Cisco Unified Communications Manager 12

April 2, 2019

Introduction

This document is intended to support you with the integration of XCAPI into an existing environment of the Cisco Unified Communications Manager. In the following sections we describe the essential configuration steps for SIP trunking to allow optimal interworking of both, the XCAPI and the Cisco Unified Communications Manager.

Though being based on the Cisco Unified Communications Manager release 12 and 12.5, this document is applicable with other versions given a few adjustments.

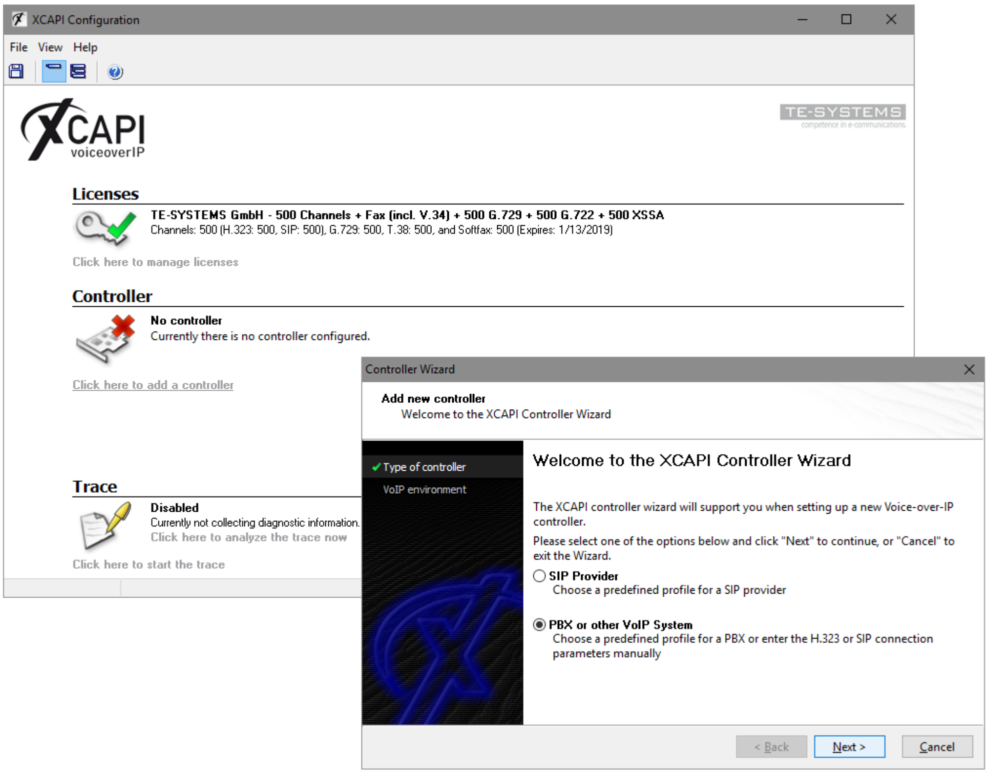
At this point we suppose that the Cisco Unified Communications Manager environment and the physical or virtual application server is available and accessible through the network. Application server in this context mean, a server with a recent available Microsoft Windows operating system with latest updates and patches included. Further, that the XCAPI and the CAPI 2.0 voice or fax application is properly installed. It is also supposed that the public network access via ISDN and/or SIP is given and properly working, also in context with the custom and country dependent numberings and call routings. The same goes for the networking (LAN, WAN, DMZ, NAT, Firewall) itself as such topics are beyond the scope of this document and thus not shown here at all. Please refer to the respective manufacturer documentations, manuals and examples in such cases.

However, independent of the deployed application, the SIP connection can be tested with the XCAPI's included test application (xtest.exe) that is available within the XCAPI's installation folder (by default `\\Program Files (x86)\TE-SYSTEMS\XCAPI\`). This test tool allows to check with inbound and outbound calls, fax and testing several supplementary services.

We recommend to visit our YouTube channel frequently for XCAPI related tutorials about licensing, the test tool, line monitor, tracing, analyzing and others. Registered community users can check about latest documents, TechNotes and releases for XCAPI.

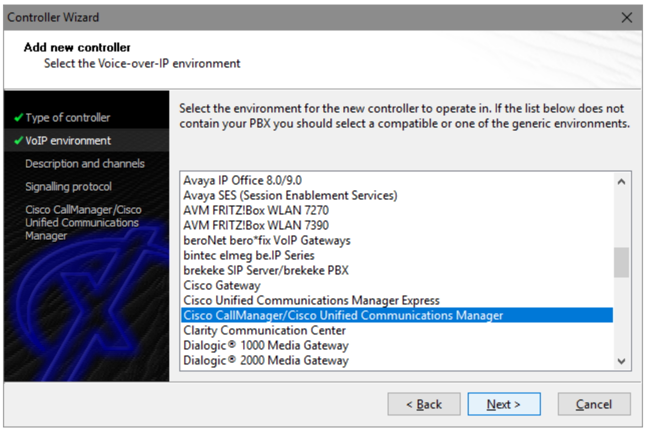
XCAPI Configuration

Please start up the XCAPI configuration to create a new controller assigned to the Cisco Unified Communications Manager.
If you've just installed the XCAPI and start the configuration tool for the first time or no controller is available at all, the XCAPI controller wizard will pop up automatically. To start up the XCAPI controller wizard manually, the hyperlink labeled **Click here to add a controller** on the main page has to be clicked.
Next select **PBX or other VoIP System** in the initial **Type of controller** dialog and proceed with the **Next** button.



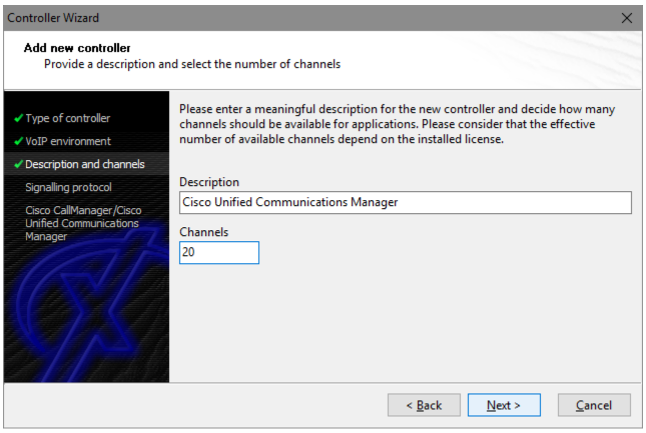
2.1 VoIP Environment

The next dialog lists some common Voice-over-IP environments. Selecting one of those will set up the XCAPI controller with a selection of near-optimal presets, sparing you manual configuration.



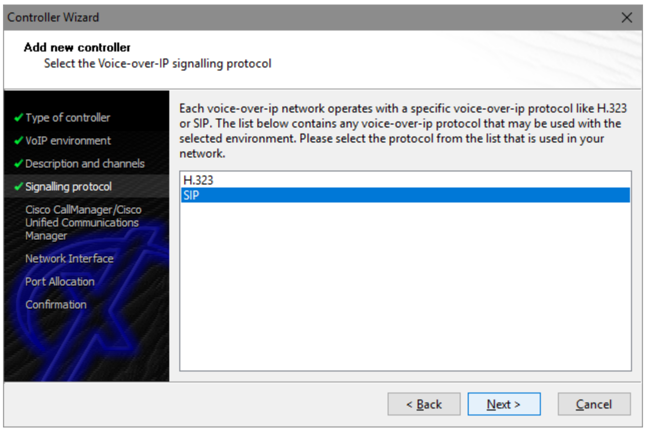
2.2 Description and Channels

When the VoIP environment was selected, the next dialog allows to set a description for the controller. Also the number of channels that the new controller will be able to provide can be set. Here you enter how many simultaneous connections the XCAPI controller should handle when communicating with the Cisco Unified Communications Manager and the bound CAPI 2.0 application.



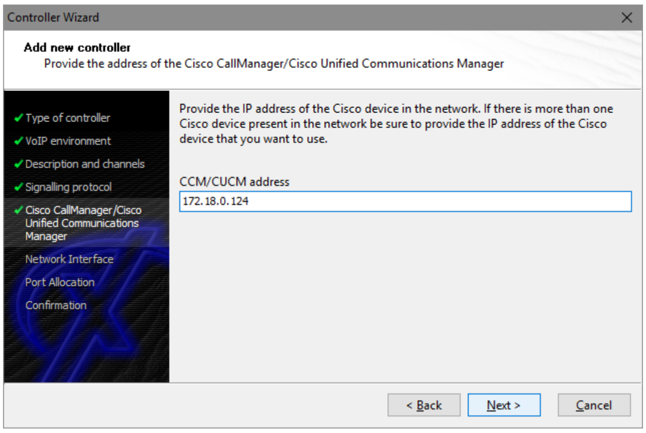
2.3 Signaling Protocol

The next dialog shows a list of signaling protocols which are supported for the given Voice-over-IP environment. According to this example the SIP protocol is selected.



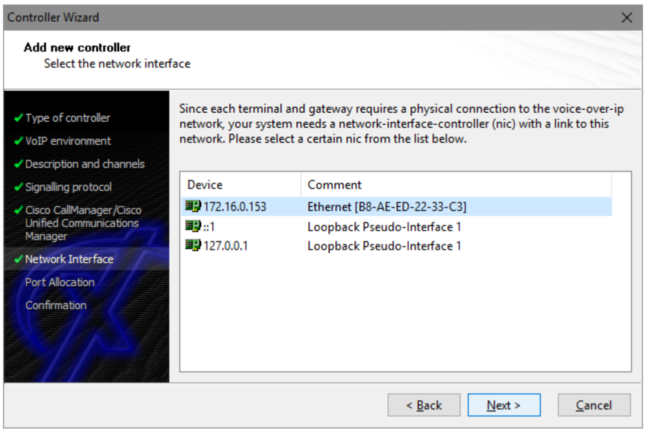
2.4 IP Address of the Cisco Unified Communications Manager

Next the IP address or host name of Cisco's environment must be provided. In this example the CUCM's Ethernet address is using 172.18.0.124.



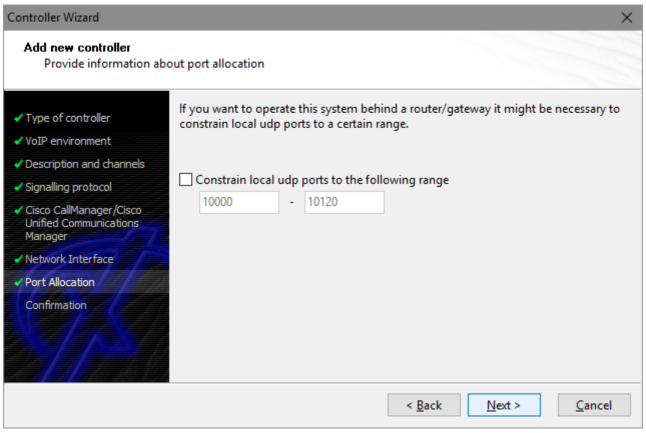
2.5 Network Interface

Afterwards, select the network interface that will control the inbound and outbound communications. Note that this is the XCAPI controller used Ethernet interface which will be leveraged for the SIP communication with the Cisco Unified Communications Manager.



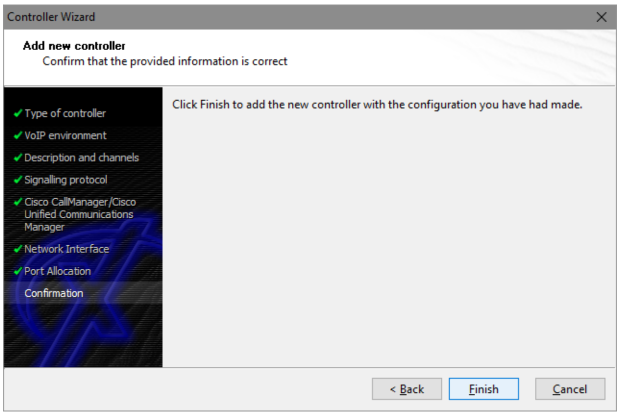
2.6 Port Allocation

On demand and in the case of any router or firewall restrictions for UDP (RTP/T.38) a port range can be specified. In this example no range will be set which allows the XCAPI controller to use a random port range between 1024 and 65535.

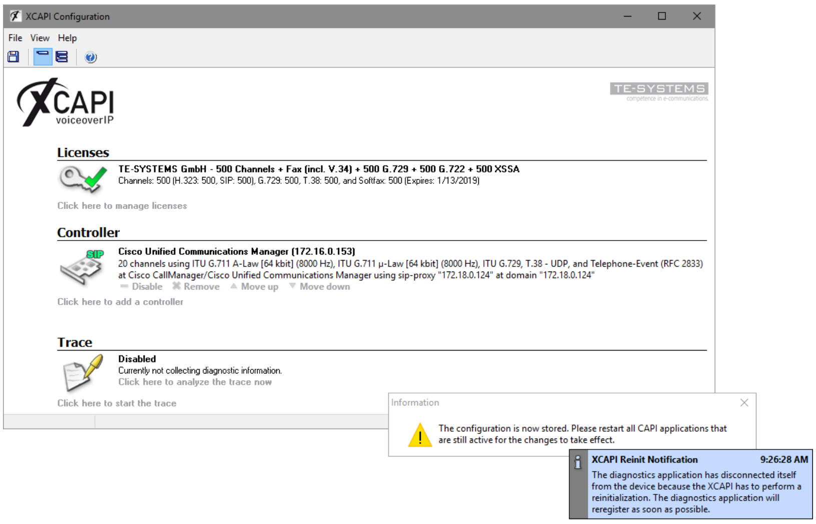


2.7 Confirmation

The final wizard dialog performs some checks on the configuration parameters you've made. If errors are detected, use the **Back** button to correct the respective erroneous dialogs. Use the **Finish** button in order to create the new controller.



