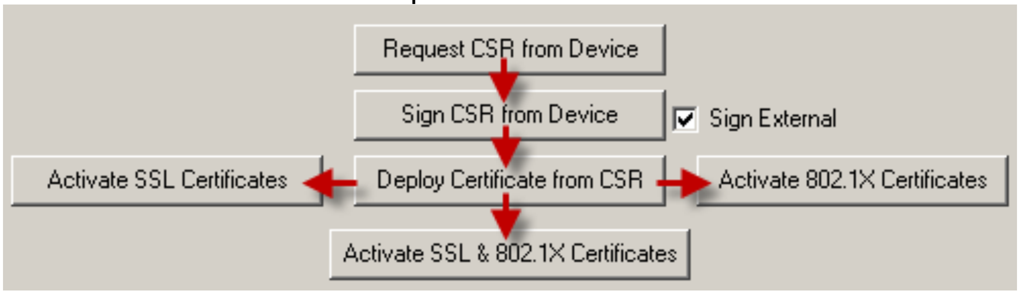


Automation
1
Generate SSL Certificates
Invoke SCEP for SSL
Request CSR from Device
Invoke SCEP for 802.1X
Generate 802.1X Certificates
Deploy SSL Certificates
Sign CSR from Device
Sign External
Deploy 802.1X Certificates
Restart
Activate SSL Certificates
Deploy Certificate from CSR
Activate 802.1X Certificates
Activate SSL & 802.1X Certificates
Enable TLS
Disable TLS
Deploy CA Certificates
Remove Unused Certificates
Get Active Certificates

IP/Hostname	CommonName	CertificateName	802.1X Login	802.1X Radius	Status	On	Auto Command	SAN's
192.168.0.204	IRC355	ca-root.cer			Successfully installed CA certificate.	X	InstallCACertificate	

Automation for Certificates with CSR

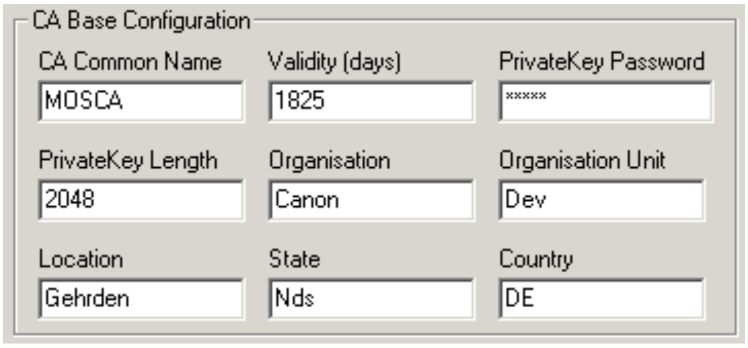
MOSCA can generate a Certificate Signing Request (CSR) on the Canon device, sign this request in OpenSSL or against an external Certificate Authority (CA) and deploy and activate the certificates on the Canon devices. This section will describe the functions offered for that process.


Request CSR from Device
Sign CSR from Device
Sign External
Activate SSL Certificates
Deploy Certificate from CSR
Activate 802.1X Certificates
Activate SSL & 802.1X Certificates

Request CSR's from devices

To request a CSR from a Canon device you have to provide at least the first 3 columns in the device list.

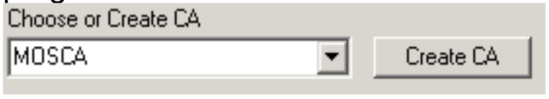
All other information to request the CSR like "Organisation", "Organisation Unit", etc. will be taken from the text fields of the CA Base Configuration section.



CA Base Configuration

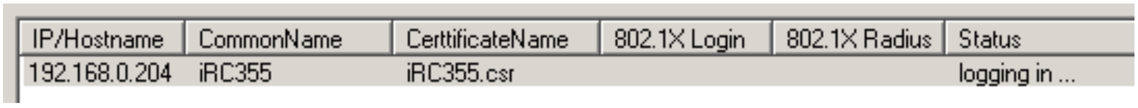
CA Common Name	Validity (days)	PrivateKey Password
MOSCA	1825	*****
PrivateKey Length	**Organisation**	**Organisation Unit**
2048	Canon	Dev
Location	**State**	**Country**
Gehrden	Nds	DE

You can fill these fields manually or choose or create a CA in the top left of the program window.



Choose or Create CA: MOSCA

Press the button "Request CSR from Device" and MOSCA will try to request a CSR for each online device in the device list. During the request the actual request state will be shown in the Status column of the device list.



IP/Hostname	CommonName	CertificateName	802.1X Login	802.1X Radius	Status
192.168.0.204	iRC355	iRC355.csr			logging in ...

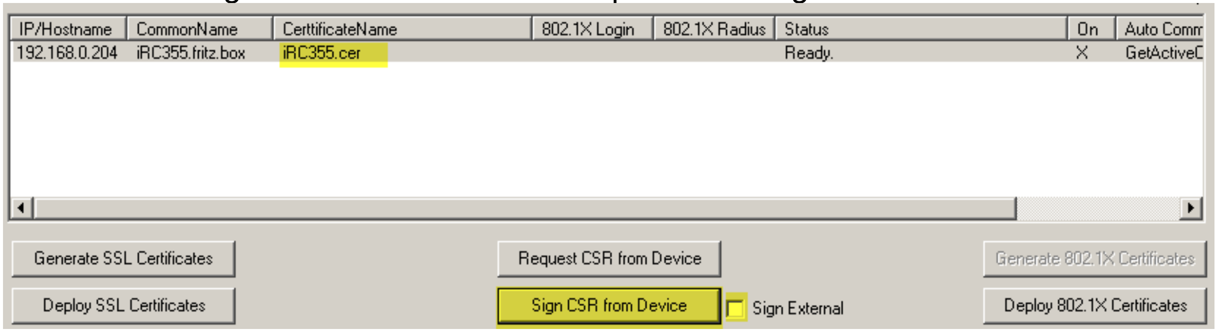
If the requests were successful, the CSR's will be saved in the CA subdirectory.

Now you can decide what to do next:
- Sign the CSR's with OpenSSL in MOSCA, or
- copy the CSR's to any place you like to sign them manuelly in your CA and copy the signed certificates back to the CA folder, or
- sign the CSR's external with a shell command against your CA.

Sign CSR's from devices

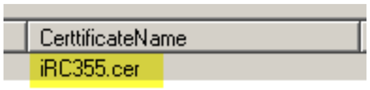
With OpenSSL

If you want to sign a Certificate Signing Request (CSR) in MOSCA via OpenSSL just uncheck the "Sign External" checkbox and press the "Sign CSR from Device" button.



IP/Hostname	CommonName	CerttificateName	802.1X Login	802.1X Radius	Status	On	Auto Comm
192.168.0.204	iRC355.fritz.box	iRC355.cer			Ready.	X	GetActiveC

The CSR's will be signed and the signed certificates will be stored in the CA folder as *.cer" files.
The suffix of the "CertificateName" entries in the device list will be changed from "*.csr" to "*.cer".