Now, the created controller is listed on the main page of XCAPI's configuration tool. Use the **save** button and exit the tool.



Please note that the bound CAPI 2.0 application with its services must be completely stopped and restarted for the XCAPI controller changes to take effect. Restarting any of the XCAPI services won't help at all. Alternatively the Server where XCAPI is running on can be restarted. If enabled, the XCAPI diagnostic monitor pops-up with a re-initialization notification on success. Alternatively check with the **Events** tab of the **XCAPI Line Monitor** about a configuration update notification (Event ID 20).

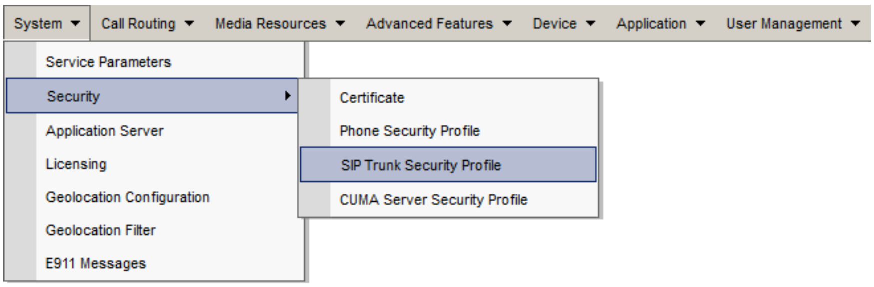
Configuring the Cisco Unified Communications Manager

In order to establish the communication between the Cisco Unified Communications Manager and the created XCAPI controller, a SIP trunk must be created. This enables XCAPI to be recognized as device handler for the Cisco environment. After creating the SIP trunk, a **Route Pattern** must be created for proper call-legs and call routings.

The SIP trunk must be related to some SIP and SIP Security Profiles. Some examples will be described in the following sections.

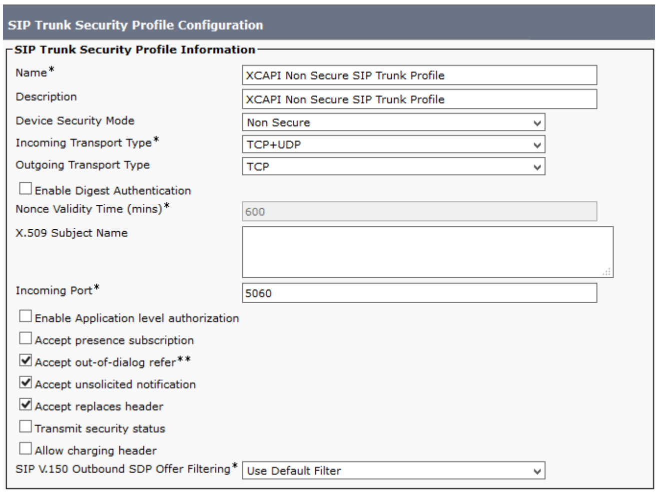
3.1 SIP Trunk Security Profile

First of all it is necessary to specify a **SIP Trunk Security Profile** which has to be applied to the XCAPI SIP trunk. The **SIP Trunk Security Profile** can be created or changed through the **Security** submenu of the **[System ▼]** tab. This profile can be used with or without **Digest Authentication**. Both methods will be described in detail here.
Besides the profile defaults, you may have to set the parameters **Accept Out-of-Dialog REFER**, **Accept Unsolicited Notification** and **Accept Replaces Header** for allowing supplementary services such as call transfer via SIP refer or message waiting indications via SIP Notify. Such services and XCAPI related configurations will be described in the chapter **Call Transfer** and **Message Waiting Indications** from page 37.



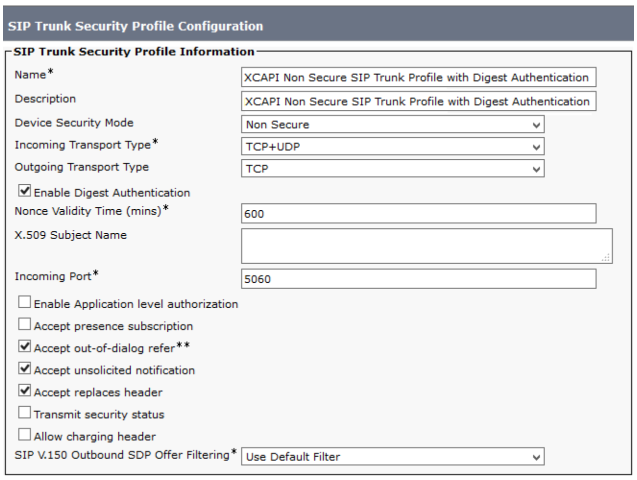
3.1.1 SIP Trunk Security Profile without Digest Authentication

For running a SIP trunk without any digest authentication the **Enable Digest Authentication** must be disabled.



3.1.2 SIP Trunk Security Profile with Digest Authentication

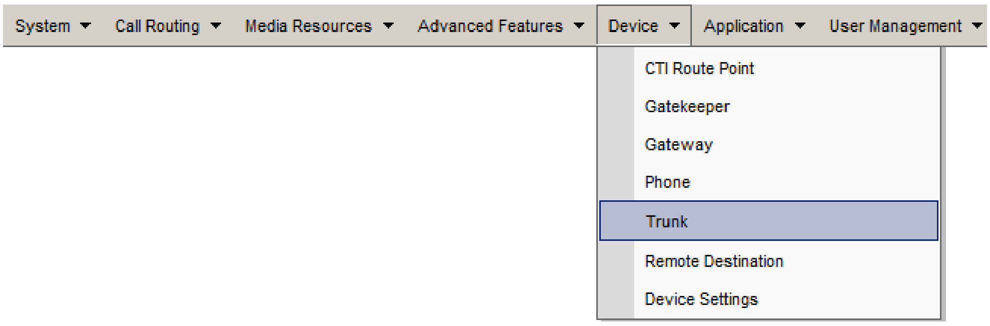
For this example the existing **Non Secure SIP Trunk Profile** will be copied, renamed to **XCAPI Non Secure SIP Trunk Profile with Digest Authentication** and adapted for using digest authentication. Of course a new SIP trunk could be created, it is just mandatory to set the **Enable Digest Authentication** parameter.



3.2 SIP Trunking

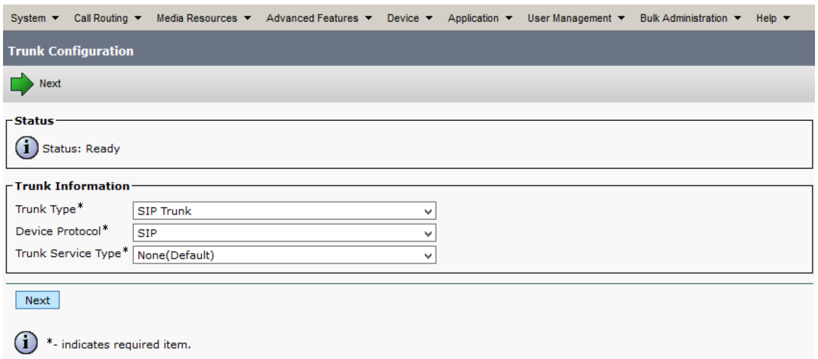
A new SIP trunk can be created by selecting the **Trunk** entry through the Cisco Unified Communications Manager **[Device ▼]** menu.

As described in the previous **SIP Trunk Security Profiles** chapters from page 8, the XCAPI related **SIP Trunk** can be used with or without digest authentication. The only difference has to be made by the selection of the corresponding **SIP Trunk Security Profile** which has the **Enable Digest Authentication** parameter set or not. If digest authentication is required additional configurations have to be made.

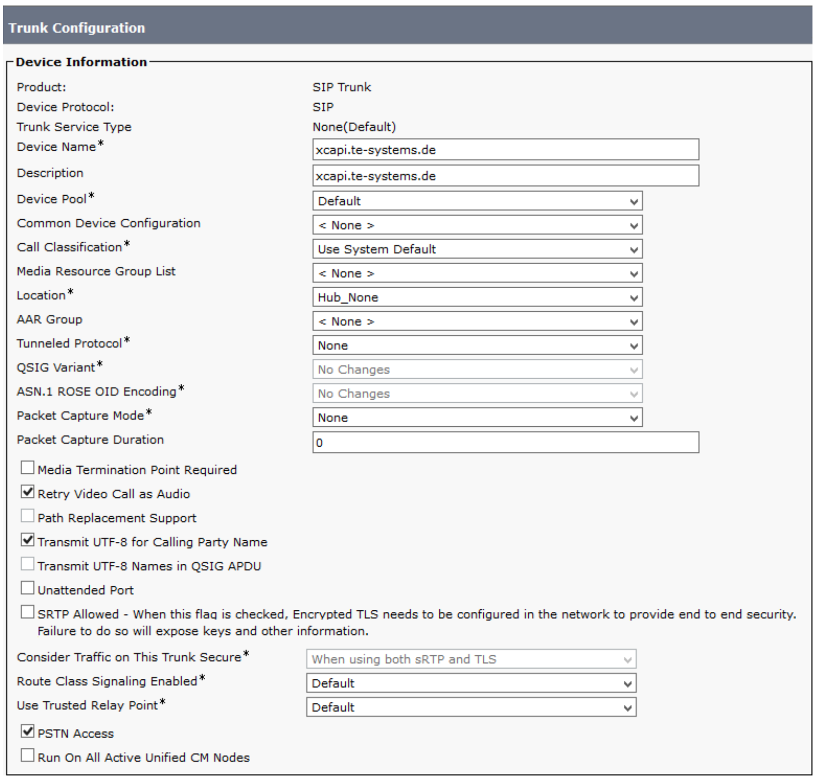


3.2.1 SIP Trunking without Digest Authentication

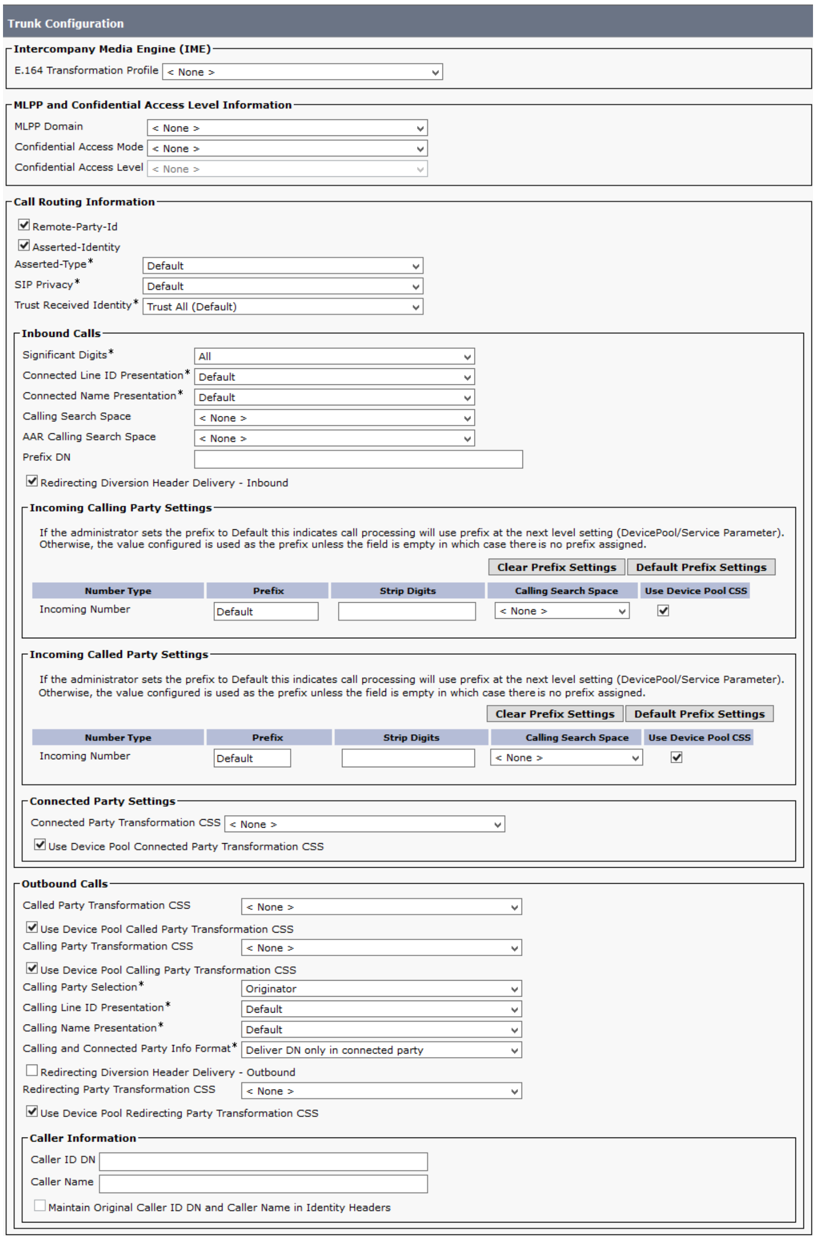
According to the selected protocol and the XCAPI SIP controller, the **Trunk Type** must be assigned to **SIP**. The **Device Protocol** parameter will be automatically set to **SIP** and the **Trunk Service Type** is used with **None (Default)**.



The shown **Trunk Configuration** is basically used with the system given defaults. The **Device Name** identifier as well as the **Description** is set to xcapi.te-systems.de, the host name of the XCAPI controller's assigned Ethernet interface IP address.



For the **Call Routing Information (Inbound and Outbound Call)** the parameter **Redirecting Diversion Header Delivery** has to be set. This enables the delivery of the origin and redirecting number through SIP. All other parameters are used with their defaults.



Trunk Configuration

Intercompany Media Engine (IME)

E.164 Transformation Profile: < None >

MLPP and Confidential Access Level Information

MLPP Domain: < None >
Confidential Access Mode: < None >
Confidential Access Level: < None >

Call Routing Information

☑ Remote-Party-Id
☑ Asserted-Identity
Asserted-Type*: Default
SIP Privacy*: Default
Trust Received Identity*: Trust All (Default)

Inbound Calls

Significant Digits*: All
Connected Line ID Presentation*: Default
Connected Name Presentation*: Default
Calling Search Space: < None >
AAR Calling Search Space: < None >
Prefix DN:
☑ Redirecting Diversion Header Delivery - Inbound

Incoming Calling Party Settings

If the administrator sets the prefix to Default this indicates call processing will use prefix at the next level setting (DevicePool/Service Parameter). Otherwise, the value configured is used as the prefix unless the field is empty in which case there is no prefix assigned.

Clear Prefix Settings | Default Prefix Settings

Number Type	Prefix	Strip Digits	Calling Search Space	Use Device Pool CSS
Incoming Number	Default		< None >	☑

Incoming Called Party Settings

If the administrator sets the prefix to Default this indicates call processing will use prefix at the next level setting (DevicePool/Service Parameter). Otherwise, the value configured is used as the prefix unless the field is empty in which case there is no prefix assigned.

Clear Prefix Settings | Default Prefix Settings

Number Type	Prefix	Strip Digits	Calling Search Space	Use Device Pool CSS
Incoming Number	Default		< None >	☑

Connected Party Settings

Connected Party Transformation CSS: < None >
☑ Use Device Pool Connected Party Transformation CSS

Outbound Calls

Called Party Transformation CSS: < None >
☑ Use Device Pool Called Party Transformation CSS
Calling Party Transformation CSS: < None >
☑ Use Device Pool Calling Party Transformation CSS
Calling Party Selection*: Originator
Calling Line ID Presentation*: Default
Calling Name Presentation*: Default
Calling and Connected Party Info Format*: Deliver DN only in connected party
☐ Redirecting Diversion Header Delivery - Outbound
Redirecting Party Transformation CSS: < None >
☑ Use Device Pool Redirecting Party Transformation CSS

Caller Information

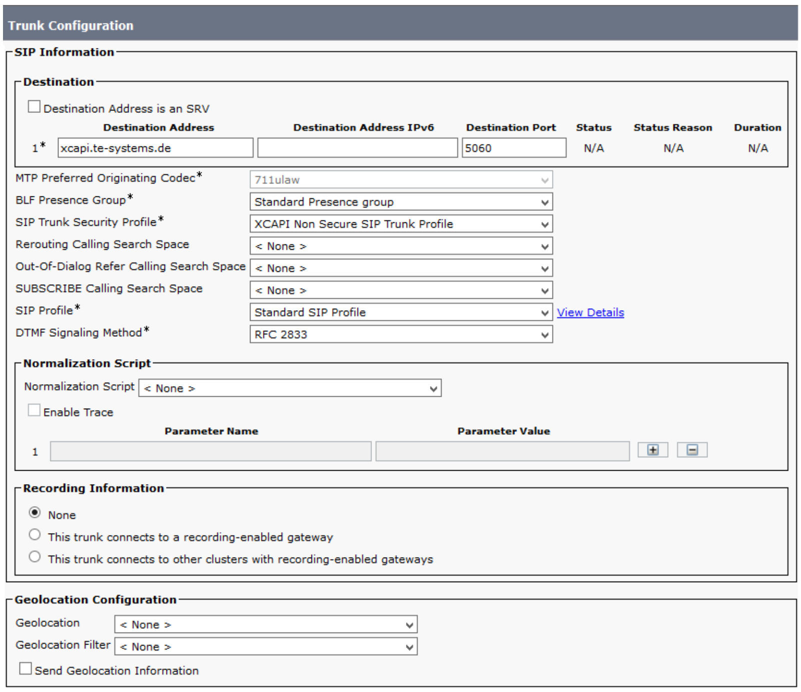
Caller ID DN:
Caller Name:
☐ Maintain Original Caller ID DN and Caller Name in Identity Headers

In this example the **Destination Address** is set to the host name xcapi.te-systems.de. The SIP Trunk Security Profile as mentioned in the chapter from page 8, is set to XCAPI Non Secure SIP Trunk Profile. The parameters Destination Address is an SRV, Destination Port, MTP Preferred Originating Codec and Presence Group of the SIP Information section are used with their defaults.

If required the Rerouting, Out-Of-Dialog Refer and SUBSCRIBE Calling Search Space parameters has to be set.

The DTMF Signaling Method is used with RFC 2833.

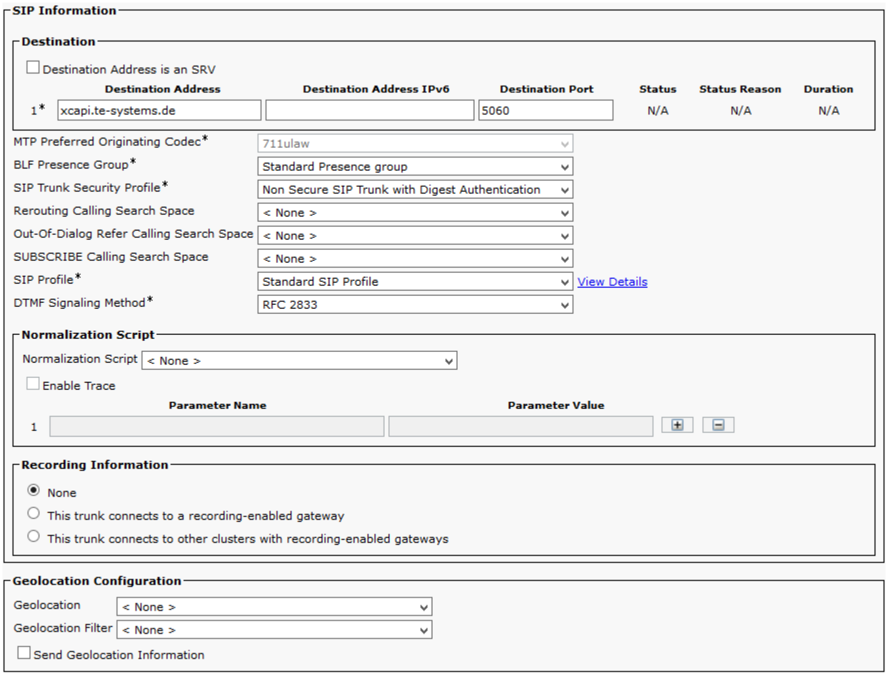
Note that it is not necessary to reset the newly created SIP trunk when the Route Pattern will be added afterwards.



Please note that the **Rerouting, Out-Of-Dialog Refer and SUBSCRIBE Calling Search Space** parameters must be assigned for appropriate SIP trunk rights. Wrong calling search space relations will cause call and/or call transfer failures. A good indicator for of incorrect routing would be **404 Not Found** notifications in reply of a SIP Invite or SIP Refer from the Cisco Unified Communications Manager.

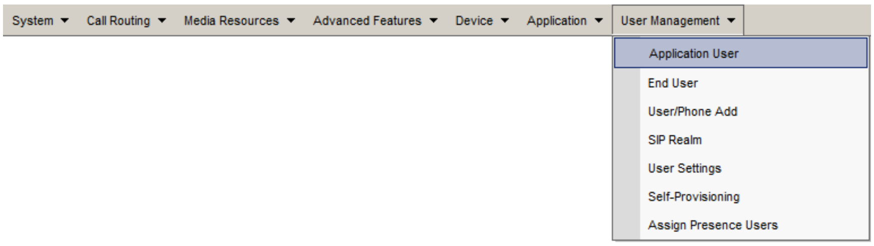
3.2.2 SIP Trunking with Digest Authentication

Creating SIP Trunks with Digest Authentication is similar to the ones without any authentication as previously described in the chapter **SIP Trunks** on page 10. Please note that the **SIP Trunk Security profile** for **Digest Authentication** must be handled in a separate profile. The SIP trunk profile, named **Non Secure SIP Trunk with Digest Authentication**, will be described in the chapter **SIP Trunk Security Profiles** starting on page 8.

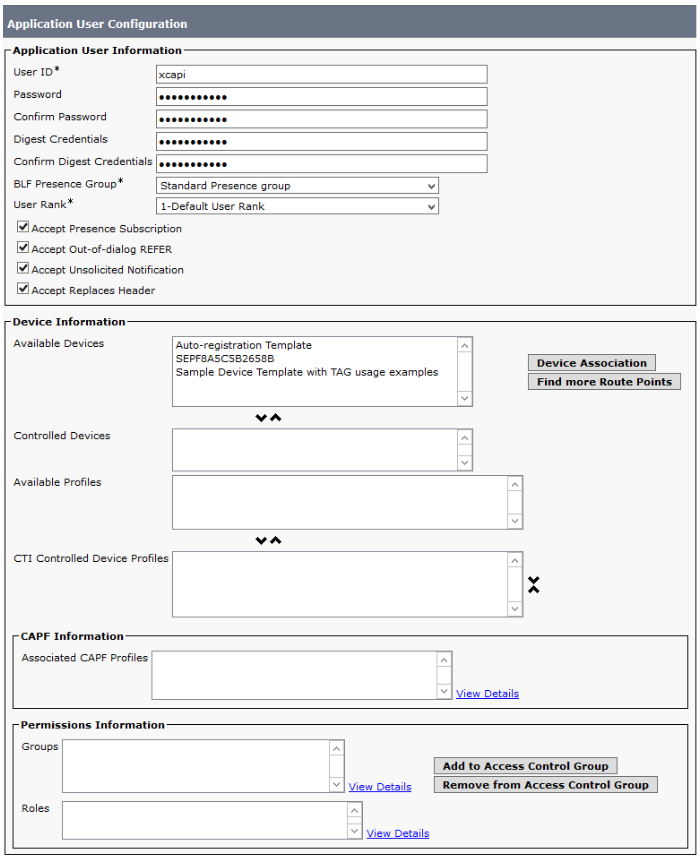


3.2.2.1 User Management

An **Application User** must be created for allowing **Digest Authentication**. For this please select the **Application User's** configuration in the Cisco's **[User Management ▼]** tab.



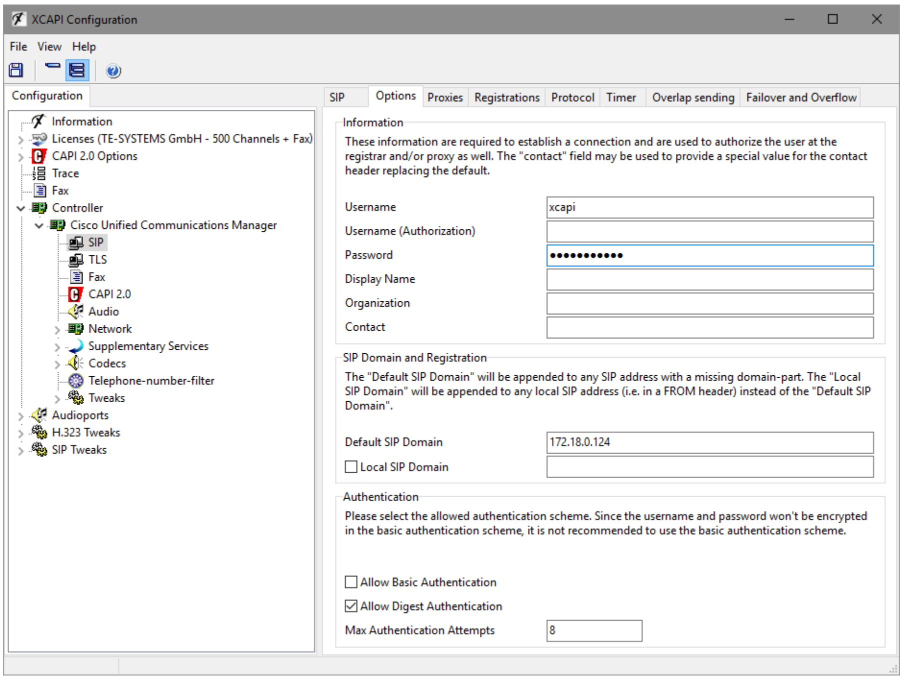
For the **Application User Information** configuration, the required authentication information has to be defined. In this example, the **User ID** is set to **xcapi** and used with an arbitrary password. The parameters **Digest Credentials** and **Confirm Digest Credentials** are used for the **Digest Authentication** method. This is your SIP password that has to be set to the XCAPI controller as shown in the chapter **XCAPI with Digest Authentication** on page 16. The parameters **Accept Presence Subscription**, **Accept Out-of-Dialog REFER**, **Accept Unsolicited Notification** and **Accept Replaces Header** are enabled. The **Device Information** parameters aren't modified at all.