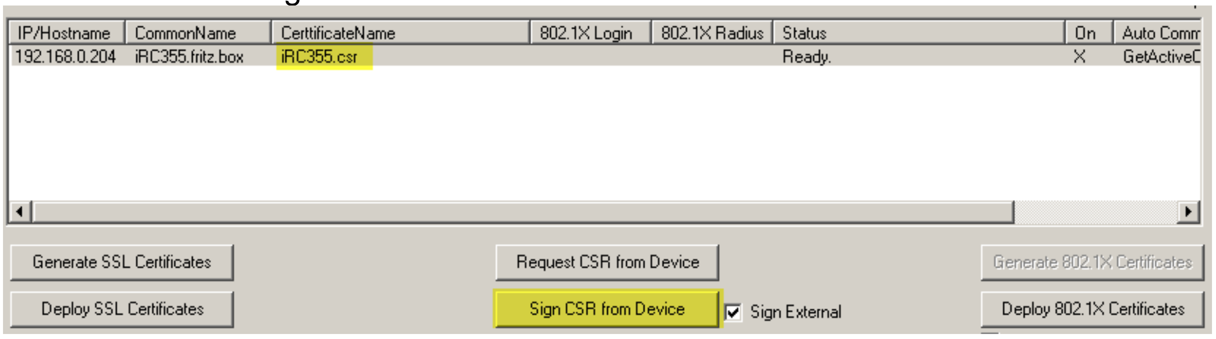
CerttificateName
iRC355.cer

With an external Certificate Authority

If you want to sign a CSR with an external Certificate Authority (CA) you have to check the "Sign External" checkbox.

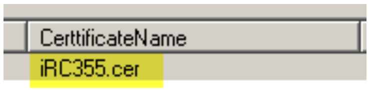
Please refer to the section "Base Configuration / ExternalSigning" and "Requirements / External Signing" to learn more about the configuration and the requirements to do external signing and the use of the Subject Alternative Names (SANs) during this process.

Press the button "Sign CSR from Device".



IP/Hostname	CommonName	CerttificateName	802.1X Login	802.1X Radius	Status	On	Auto Comm
192.168.0.204	iRC355.fritz.box	iRC355.csr			Ready.	X	GetActiveC

The CSR's will be signed and the signed certificates will be stored in the CA folder as *.cer" files.
The suffix of the "CertificateName" entries in the device list will be changed from "*.csr" to "*.cer".



CerttificateName
iRC355.cer

Deploy Certificates from CSR's

After the signing process the certificates can be deployed on the devices. With this function the certificates will just be stored on the devices and have be activated in the last step of the deployment process.

To deploy the certificates just press the "Deploy Certificate from CSR" button and MOSCA will deploy the certificates on the devices in the device list.

Activate Certificates from CSR's

After the deployment the certificates can be activated on the devices. You have to chosse how the certificate should be used on the device:

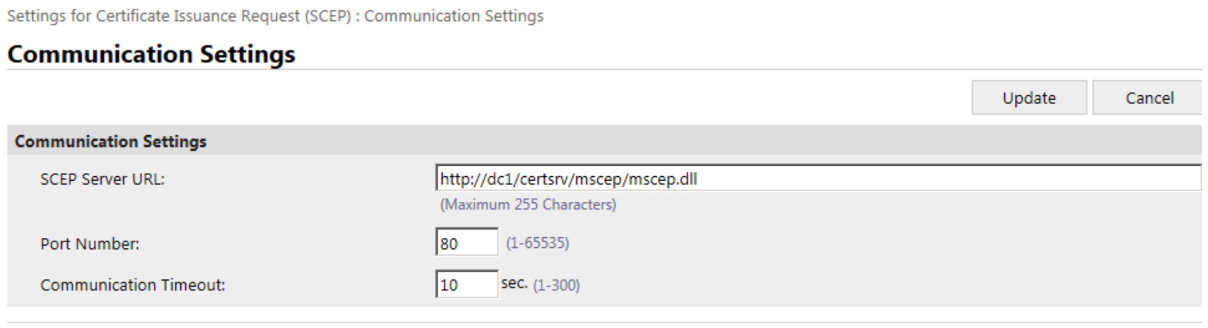
- If you want to use the certificate for SSL only press the "Activate SSL Certificate" button. MOSCA will activate the certificate for TSL / SSL use and activte SSL in general on the device if not already done. Finally the device will be restarted.

- If you want to use the certificate for 802.1X and want to activate EAP-TLS you have to provide the additional information for "802.1X Login" and "802.1X Radius" in the device list. Then press the "Activate 802.1X Certificate" button and MOSCA will activate the certificate for 802.1X use and activte EAP-TLS in general on the device if Login and RADIUS server name are given. Finally the device will be restarted.

- If you want to use the certificate for both SSL and 802.1X (dual use) set the information as described before and press the "Activate SSL & 802.1X Certificate" button. MOSCA will then do all the steps for SSL and 802.1X certificate use at once.

Invoke SCEP

Many Canon devices are able to obtain their certificates using the Simple Certificate Enrollment Protocol (SCEP). Here is a sample of a correct Canon device SCEP configuration:



Settings for Certificate Issuance Request (SCEP) : Communication Settings

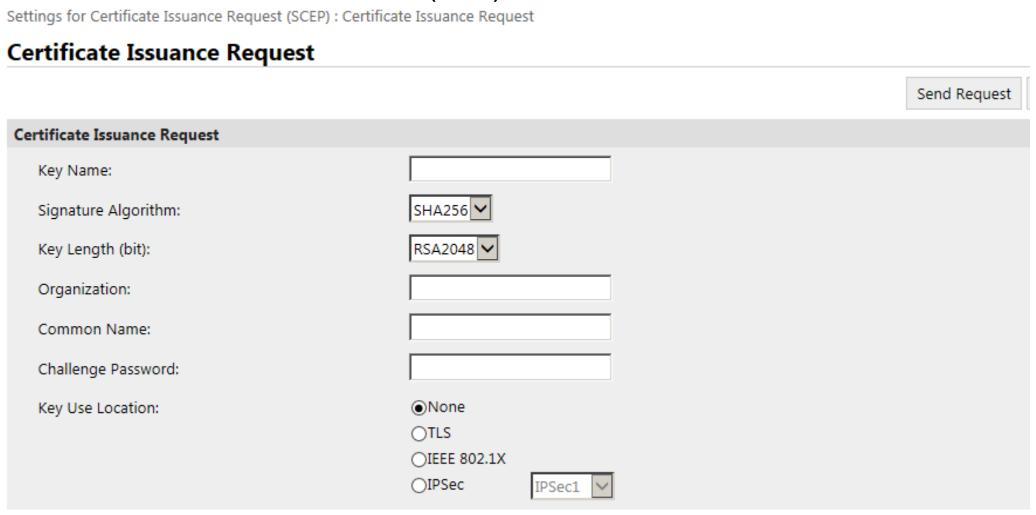
Communication Settings

Communication Settings

SCEP Server URL:	http://dc1/certsrv/mscep/mscep.dll (Maximum 255 Characters)
Port Number:	80 (1-65535)
Communication Timeout:	10 sec. (1-300)

There are two possibilities to get a certificate when SCEP is configured on the devices. One is to set up a timer for a recurring Certificate Issuance Auto Request. The problem with this method is, that there has be be an enduring challenge password on the SCEP server. This password has to be valid for at least one life cycle of a certificate to ensure that the next certificate can still be requested. Another problem of this method is, that only one timer with one type of certificate (SSL OR 802.1X) can be set up. So Auto-SCEP on Canon devices is only possible for one certificate usage.