3.2.2.2 XCAPI with Digest Authentication

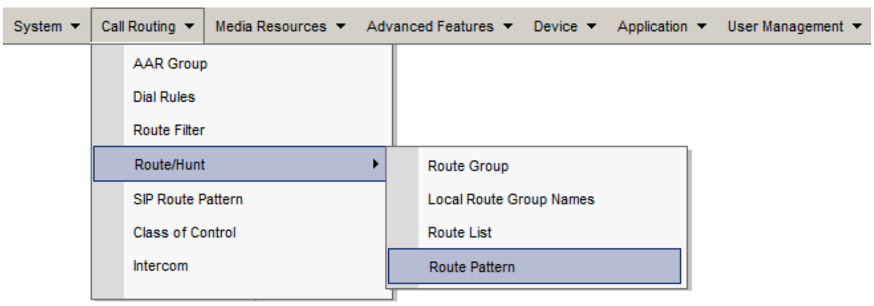
In accordance with Cisco's defined application user information, the given credentials must also be set to the XCAPI SIP controller. That ensures that the correct username and password will be used for proper authentication.

For enabling the authentication ensure that the **Allow Digest Authentication** is set.

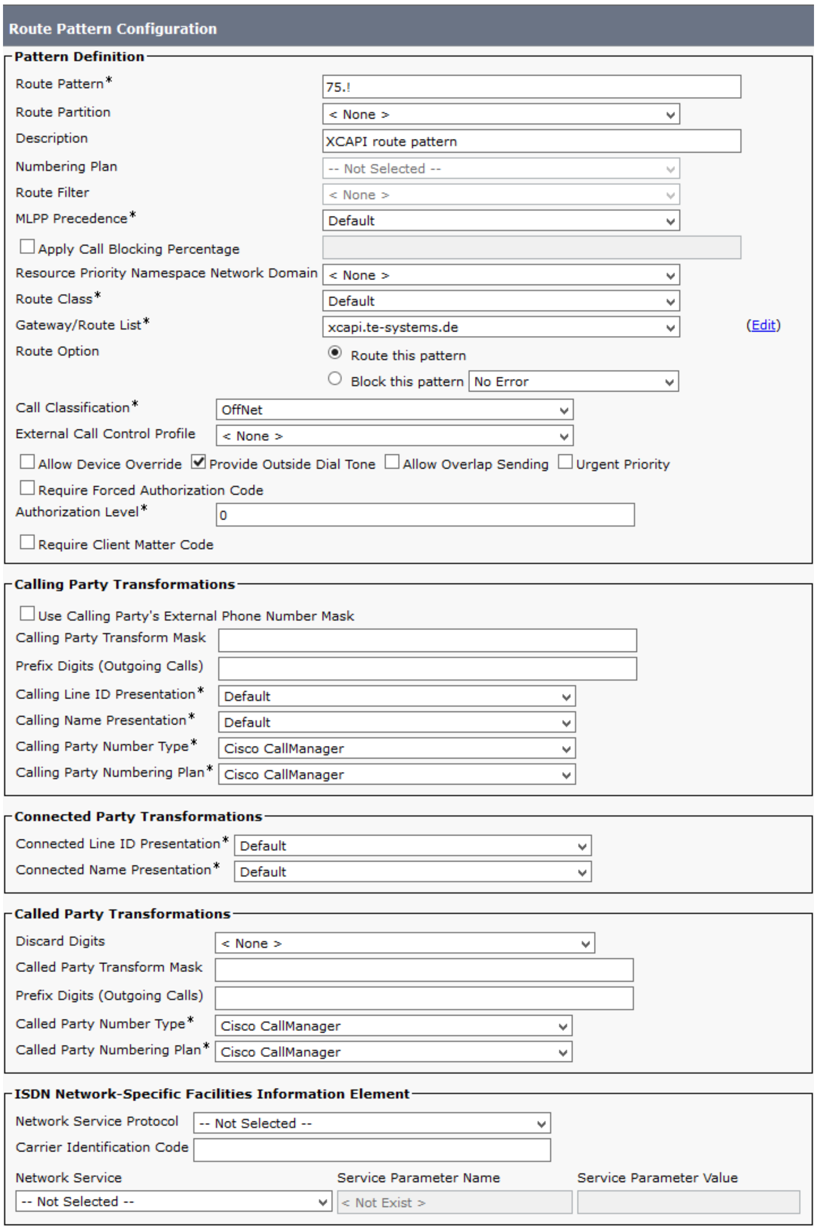


3.3 Route Pattern

Define XCAPI's SIP trunk required **Route Pattern** through the **[Call Routing ▼]** menu.



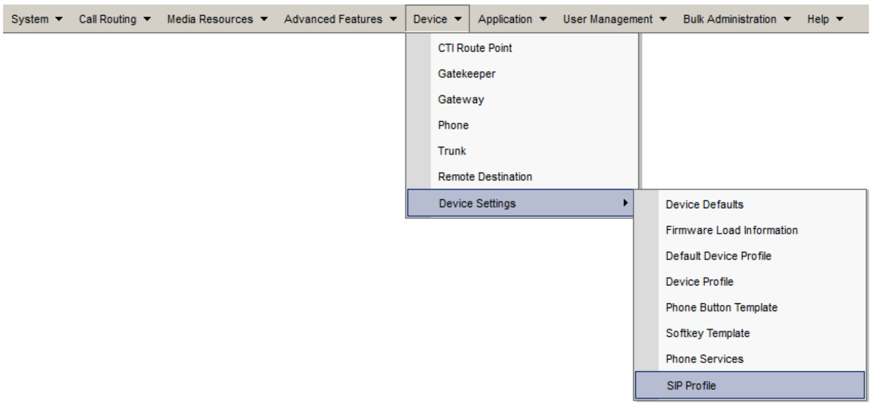
In this example the route pattern **75.!** is used for XCAPI's SIP trunk **xcapi.te-systems.de**.



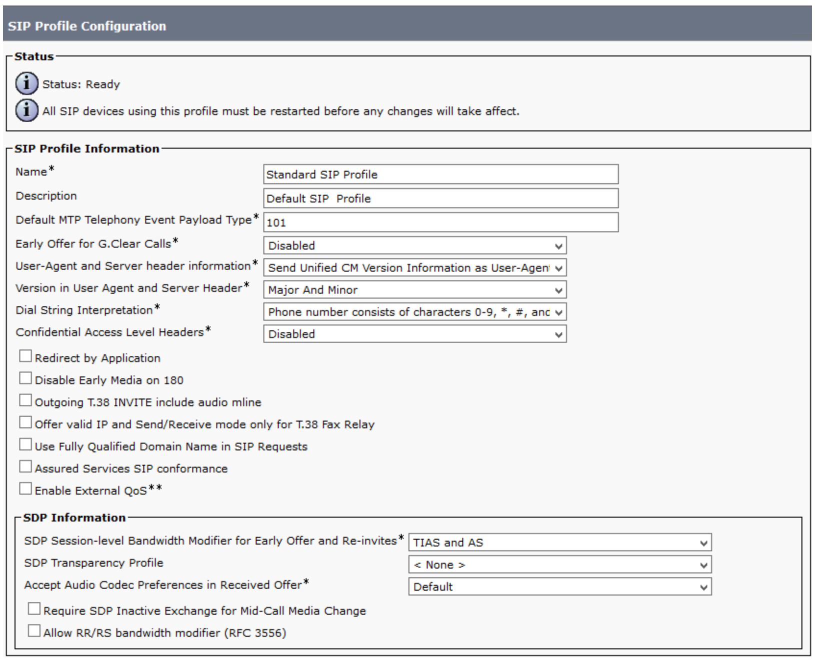
Please ensure that the appropriate **Route Partition** is assigned to the SIP trunk's Calling Search Space for proper basic call and call transfer behavior.

3.4 SIP Profile

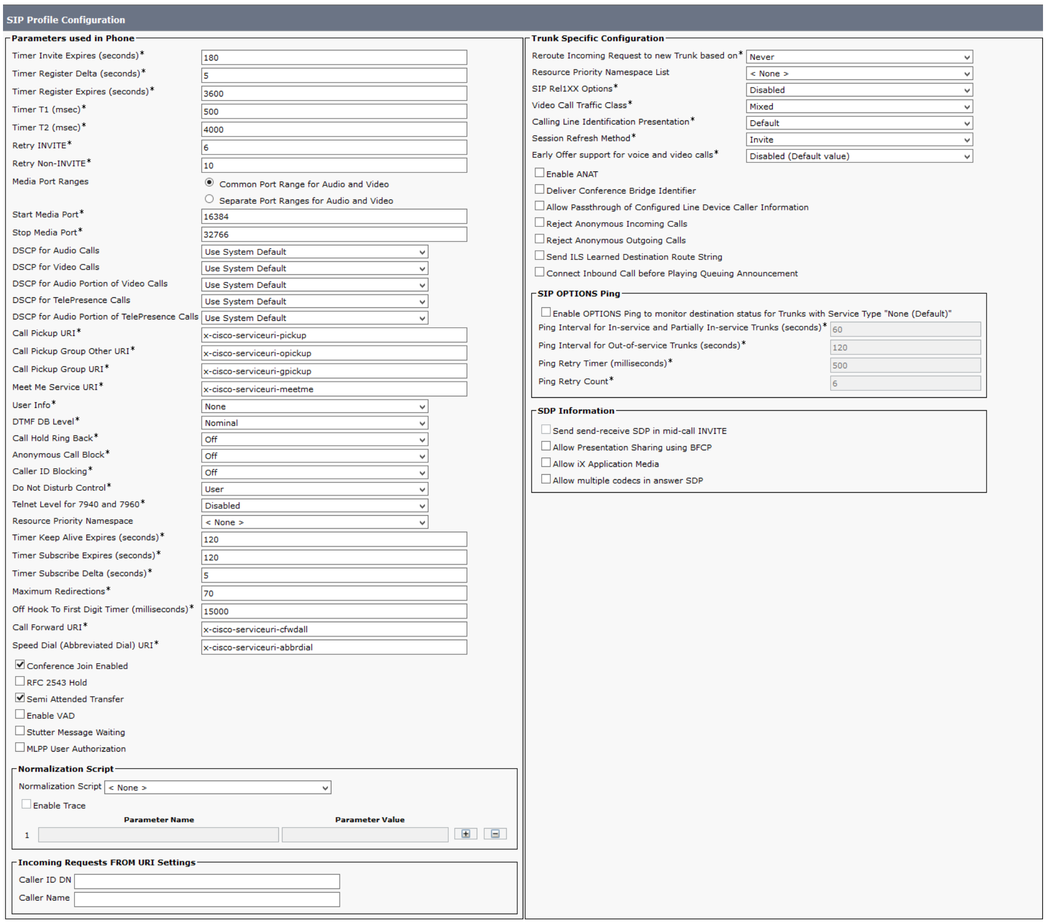
A SIP profile can be configured through Cisco's **[Device ▼] [Device Settings ▶]** menu. You can specify a set of SIP attributes (timings, ports etc.) to the appropriate SIP trunks and SIP endpoints.



In this example the **Standard SIP Profile** is used and assigned to XCAPI's SIP trunk as shown in the SIP trunking chapter on page 10.



The following SIP profile parameters are used with their defaults.



Transport Layer Security

The requirements and configuration procedure for **TLS (Transport Layer Security)** will be described in the following sections.

4.1 XCAPI SIP Security Additions

To enable **XCAPI SIP Security Additions (XSSA)**, it is necessary to run the **XSSA installer**, on the application/XCAPI server. The current version is **1.8.3**. Please note that a server reboot is required after the XSSA installation.
It is possible to use the **XCAPI SIP Security Additions (XSSA)** application (the xssa-ldr executable) for generating RSA keys, self-signed certificates and certificate signing requests. Please note that those **RSA** keys will be generated within the folder where the **xssa-ldr** executable is called.