The other method to use SCEP is to invoke a one time Certificate Issuance Request on the Remote User Interface (RUI) of the device.



Settings for Certificate Issuance Request (SCEP) : Certificate Issuance Request

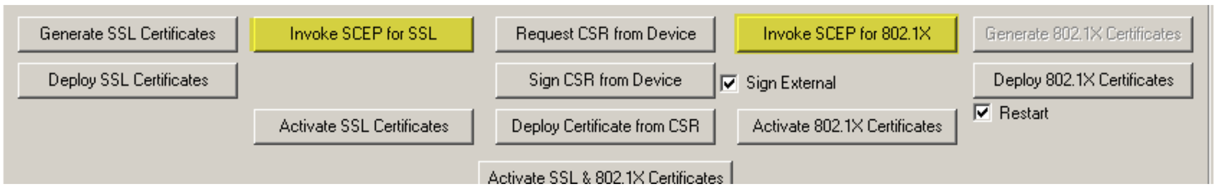
Certificate Issuance Request

Certificate Issuance Request

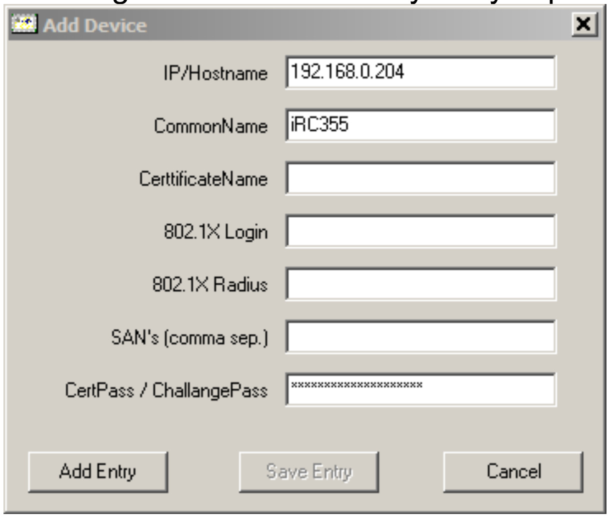
Key Name:	
Signature Algorithm:	SHA256
Key Length (bit):	RSA2048
Organization:	
Common Name:	
Challenge Password:	
Key Use Location:	[✓ None] TLS IEEE 802.1X IPSec IPSec1

Each time a certificate is requested, a new challenge password and the certificate type can be set up. Unfortunately this method requires a lot of manual work on the devices RUI.

This is where MOSCA comes in with its “Invoke SCEP” funtion:


Generate SSL Certificates
Invoke SCEP for SSL
Request CSR from Device
Invoke SCEP for 802.1X
Generate 802.1X Certificates
Deploy SSL Certificates
Sign CSR from Device
✓ Sign External
Deploy 802.1X Certificates
✓ Restart
Activate SSL Certificates
Deploy Certificate from CSR
Activate 802.1X Certificates
Activate SSL & 802.1X Certificates

To use Invoke SCEP, you have to fill the columns IP/Hostname, CommonName and ChallengePassword manually or by importing a file to the automation list.


Add Device

IP/Hostname	192.168.0.204
CommonName	iRC355
CerttificateName	
802.1X Login	
802.1X Radius	
SAN's (comma sep.)	
CertPass / ChallangePass	********************

Then just press “Invoke SCEP for SSL” or “Invoke SCEP for 802.1X” and MOSCA will try to invoke a SCEP request against the configured SCEP server. After a new certificate is obtained, MOSCA will restart the device to activate the new certificate.

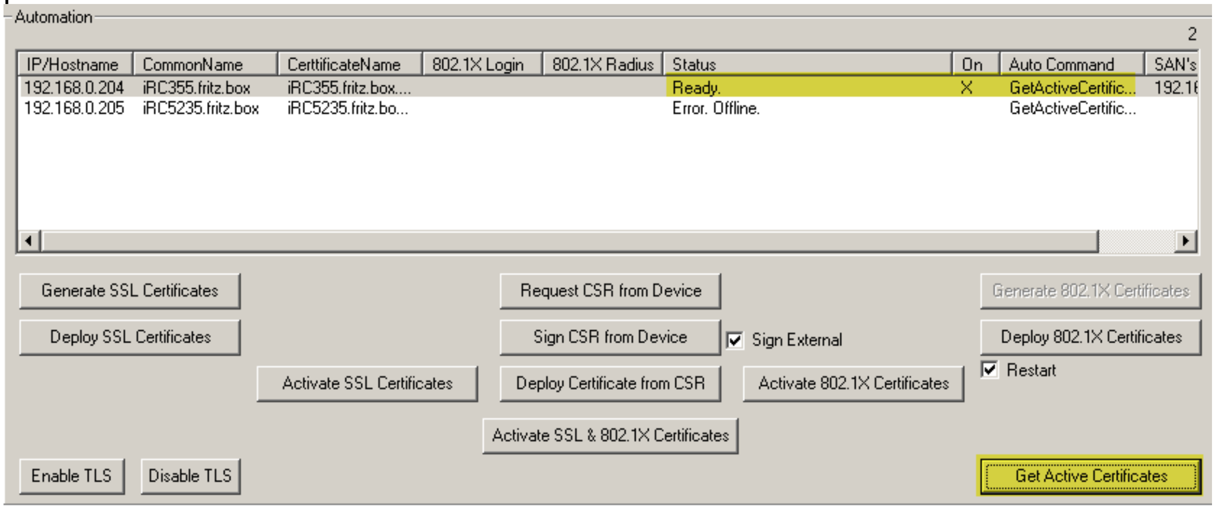
Automation for Special Functions

Control TLS

If you want to turn TLS on or off for all devices in the device list you can use the buttons “Enable TLS” or “Disable TLS”. TLS will be configured and the devies will be restarted.

Query Active Certificates

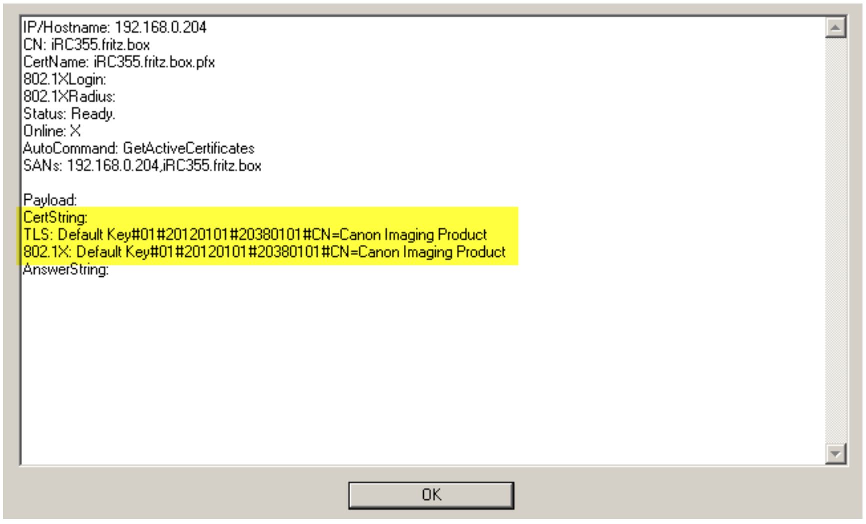
If you want to ask all devices for their active certificates for SSL and 802.1X, you can press the “Get Active Certificates” button. MOSCA will read all active certificates.



IP/Hostname	CommonName	CerttificateName	802.1X Login	802.1X Radius	Status	On	Auto Command	SAN's
192.168.0.204	iRC355.fritz.box	iRC355.fritz.box....			Ready.	X	GetActiveCertific...	192.16
192.168.0.205	iRC5235.fritz.box	iRC5235.fritz.bo...			Error. Offline.		GetActiveCertific...	

Automation
2
Generate SSL Certificates
Deploy SSL Certificates
Activate SSL Certificates
Request CSR from Device
Sign CSR from Device
Sign External
Deploy Certificate from CSR
Activate 802.1X Certificates
Generate 802.1X Certificates
Deploy 802.1X Certificates
Restart
Activate SSL & 802.1X Certificates
Enable TLS
Disable TLS
Get Active Certificates

You can display the active certificates by double clicking the device line in the device list:

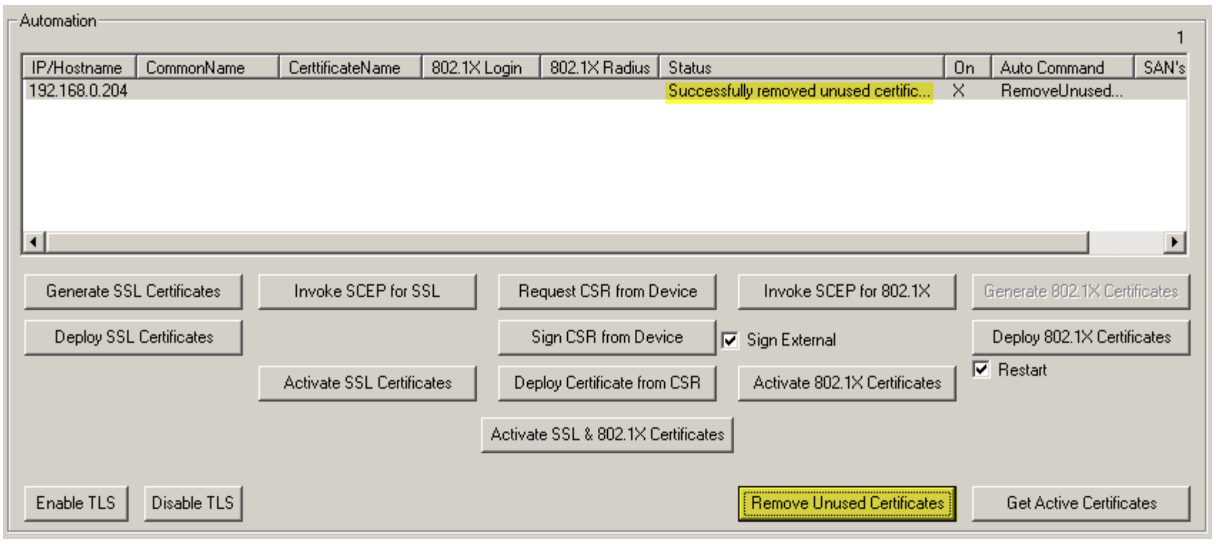


```
IP/Hostname: 192.168.0.204
CN: iRC355.fritz.box
CertName: iRC355.fritz.box.pfx
802.1XLogin:
802.1XRadius:
Status: Ready.
Online: X
AutoCommand: GetActiveCertificates
SANs: 192.168.0.204,iRC355.fritz.box

Payload:
CertString:
TLS: Default Key#01#20120101#20380101#CN=Canon Imaging Product
802.1X: Default Key#01#20120101#20380101#CN=Canon Imaging Product
AnswerString:
```

Remove Unused Certificates

The number of certificates that can be installed on a device is limited. Therefore, it might be necessary to remove older unused certificates before deploying new certificates. With the function “Remove Unused Certificates” MOSCA will search each device in the automation list for unused certificates and remove them from the device. Only the IP/Hostname column has to be filled to use this function.



IP/Hostname	CommonName	CertificateName	802.1X Login	802.1X Radius	Status	On	Auto Command	SAN's
192.168.0.204					Successfully removed unused certific...	X	RemoveUnused...	

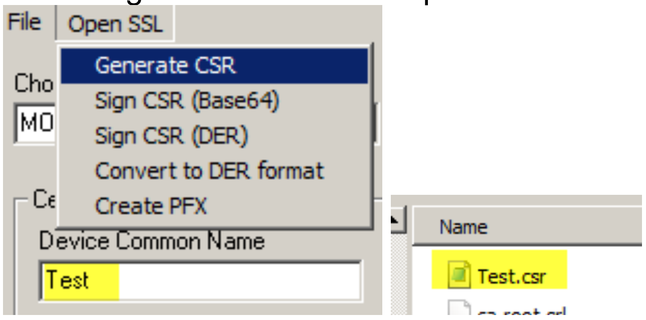
Generate SSL Certificates
Invoke SCEP for SSL
Request CSR from Device
Invoke SCEP for 802.1X
Generate 802.1X Certificates
Deploy SSL Certificates
Sign CSR from Device
Sign External
Deploy 802.1X Certificates
Restart
Activate SSL Certificates
Deploy Certificate from CSR
Activate 802.1X Certificates
Activate SSL & 802.1X Certificates
Enable TLS
Disable TLS
Remove Unused Certificates
Get Active Certificates

Quick OpenSSL Commands

For testing purpose you can call some OpenSSL commands via the “Open SSL” menu in MOSCA. All certificate files used by these commands have to exist or will be put in the folder of the chosen CA.

Generate CSR

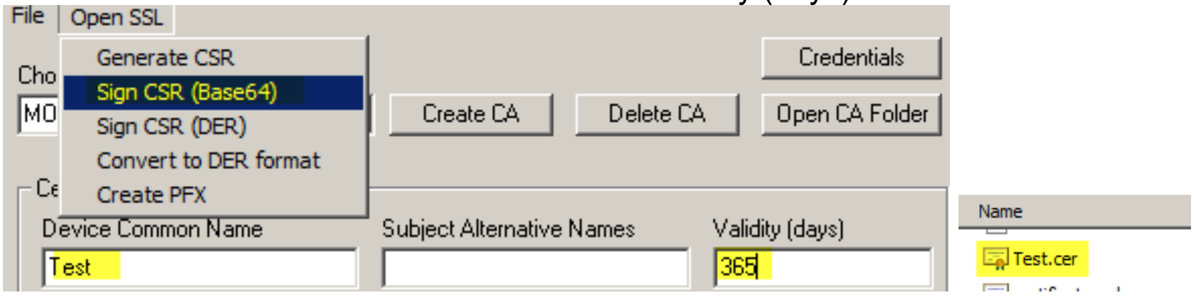
This will generate a CSR in OpenSSL. You have to enter a “Device Common Name”.