4.1.1 RSA Keys & Self-Signed Certificates

The Cisco UCM can handle RSA keys with an encryption level up to **2048 bit**. For this example the XSSA-loader (xssa-ldr.exe) is used to generate a 2048 bit RSA key via the command line using the hostname of the XCAPI server. The private key is stored as **xcapi-private-key.pem** while the **xcapi-public-key.pem** filename is used for the public key. The corresponding command line for this is used as shown below:

```
C\>xssa-ldr crytool generate rsa --bits=2048
                                 --private=xcapi-private-key.pem
                                 --public=xcapi-public-key.pem
```

Next, this RSA key is used for generating a self-signed certificate. This **xcapi-certificate.pem** is valid for 365 days.

```
C:\>xssa-ldr crytool generate certificate --private=xcapi-private-key.pem
                                          --cn=xcapi.te-systems.de
                                          --idn=xcapi.te-systems.de
                                          --certificate=xcapi-certificate.pem
                                          --days=365
```

4.1.2 CA-Signed Certificate

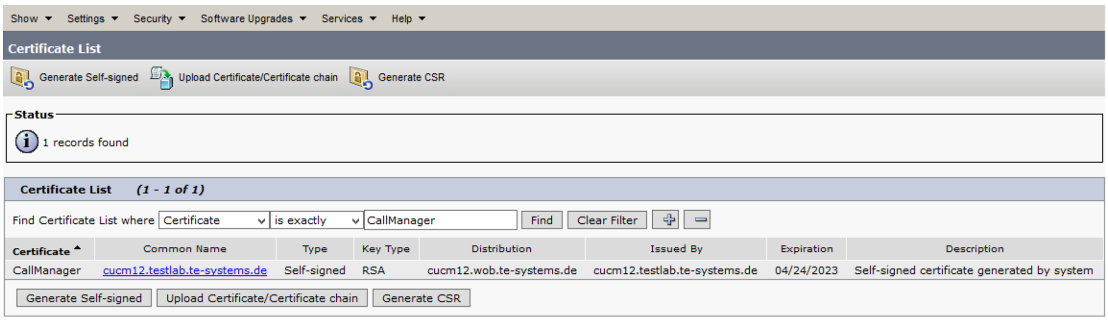
You can use the private key can to generate a **CSR (Certificate Signing Request)** file for requesting a CA-signed certificate. The next example shows how to create the **xcapi-csr.pem** file which is used for requesting a CA-signed certificate.

```
C:>xssa-ldr crytool generate csr --private=xcapi-private-key.pem
                                 --cn=xcapi.te-systems.de
                                 --idn=xcapi.te-systems.de
                                 --csr=xcapi-csr.pem
```

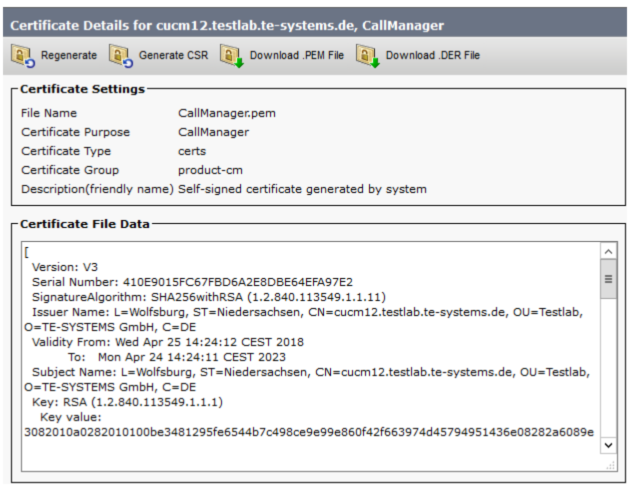
4.2 Certificate Management

The **Certificate Management** is handled through the **[Security ▼]** menu of the **Cisco Unified Operating System Administration**.

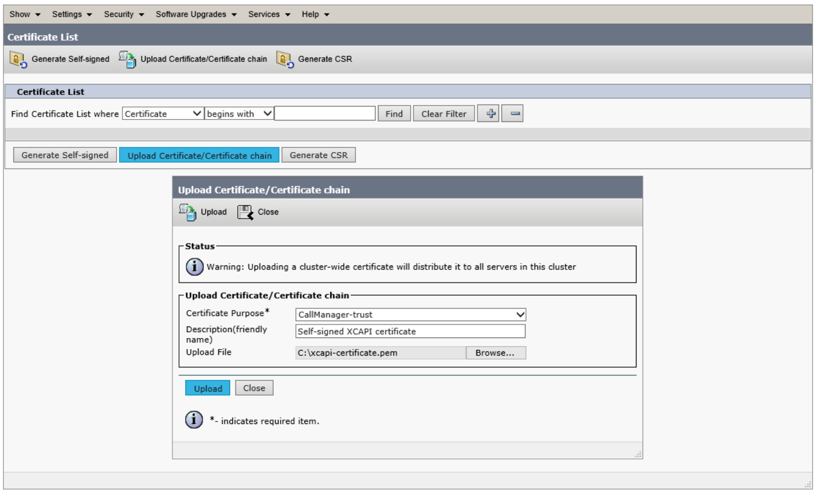
The CUCMs generated **CallManager.pem** certificate, which is used for this example, is shown in the certificates list.



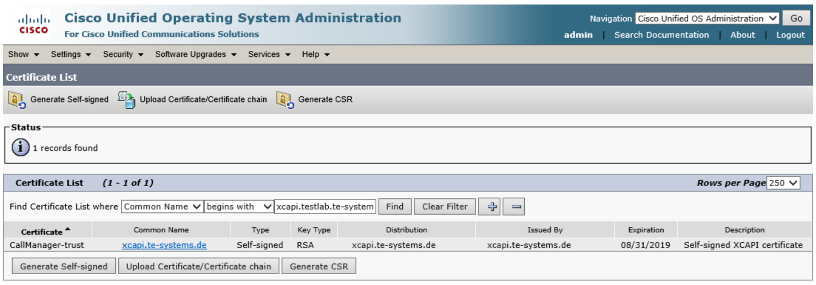
The **CallManager.pem** will be locally stored and has to be imported as **Trusted Certificate** to the XCAPI controller, which is described in detail in the chapter **Configuring the XCAPI SIP Security Additions** on page 28.



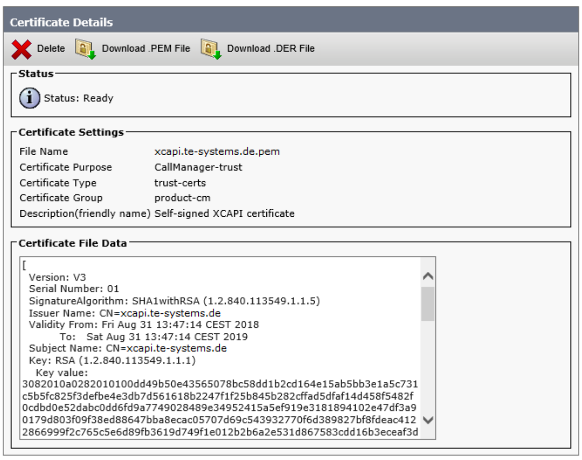
The generated XSSA certificate **xcapi-certificate.pem** has to be imported to the CallManager.



Afterwards, the XCAPI certificate will be shown in the certificate list.

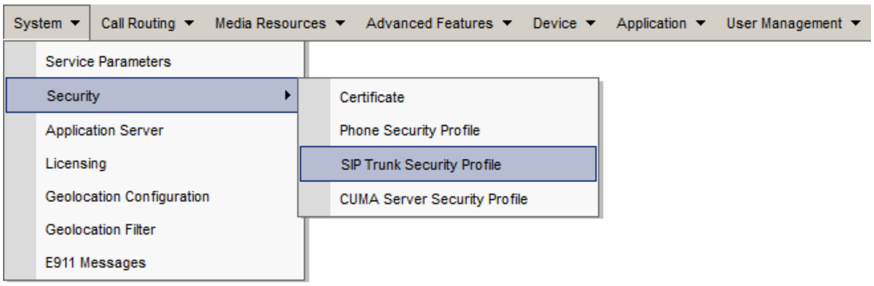


Please ensure that the **Subject** line, in this example **Subject: CN=xcapi.te-systems.de**, displays the correct host name. This must be correct for the **SIP Trunk Security Profile**, as shown in the next chapter on page 23.



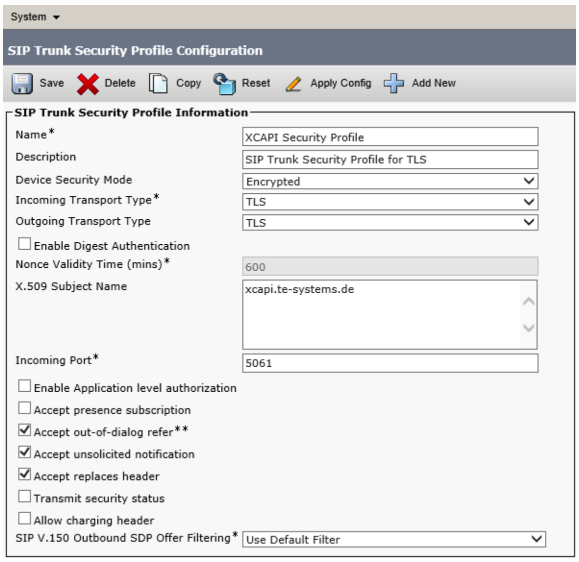
4.3 SIP Trunk Security Profile for TLS

Enabling TLS requires a properly configured **SIP Trunk Security Profile**.



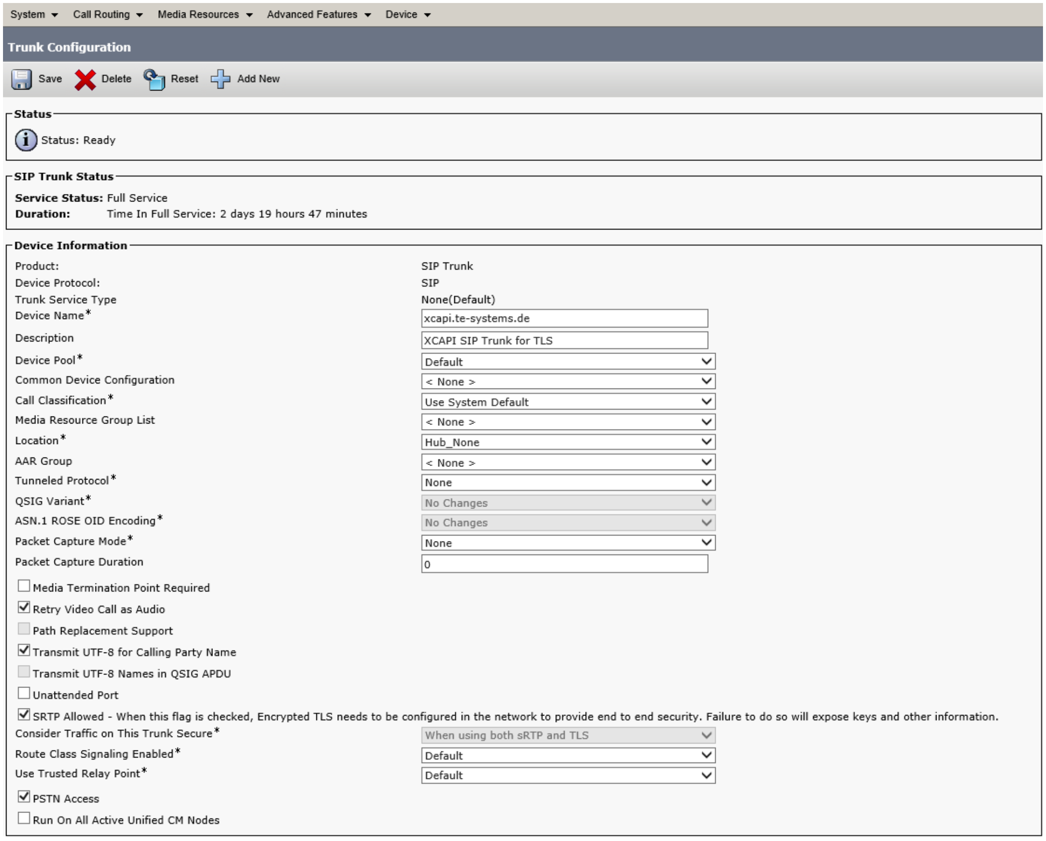
In this example the profile is used as follows:

- The **Device Security Mode** must be set to **Encrypted**.
- The **Incoming** and **Outgoing Transport Type** must be set to **TLS**.
- The **X.509 Subject Name** must be equivalent to the one of the XCAPI certificates, here **xcapi.te-systems.de**.
- The **Incoming Port** is set to **5061** which is also used as default TLS port by the XCAPI controller.
- The **Accept out-of-dialog refer**, **Accept unsolicited notification** and **accept replaces header** are used enabled.