Device Common Name
Test

Sign CSR

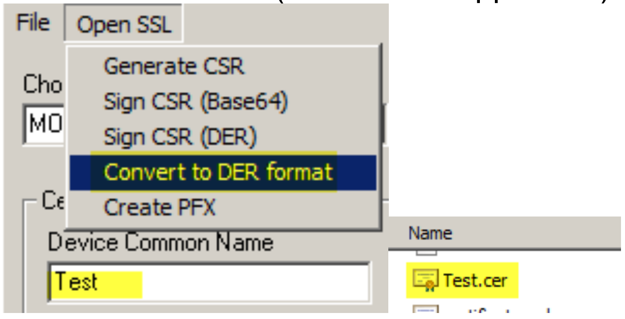
With the functions „Sign CSR (Base64)” and “Sign CSR (DER)” you can sign a certificate request file with OpenSSL in the CA folder either in Base64 format or in the binary DER format.
Just enter the “Devie Common Name” and a “Validity (days)”.



Device Common Name
Test
Subject Alternative Names
Validity (days)
365

Convert to DER format

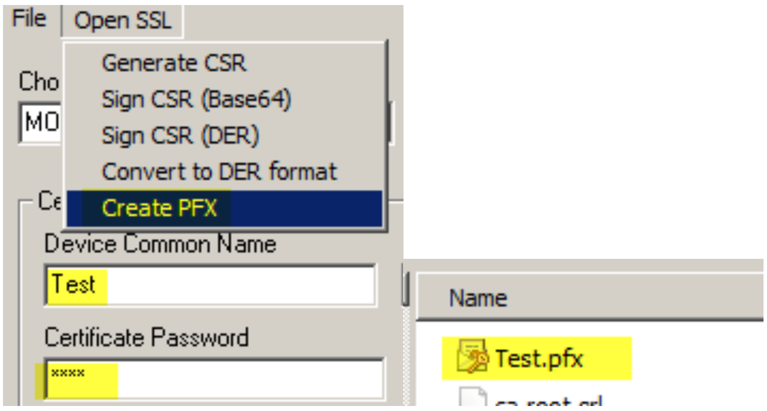
With this command a Base64 encode certificate will be converted to the binary DER format. You have to provide the “Device Common Name” for the certificate file name without extensions (*.cer will be appended).



Device Common Name
Test

Create PFX

If you have created a CSR in MOSCA and signed it externally in a CA you got a certificate in *.cer format. If you want to use this certificate on a Canon device, you have to convert in PKCS#12 (or *.pfx) format. Just enter the “Device Common Name” for the certificate file name without extensions (*.cer will be appended) and a “Certificate Password”.



Device Common Name: Test

Certificate Password: ****

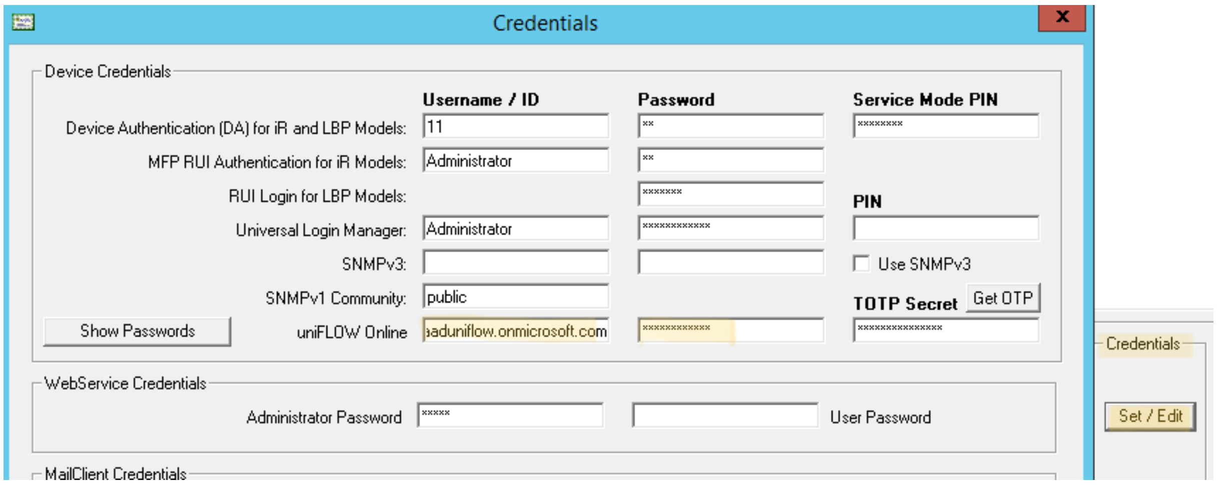
Test.pfx

Devices in uniFLOW Online

If you have devices, that are attached to a uniFLOW Online tenant, you have to gain access for MOSCA to uniFLOW Online to log on to the devices.

Account without MFA

If you have an account **without** multi factor authentication (MFA) you can just enter the username and password for a device manager account in the “Credentials” section by pressing “Set / Edit”.



Credentials

Device Credentials

	Username / ID	Password	Service Mode PIN
Device Authentication (DA) for iR and LBP Models:	11	**	********
MFP RUI Authentication for iR Models:	Administrator	**	
RUI Login for LBP Models:		*******	PIN
Universal Login Manager:	Administrator	************	
SNMPv3:			☐ Use SNMPv3
SNMPv1 Community:	public		TOTP Secret Get OTP
uniFLOW Online	ıaduniflow.onmicrosoft.com	************	***************

Show Passwords

WebService Credentials

Administrator Password ***** User Password

MailClient Credentials

Credentials

Set / Edit

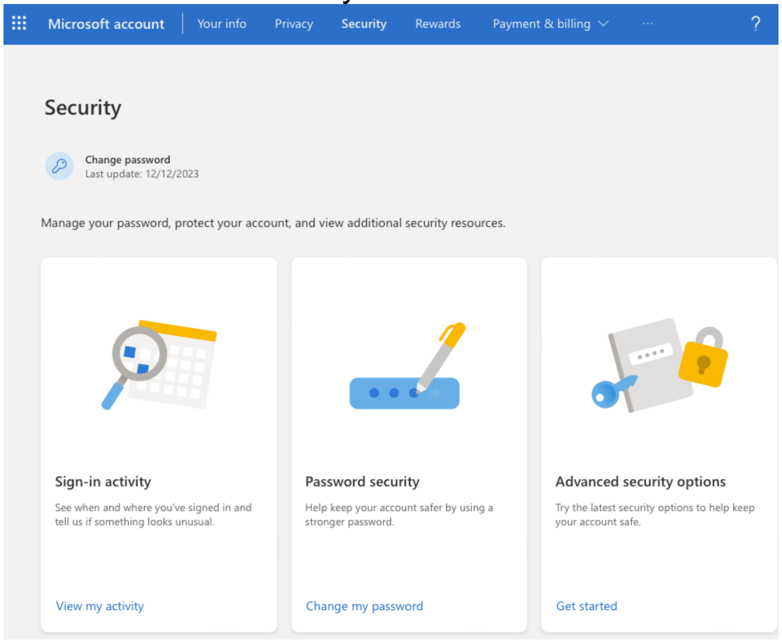
Account with MFA

If you want to use an account **with** MFA, you also have to enter username and password, but in this case MOSCA additionally needs the TOTP (Time based One Time Password) Secret. MOSCA can then generate a code, if a multi factor authentication is requested by the device login.