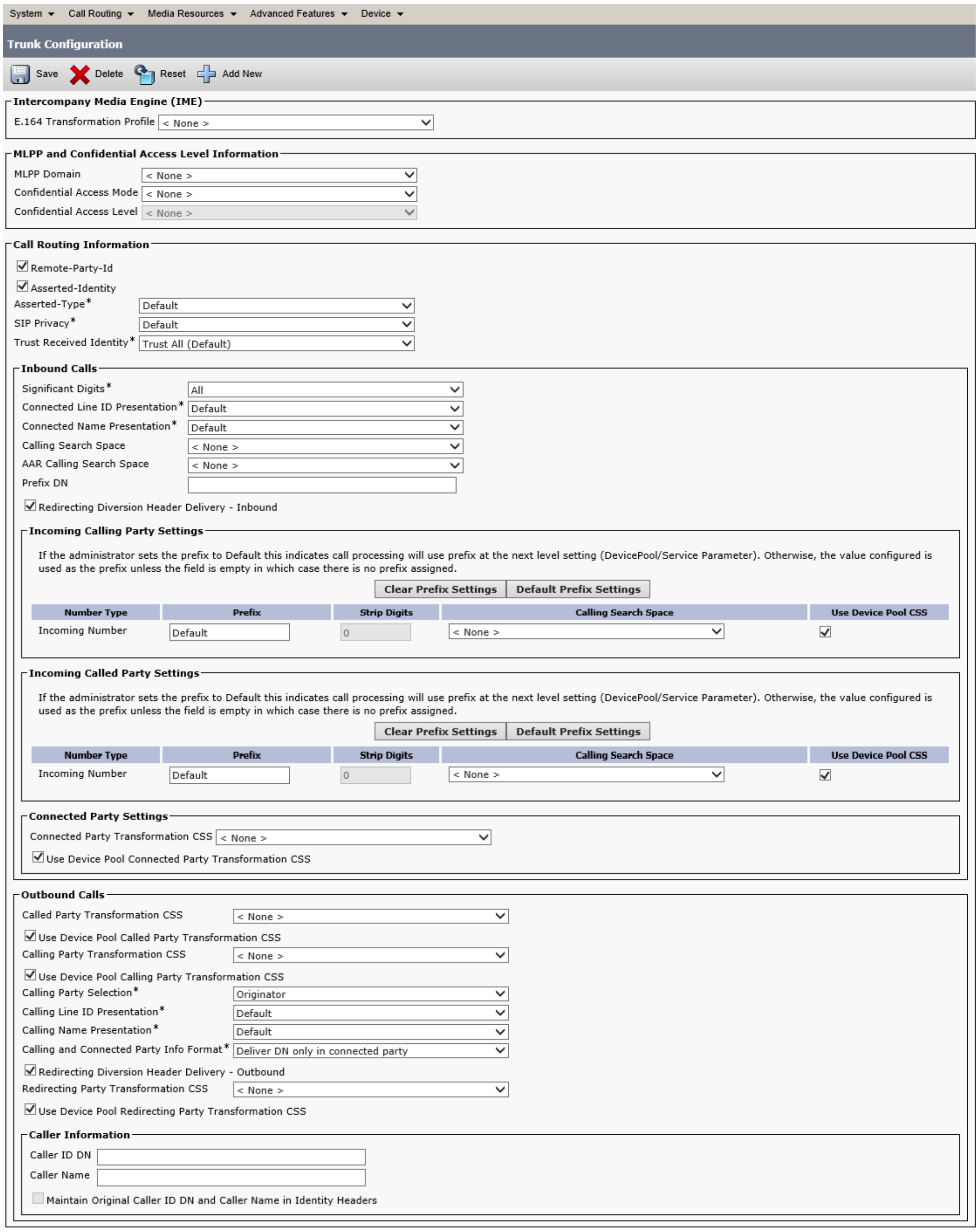


4.4 SIP Trunking with TLS

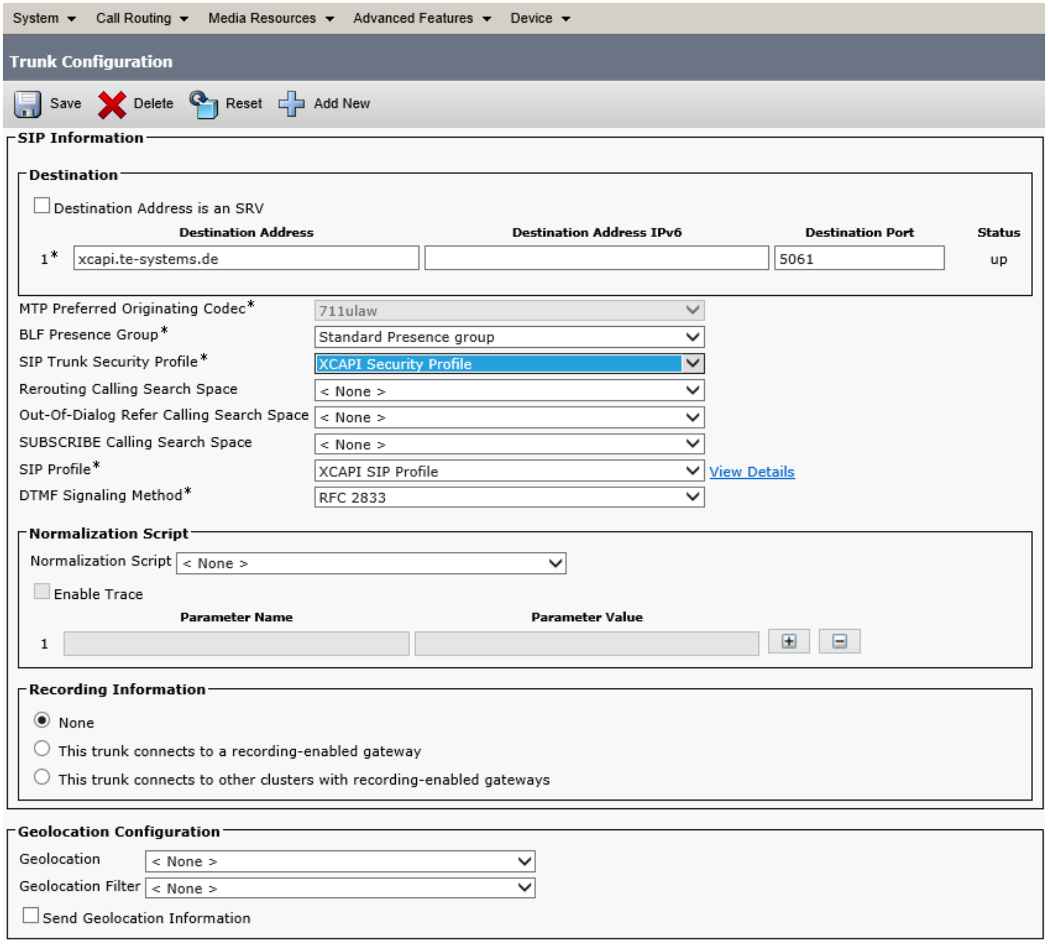
The SIP trunk for TLS has to be created as a standard SIP trunk (see chapter **SIP Trunking** on page 10). Additionally the TLS secured SIP trunk must be used with an enabled **SRTP Allowed** parameter. In detail this trunk will be used as follows:



Please ensure that the parameters for standard SIP trunking, **Redirecting Diversion Header Delivery - Inbound** and **Redirecting Diversion Header Delivery - Outbound** are enabled for redirection numbering support.

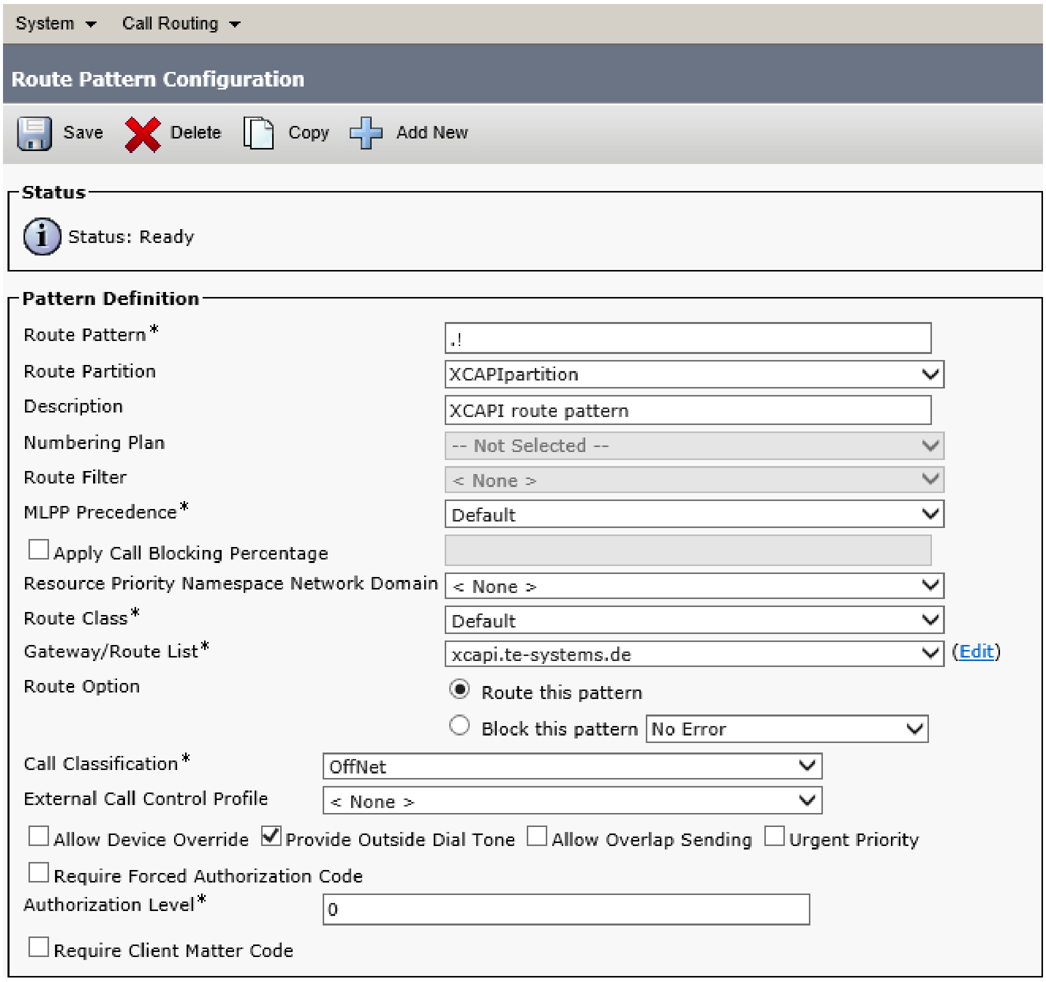


Beside of the default values within the **SIP Information** dialog, the **Destination Address** is used with the host address **xcapi.te-systems.de** and the default port for TLS **5061**. The **SIP Trunk Security Profile** is associated to the **XCAPI-Server-TLS** security profile.



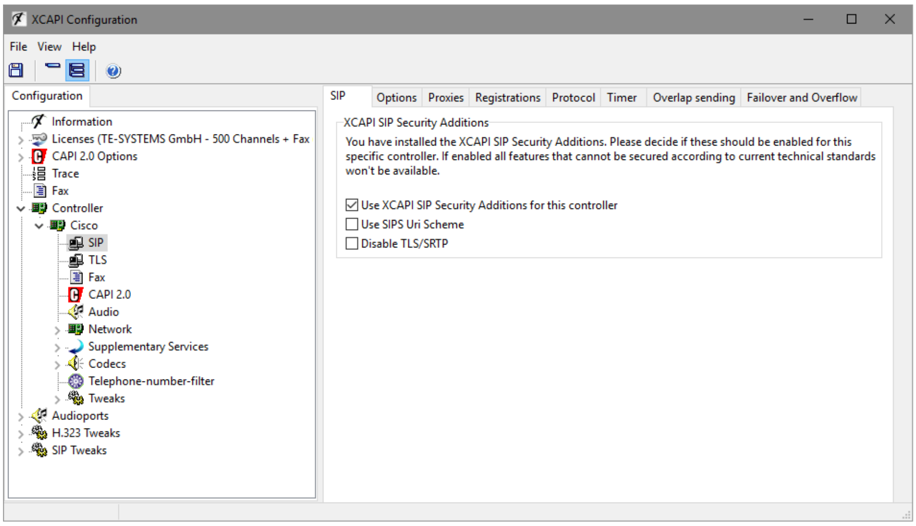
4.5 Route Pattern

The **Route Pattern** for the TLS SIP trunk is used as shown:

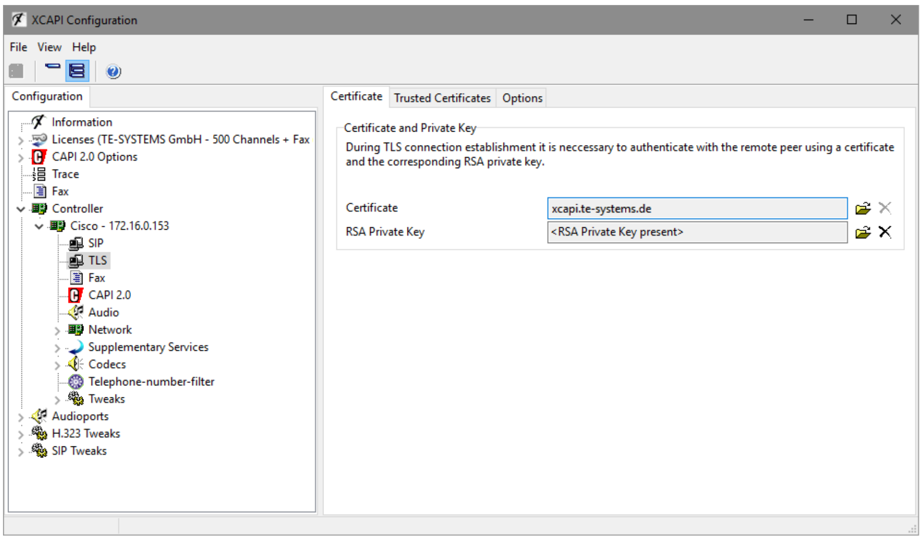


4.6 Configuring the XCAPI SIP Security Additions

For running XSSA it is necessary to enable the **Use XCAPI SIP Security Additions for this controller** option.