The TOTP secret can be accessed when creating or editing the multi factor authentication for an account. Here is a sample for activating MFA for an MS online account.

Login to your MS online account and choose “My Microsoft account”.

Then select the “Security” tab.



Security

Change password
Last update: 12/12/2023

Manage your password, protect your account, and view additional security resources.

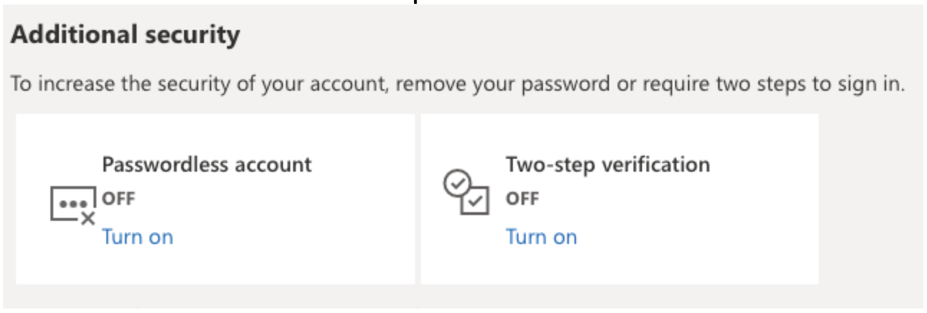
Sign-in activity
See when and where you've signed in and tell us if something looks unusual.
View my activity

Password security
Help keep your account safer by using a stronger password.
Change my password

Advanced security options
Try the latest security options to help keep your account safe.
Get started

Choose “Advances security options”.

Select “Turn on” for “Two-step-verification”:



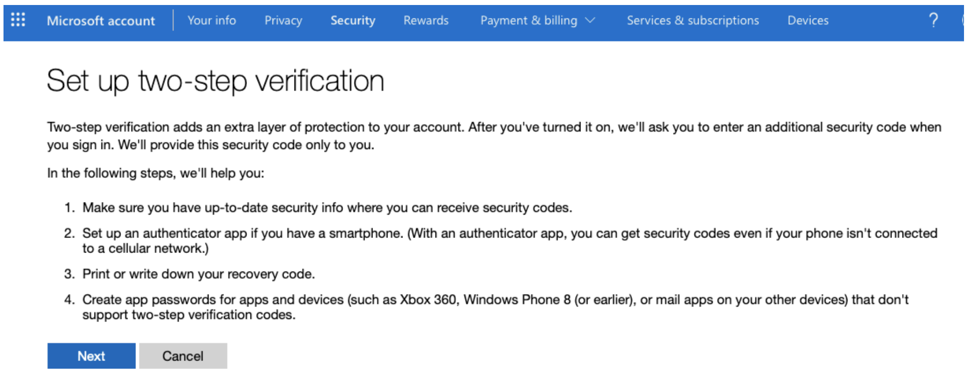
Additional security

To increase the security of your account, remove your password or require two steps to sign in.

Passwordless account
OFF
Turn on

Two-step verification
OFF
Turn on

Click “Next”



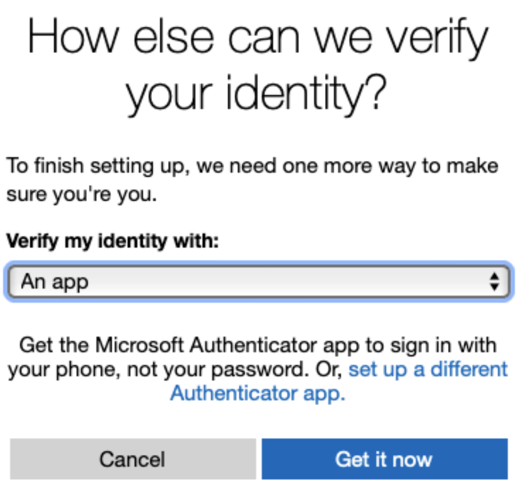
Set up two-step verification

Two-step verification adds an extra layer of protection to your account. After you've turned it on, we'll ask you to enter an additional security code when you sign in. We'll provide this security code only to you.

In the following steps, we'll help you:

1. Make sure you have up-to-date security info where you can receive security codes.
2. Set up an authenticator app if you have a smartphone. (With an authenticator app, you can get security codes even if your phone isn't connected to a cellular network.)
3. Print or write down your recovery code.
4. Create app passwords for apps and devices (such as Xbox 360, Windows Phone 8 (or earlier), or mail apps on your other devices) that don't support two-step verification codes.

For the second factor choose “An app” and press on “set up a different Authenticator app”.



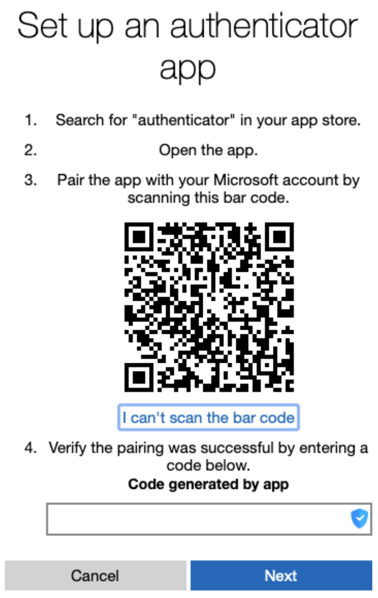
How else can we verify your identity?

To finish setting up, we need one more way to make sure you're you.

Verify my identity with:

An app

Get the Microsoft Authenticator app to sign in with your phone, not your password. Or, set up a different Authenticator app.



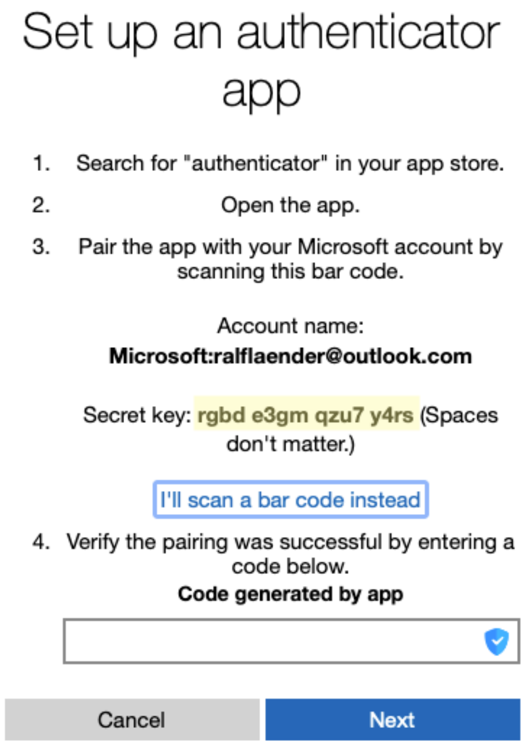
Set up an authenticator app

1. Search for "authenticator" in your app store.
2. Open the app.
3. Pair the app with your Microsoft account by scanning this bar code.