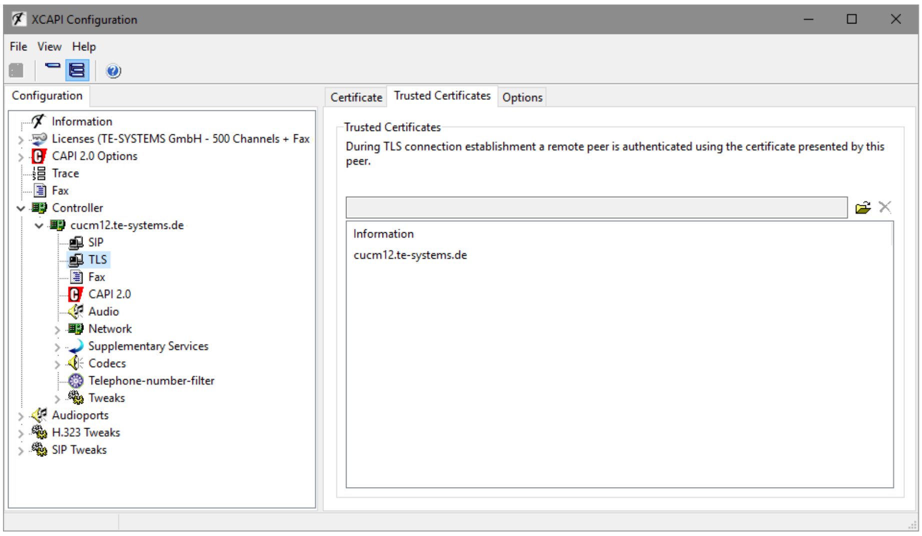


The self-generated **xcapi-certificate.pem** file, as described in the chapter **Certificate Management** on page 21 and the associated RSA key **xcapi-private-key.pem** must be uploaded through the XCAPI controllers **TLS Certificate** dialog.



Within the **Trusted Certificates** dialog you have to import the **CallManager.pem** certificate, as shown in the chapter **Certificate Management** on page 21.

Finally you have to save the XCAPI controller changes and need to restart the CAPI application services.



Fax Services

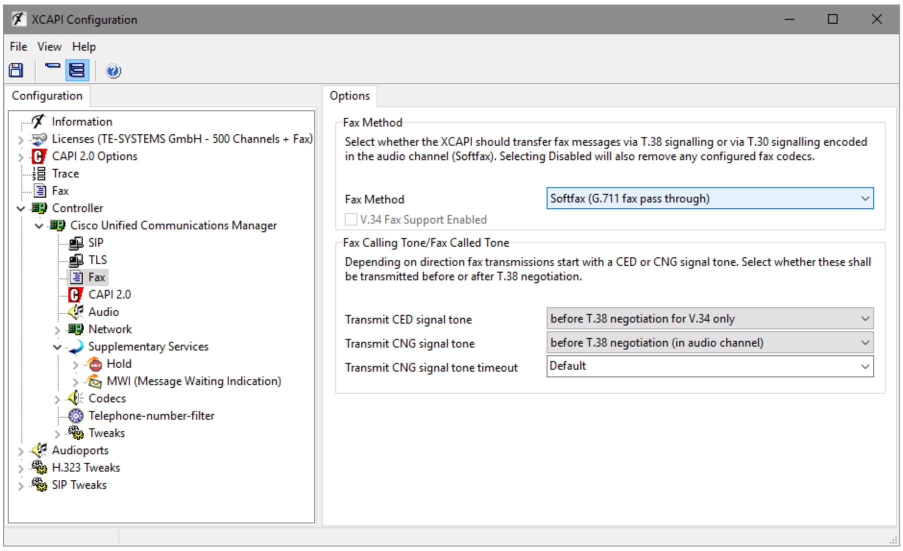
In this chapter, we are going to describe configuring the fax services leveraging T.38 (including V.34), Softfax (G.711) and T.38 to Softfax fallback.
For faxing to function correctly it must be ensured that the Codec, Framing, Bandwidth and DTMF settings are set conform to the ones of the XCAPI controller configuration and other participating SIP instances.

Note that the XCAPI controller Fax dialog as well as T.38 (including V.34 support) to G.711 fallback support is available from XCAPI version 3.5.0. We strongly recommend using latest XCAPI versions for best results and it might be even be mandatory with latest manufacturer releases and firmware versions.

The fax related configurations for the Cisco gateway will be described in the chapter **Troubleshooting, Hints and Configuration Examples** from page 33. Please note that XCAPI does not support the **T.38** fax protocol through XSSA and enabled TLS.

5.1 SoftFax (G.711 Fax Pass Through)

In the **SoftFax** mode, the XCAPI simulates an analog fax device by transmitting modulated fax signals like a modem through the established G.711 audio channels. The **SoftFax (G.711 fax pass through)** fax method has to be enabled as shown below.

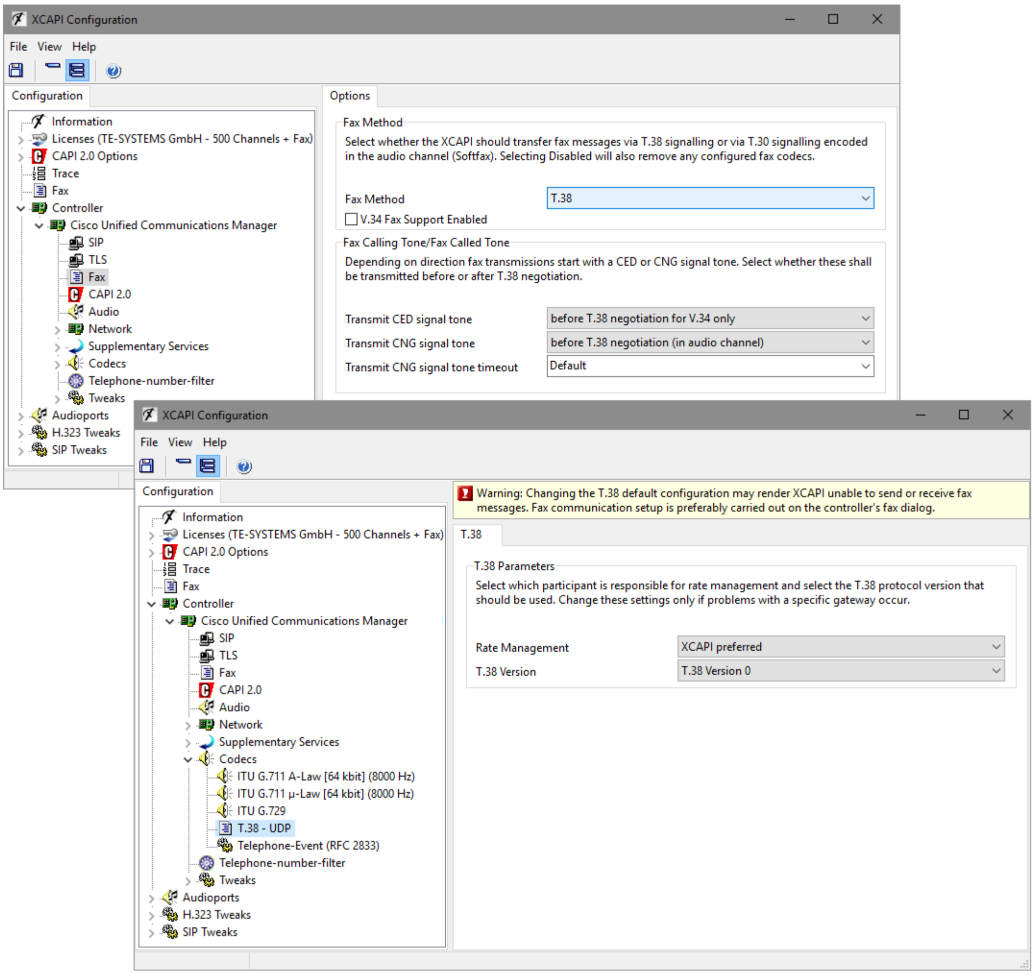


5.2 T.38

In the case of T.38, using this fax method must also be supported and enabled for all other participating instances in between (SIP gateways, SIP provider, SBCs etc.). It is strongly recommended to avoid any kind of unnecessary transcoding (for e.g. G.711 to T.38 or vice versa) and using standard fax methods for all participating instances.

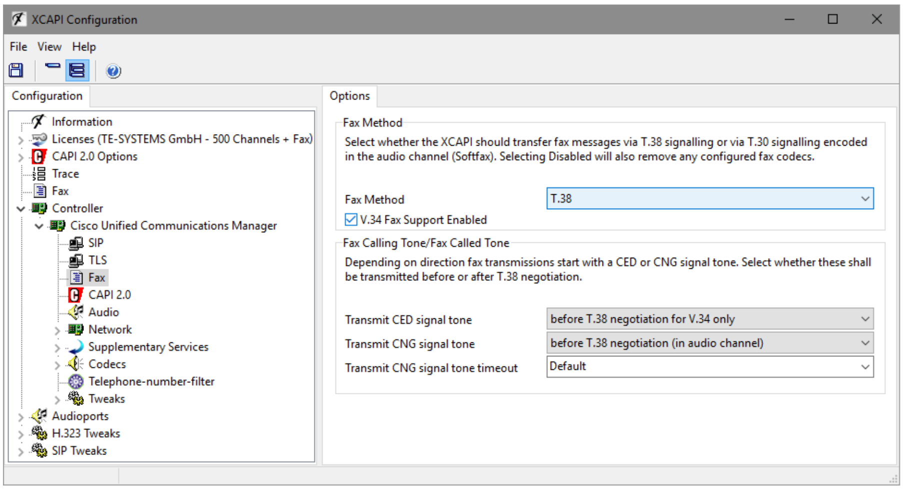
For enabling T.38 this **Fax Method** must be set as shown on the next screenshot.

Ensure that the **T.38 - UDP** is available and enabled within the **Codecs** tab of the XCAPI controller configuration. One speech codec (in common G.711law or G.711μ-law) must be enabled for the initial call establishment.



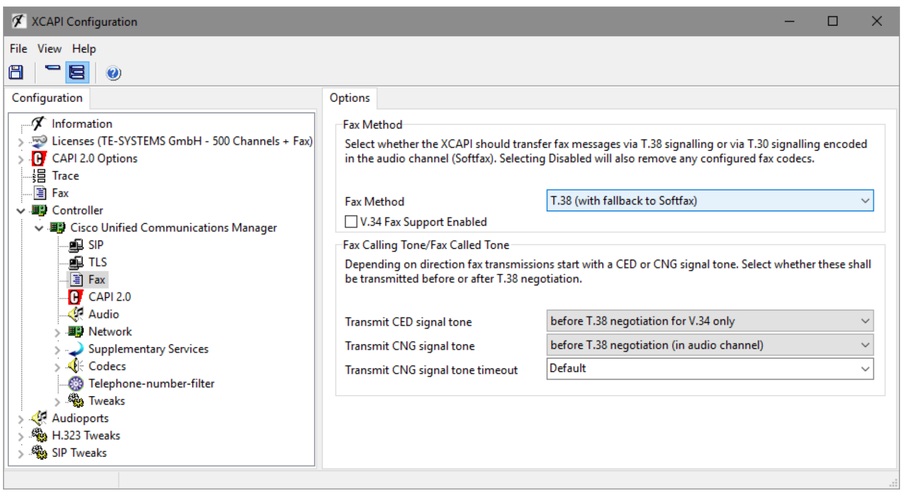
5.3 T.38 with V.34 Support

T.38 with V.34 is available from XCAPI version 3.5.0 and Cisco VoIP gateways from version 15.1. To enable T.38 with V.34, **T.38** as well as **V.34 Fax Support** must be enabled within the XCAPI controllers **Fax** tab. The appropriate Cisco configurations will be described within the **Troubleshooting** section starting on page 33.



5.4 T.38 with G.711 Fax Fallback

The fax fallback can be enabled, also with **V.34 Fax Support**, as shown on the screenshot below. The corresponding Cisco configurations will be described within the **Troubleshooting** section starting on page 33. It is strongly recommended to check if this mode is supported by all participating VoIP instances, especially in the case of session border controller's or connected SIP providers. Depending on the VoIP environment additional configurations might be required. Incorrect configurations (not only for the ones of the XCAPI controller) will result in bad or non-working fax transmissions.



Troubleshooting, Hints and Configuration Examples

For best practice and functionality please read through the hints and examples of this section. The XCAPI related configurations for the given fax dial-peer examples can be reviewed in the chapter **Fax Services** from page 30.

6.1 Common Hints

- There are several protocols like **H.323**, **SIP** or **MGCP** that can be used for building up the connectivity between the Cisco Unified Communications Manager and a Cisco gateway. If the Cisco gateway and Cisco Unified Communications Manager connectivity is interacting via the SIP or H.323 VoIP protocol, the same protocol has to be used for the XCAPI trunk. Using different protocols for the VoIP environment commonly causes more issues (like DTMF functionality) and other side effects which require in-depth analysis.
- The dial-peer command **destination-pattern** is used for setting up the routing for the Cisco Unified Communications Manager and its connected gateway and can be used as well for the XCAPI trunk.
- You should give consideration to configuring dial-peers for routing the calls from the Cisco Unified Communications Manager to its gateway, as you cannot setup all necessary parameters within the global **voice service voip** dialog.
- The **called-number** dial-peer command can be used for utilizing its parameters for outgoing (outbound) call legs.
- In practice a wide range of matching calling numbers has to be routed which can be invoked with the **incoming called-number T** command.
- Use the dial-peer command **answer-address** for matching a specific **calling number**.

6.2 Frequent Issues

- In a case of working incoming (inbound) faxes with the outgoing (outbound) transmission always failing, it is recommended you check with the dial-peer that is used for the outbound route. In most cases it is incorrectly configured.
- If the XCAPI controller is configured to use the Softfax (G.711 Fax Pass Through) method but no outbound (outgoing) dial-peer is assigned a corresponding G.711 codec, the gateway will use the globally defined **voice service voip** code settings, which will probably be T.38. You can correct this by using commands like **incoming called-number T** or **answer-address 123456** for proper dial-peer matchings.
- If connections are rejected immediately or terminated after the call establishment, the root cause is mostly due to wrong or not conformed codec configurations. The related Cisco Unified Communications Manager dial-peer should be configured with a G.711 μ-Law codec which has to be enabled in the XCAPI controller also. However, this is normally the default setting for both instances.

6.3 Network Clock

Wrong or faulty network clock configurations can be the reason for aborted faxes due to clocking and frame errors on the PRI. So if utilized, please check the proper PRI configurations and clocking or TX\RX errors. Example for the network:

```
network-clock-select 1 E1 0/0/0
```

6.4 MGCP

If using the **SoftFax (G.711 fax pass through)** method through an MGCP configured gateway, the dial-peer commands should be handled as follows. Do not set any of these MGCP commands:

```
mgcp modem passthrough voip mode nse
mgcp modem passthrough voip codec g711alaw (or codec g711ulaw)
mgcp fax t38 inhibit
mgcp fax t38 gateway force
```

Ensure that this MGCP command is set:

```
mgcp fax rate disabled
```

6.5 Using SoftFax (G.711 Fax Pass Through)

When running the SoftFax (G.711 fax pass through) method, you should avoid to enable commands like **fax protocol pass-through** or **fax protocol t.38**. Use the **fax rate disabled** command for disabling any gateway-sided fax detection for the related dial-peer.

SIP dial-peer example for using SoftFax (G.711 fax pass through):

```
dial-peer voice 800 voip
        destination-pattern 8...
        codec g711ulaw
        session protocol sipv2
        session target ipv4:192.168.1.100
        incoming called-number T

        dtmf-relay rtp-nte
        fax rate disable
```

6.6 Using SoftFax (G.711 Fax Pass Through) in Virtual Environments

The parameters **playout-delay nominal 250** and **playout-delay mode fixed** are used to specify a more graceful jitter buffer. So the handling of UDP/RTP packets might be handled in a more efficient way.

SIP dial-peer example, which is only used for outgoing facsimile transmissions when matching a specific prefix, for using SoftFax (G.711 fax pass through) in virtual environments:

```
translation-rule 2
        Rule 1 8999990  0

dial-peer voice 8999990 voip
        translate-outgoing called 2
        incoming called-number 8999990
        playout-delay nominal 250
        playout-delay mode fixed
        codec g711ulaw
        fax rate disable
        no vad
```

6.7 Using T.38

Using the T.38 fax protocol requires to set the **fax protocol t.38** command. It is recommended you enable **ECM** error correction mode. For this, you need to ensure that the **fax-relay ecm disable** command is NOT used.

SIP dial-peer example for using T.38:

```
dial-peer voice 800 voip
        destination-pattern 8...
        codec g711ulaw
        session protocol sipv2
        session target ipv4:192.168.1.100
        incoming called-number T
        dtmf-relay rtp-nte

        fax protocol t38 ls-redundancy 0 hs-redundancy
```

6.8 Using T.38 with V.34

Using the T.38 fax protocol requires you set up the **fax protocol t38 version 3** command. Make certain that the **fax-relay ecm disable** command has **NOT** been set because V.34 requires the error correction mode.

SIP dial-peer example for using T.38 with V.34:

```
dial-peer voice 800 voip
        destination-pattern 8...
        codec g711ulaw
        session protocol sipv2
        session target ipv4:192.168.1.100
        incoming called-number T
        dtmf-relay rtp-nte

        fax protocol t38 version 3 ls-redundancy 0 hs-redundancy 0 fallback none
```

6.9 Using T.38 with G.711 Fax Fallback

Using the T.38 fax protocol requires to set the **fax protocol t.38** command. We recommend enabling the **ECM** mode. For this, you need to be certain that the **fax-relay ecm disable** command is **NOT** used.

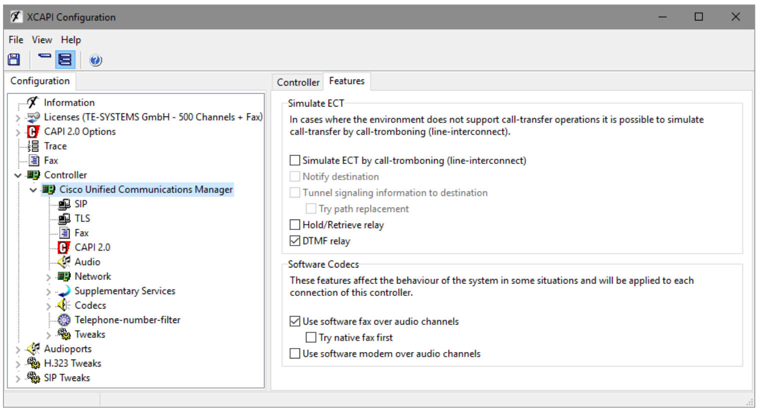
SIP dial-peer example for using T.38 with G.711 fallback:

```
dial-peer voice 800 voip
        destination-pattern 8...
        codec g711ulaw
        session protocol sipv2
        session target ipv4:192.168.1.100
        incoming called-number T
        dtmf-relay rtp-nte

        fax protocol t38 version 0 (or version 3 for V.34 support)
        ls-redundancy 0 hs-redundancy 0 fallback pass-through g711ulaw
```

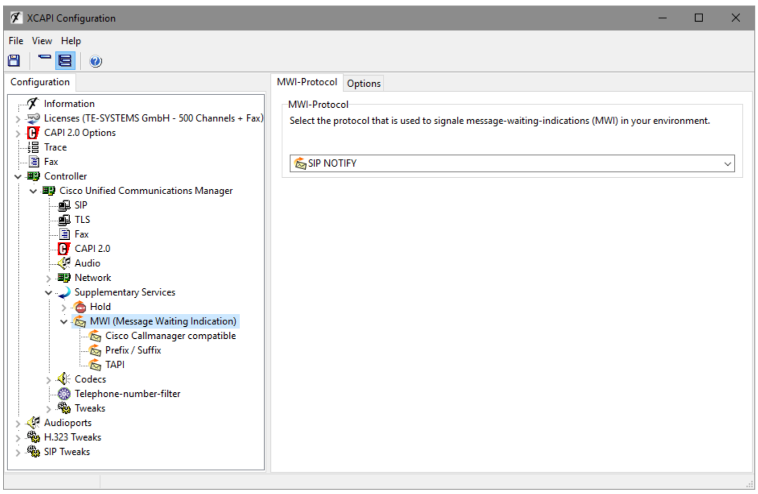
Call Transfer

For enabling call transfer via SIP refer, the **simulated ect by call-tromboning (line-interconnect)** parameter has to be disabled within the XCAPI controller **features** tab. Make certain the SIP Trunk Security Profile parameters **Accept Out-of-Dialog REFER** and **Accept Replaces Header** (see chapter **SIP Trunk Security Profiles** on page 8) and the **Application User Configurations** of the User Management dialog (see chapter **User Management** on page 15) are all enabled. You must also be certain that the corresponding Route Partition is assigned to the SIP trunk's calling search space for allowing proper basic calls and call transfers.



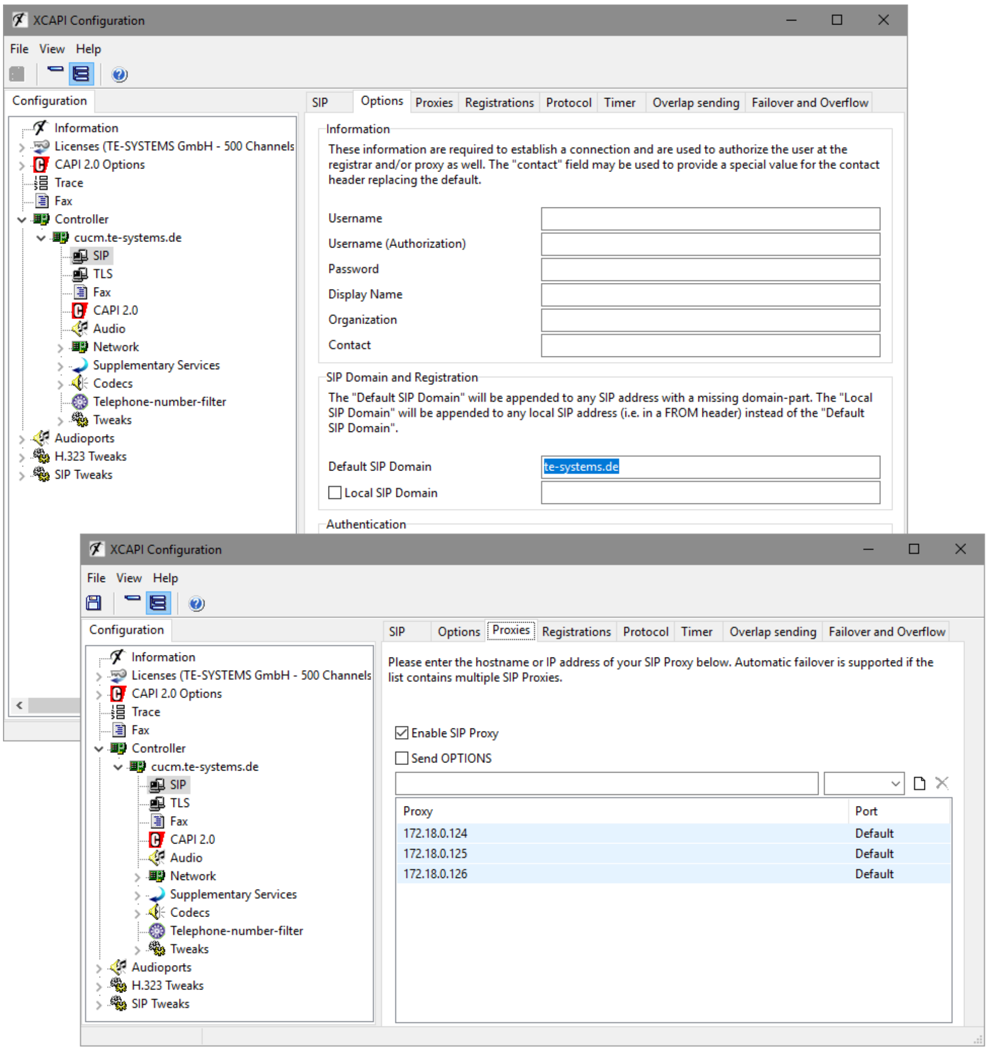
Message Waiting Indications

For Message Waiting Indications via SIP Notify, the **Accept Unsolicited Notification** parameter must be enabled in the SIP Trunk Security Profile. Also check if the **SIP NOTIFY** method is enabled for XCAPI controller.



XCAPI Outbound Failover

A XCAPI related outbound failover can be accomplished with setting up multiple gateway IP addresses within the controller **Proxies** tab. Each gateway has to be available and aware of the XCAPI SIP trunk. If required the valid **Default SIP Domain** of the Cisco environment has to be set within the XCAPI controller **Options** tab, otherwise the system may reject inbound calls from the application if XCAPI uses the wrong host part in SIP URIs. An example is given on the screenshot below.



Exclusion of Liability

Copyright © 2019 TE-SYSTEMS GmbH

All rights reserved

This document, in part or in its entirety, may not be reproduced in any form without the prior consent of TE-SYSTEMS GmbH.

The information contained in this document was correct at the time of writing. TE-SYSTEMS GmbH reserves the right to make any alterations without prior notice.

The utmost care was applied during the compilation of texts and images, as well as during the creation of the software. Nevertheless, no responsibility can be taken for the content being accurate, up to date or complete, nor for the efficient or error-free operation of the software for a particular purpose. Therefore, TE-SYSTEMS GmbH cannot be held liable for any damages resulting directly or indirectly from the use of this document.

Trademarks

All names of products or services used are trademarks or registered trademarks (also without specified indication) of the respective private or legal persons and are therefore subject to legal regulations.

Third Party Disclaimer and Limitations

„OpenSSL", developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/), written by Eric Young (eay@cryptsoft.com) and written by Tim Hudson (tjh@cryptsoft.com).

„MD2, MD4 and MD5 Message Digest Algorithms" via source code derived from the RSA Data Security, Inc.

„RFC 4634 Secure Hash Algorithm", via source code derived from the RFC 4634.

Copyright-Notices

All files included in this sample are copyrighted by TE-SYSTEMS GmbH.

All samples and the SDK may only be used in combination with the XCAPI-product.

The SDK contains code from libtiff with the following copyright-notice:

Copyright (c) 1988-1997 Sam Leffler

Copyright (c) 1991-1997 Silicon Graphics, Inc.

Permission to use, copy, modify, distribute, and sell this software and its documentation for any purpose is hereby granted without fee, provided that (i) the above copyright notices and this permission notice appear in all copies of the software and related documentation, and (ii) the names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED „AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS, WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR PERFORMANCE OF THIS SOFTWARE.