4. Verify the pairing was successful by entering a code below.

Code generated by app

When the barcode is shown, you can use it with your common authenticator app, but you also need to choose “I can’t scan the bar code” to show the TOPT secret key behind the bar code.



Set up an authenticator app

1. Search for "authenticator" in your app store.
2. Open the app.
3. Pair the app with your Microsoft account by scanning this bar code.

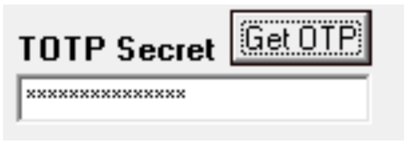
Account name:
Microsoft:ralflaender@outlook.com

Secret key: rgbd e3gm qzu7 y4rs (Spaces don't matter.)

4. Verify the pairing was successful by entering a code below.

Code generated by app

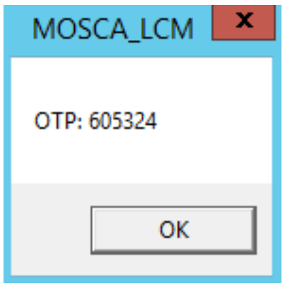
Copy the “Secret key” and enter it without the spaces in the “TOTP Secret” field in the credentials form in MOSCA-LCM.



TOTP Secret

****

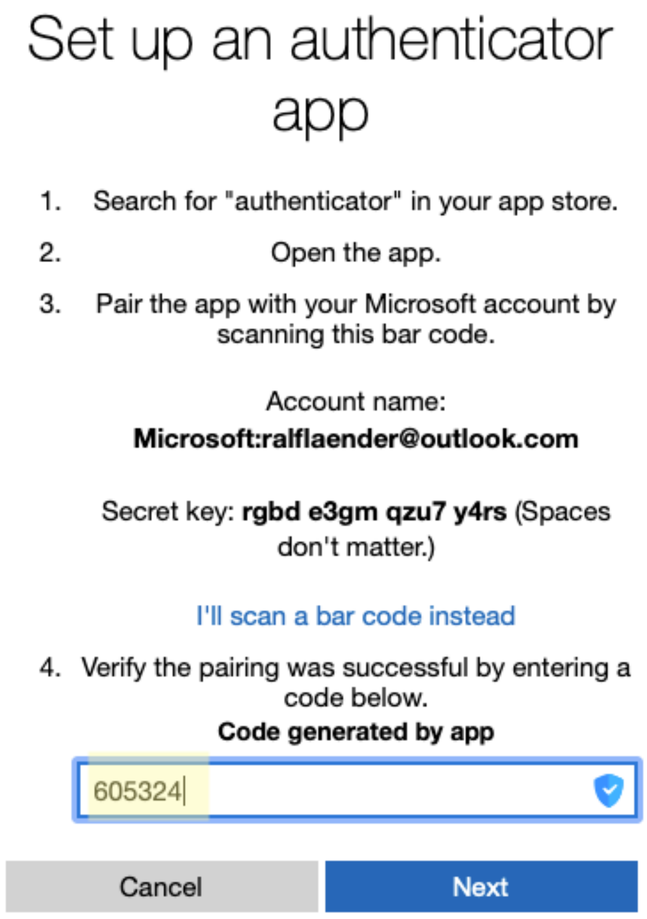
You can then click the “Get OTP” button to get an OTP generated by MOSCA.



MOSCA_LCM

OTP: 605324

Enter this code to finish the MFA set-up.



Set up an authenticator app

1. Search for "authenticator" in your app store.
2. Open the app.
3. Pair the app with your Microsoft account by scanning this bar code.

Account name:
Microsoft:ralflaender@outlook.com

Secret key: **rgbd e3gm qzu7 y4rs** (Spaces don't matter.)

I'll scan a bar code instead

4. Verify the pairing was successful by entering a code below.

Code generated by app

605324

The account used, has to be set as device manager in uniFLOW Online.

MOSCA-LCM will then be able to generate one time passwords (OTPs) and login to any devices connected to this uniFLOW Online tenant.

MOSCA Web API

MOSCA offers a web API to control the MOSCA functions from external programs or scripts. To enable the API you have to set a value for the WebServerPort in the section [Settings] in the MOSCA.ini file. The default API port is 8080.
The MOSCA API is listening to POST commands containing the payload in JSON format. The replies from MOSCA are also in JSON format.

Post the JSON formatted payload to the IP address of the server MOSCA is running on, followed by the specified API port.
Example: http://192.168.0.150:8080

General API answers

"OK"
The API command was ok and will be processed soon.

"Error. Request just exists."
There is still a pending request with the same command and CN (CommonName)

"Error."
The API command was faulty or a general error occurred.

"Queued."
The command was queued in the command list of MOSCA for processing.

"Pending."
The command is just processing and a result will be available soon.

"Ready."
The command was processed successfully.

Request CSR

This command lets MOSCA request a CSR from the given device with all given data of the request.

Post command:
{"MOSCA":"RequestCSR","Payload":{"IP":"192.168.0.204","cn":"iRC355","o":"Canon" ,"ou":"SBG","l":"Gehrden","s":"Nds","c":"DE"}}

Answer format:
{"MOSCA":"RequestCSR","Payload":{"cn":"iRC355","Status":"OK"}}

Possible Status answers:
"OK"
"Error. Request just exists."
"Error."

The status and finally the certificate request can also be requested.

Post command:
{"MOSCA":"GetCSR","Payload":{"cn":"iRC355"}}

Answer format:
{"MOSCA":"RequestCSR","Payload":{"cn":"iRC355","Status":"No request found.","CSR":""}}

Possible Status answers:
"No request found."
"Queued."
"Pending."
"Ready."
"Error."

Possible CSR answers are:
""
"----BEGIN CERTIFICATE REQUEST-----MI…"

Deploy Certificate from CSR

This command lets MOSCA deploy the certificate from a signed CSR from the given device with all given data of the request.

Post command:
{"MOSCA":"DeployCertificateFromCSR","Payload":{"IP":"192.168.0.204","cn":"iRC355","cert":"-----BEGIN CERTIFICATE-----MIIDGTC…-----END CERTIFICATE-----"}}

Answer format:
{"MOSCA":" DeployCertificateFromCSR","Payload":{"cn":"iRC355","Status":"OK"}}

Possible Status answers:
“OK”
“Error. Request just exists.”
“Error.”

The status of the certificate deployment can also be requested.

Post command:
{"MOSCA":"GetDeploymentState","Payload":{"cn":"iRC355"}}

Answer format:
{"MOSCA":"GetDeploymentState","Payload":{"cn":"iRC355","Status":"Ready."}}

Possible Status answers:
“No request found.”
“Queued.”
“Pending.”
“Ready.”
“Error.”

Activate 802.1X Certificate

This command lets MOSCA activate an installed certificate for 802.1X communication. The certificate will be selected for 802.1X, EAP-TLS will be activated and device will be rebooted.

Post command:
{"MOSCA":"Activate8021XCertificate","Payload":{"IP":"192.168.0.204","cn":"iRC355", "certname":"IRC355","loginname":"Canon","radiusname":"Radius"}}

Remark: The Radius Server Name (radiusname) is optional.

Answer format:
{"MOSCA":"Activate8021XCertificate","Payload":{"cn":"iRC355","Status":"OK"}}

Possible Status answers:
“OK”
“Error. Request just exists.”
“Error.”

The status of the certificate activation can also be requested.

Post command:
{"MOSCA":"Get8021XActivationState","Payload":{"cn":"iRC355"}}

Answer format:
{"MOSCA":"Get8021XActivationState","Payload":{"cn":"iRC355","Status":"Ready."}}

Possible Status answers:
“No request found.”
“Queued.”
“Pending.”
“Ready.”
“Error.”

Request Active Certificates

This command lets MOSCA request the active certificates for TLS and 802.1X from the given device.

Post command:
{"MOSCA":"RequestActiveCertificates","Payload":{"IP":"192.168.0.204","cn":"iRC355"}}

Answer format:
{"MOSCA":"RequestActiveCertificates","Payload":{"cn":"iRC355","Status":"OK"}}

Possible Status answers:
“OK”
“Error. Request just exists.”
“Error.”

The status and finally the certificate details can also be requested.

Post command:
{"MOSCA":"GetActiveCertificates","Payload":{"cn":"iRC355"}}

Answer format:
{"MOSCA":"GetActiveCertificates","Payload":{"cn":"Lexi","Status":"Ready.","TLSCert":{"Name":"DefaultKey",”SerialNumber”:”010203”,"ValidFrom":"2012010","ValidTo":"2038010","Issuer":"CN=CanonImagingProduct"},"802.1XCert":{"Name":"iRC355","ValidFrom":"2019031","ValidTo":"2020031","Issuer":"DC=local, DC=mydomain, CN=MOSCA"}}}

Possible Status answers are:
“No request found.”
“Queued.”
“Pending.”
“Ready.”
“Error.”

Certificate Details are:

Name: The name of the active certificate
SerialNumber: The Serial number of the active certificate
ValidFrom: The start date of the certificates valid period
ValidTo: The end date of the certificates valid period
Issuer: The distinguished names of the certificate issuer

Possible TLSCert answers are:
“”
{"Name":"DefaultKey",”SerialNumber”,”010203”,"ValidFrom":"2012010","ValidTo":"2038010","Issuer":"CN=CanonImagingProduct"}

Possible 802.1XCert answers are:

“”

{"Name":"iRC355",”SerialNumber”,”010203”,"ValidFrom":"2019031","ValidTo":"2020031","Issuer":"DC=local, DC=mydomain, CN=MOSCA"}

Request State

This command lets MOSCA request the state from the given device.

Post command:
{"MOSCA":"RequestState","Payload":{"IP":"192.168.0.204","cn":"iRC355"}}

Answer format:
{"MOSCA":"RequestState","Payload":{"cn":"iRC355","Status":"OK"}}

Possible Status answers:
“OK”
“Error. Request just exists.”
“Error.”

The status and finally the state data can also be requested.

Post command:
{"MOSCA":"GetState","Payload":{"cn":"iRC355"}}

Answer format:
{"MOSCA":"GetState","Payload":{"cn":"iRC355","Status":"Ready.","Vendor":"Canon"," Model":"iR-ADV C355","Firmware":"48.20"}}

Possible Status answers are:
“No request found.”
“Queued.”
“Pending.”
“Ready.”
“Error.”
“Auth-Error.” (if SNMPv3 authentication failed)

Possible State answers are:
“” *(while pending or if an error occured)*
Vendor: The name of the device vendor
Model: The device model
Firmware: The firmware version of the device

List of supported devices

Here you will find of a list of all by MOSCA supported and tested devices. Other untested devices may also work with MOSCA but with no warranty.

Device Name	Canon Platform
LBP1127C	NCA3.3 SFP
LBP1238	NCA3.3 SFP
LBP1861/1871	XPT2-Lite
LBP21x	NCA3.2 SFP
LBP226	NCA3.3 SFP
LBP227	NCA3.3 SFP
LBP228	NCA3.3 SFP
LBP25x	NCA3.0 SFP
LBP311/312	XPT1 SFP
LBP351	XPT1 SFP
LBP6670/6680/6780	XPT1 SFP
LBP635C	NCA3.1 SFP
LBP710	XPT1 SFP
iR C1325	NCA2.0
iR1435	NCA2.0
iR1643	NCA3.3
iR1643 II	NCA3.4
MF1127C	NCA3.3
MF635C/735C	NCA3.1
MF72xC	NCA2.0
MF1238	NCA3.3
MF1333C	NCA4.0
MF41x	NCA3.0
MF42x	NCA3.2
MF745C/746C	NCA3.3
i-SENSYS X C1533P	XPT2-Lite
i-SENSYS X C1538P	XPT2-Lite
iR-ADV xxxx	iR-ADV
iR-ADV Cxxxx	iR-ADV
iR-ADV-DX xxxx	iR-ADV
iR-ADV-DX Cxxxx	iR-ADV
Lexmark MX622	-

Used Ports and Protocols

Port Number	Protocol	Network Service	Source	Destination	Intended Use
123	UDP	NTP	Printer	MOSCA	Auto onboarding via NTPS request (optional)
161	UDP	SNMP	MOSCA	Printer	Get device Information for onboarding (optional)
80	TCP	HTTP	MOSCA	Printer	- Get active certificates - if certificate is invalid, request CSR on device and download CSR to MOSCA - Deploy and activate signed certificate
8000	TCP	HTTP	MOSCA	Printer	- Get active certificates - if certificate is invalid, request CSR on device and download CSR to MOSCA - Deploy and activate signed certificate
443	TCP	HTTPS	MOSCA	Printer	- Get active certificates (Encrypted communication) - if certificate is invalid, request CSR on device and download CSR to MOSCA (Encrypted communication) - Deploy and activate signed certificate (Encrypted communication)
8443	TCP	HTTPS	MOSCA	Printer	- Get active certificates (Encrypted communication) - if certificate is invalid, request CSR on device and download CSR to MOSCA (Encrypted communication) - Deploy and activate signed certificate (Encrypted communication)
135	TCP	RPC	MOSCA	Certificate Authority	Sign CSR against CA via Microsoft Remote Procedure Call (RPC) and by using an assigned certificate template
443	TCP	HTTPS	Client computer	MOSCA	Accessing MOSCA-LCM via a web browser (encrypted communication). Port can be configured.
80	TCP	HTTP	Client computer	MOSCA	Accessing MOSCA-LCM via a web browser. Port can be configured.