Ribbon SBC SWe Lite Interop with Asterisk, Ribbon SBC SWe Core and Ribbon C20-AS : Interoperability Guide

Interoperable Vendors

Copyright

Document Overview

This document outlines the configuration best practices for Ribbon SBC SWe Lite involving Ribbon's C20, AS, and Ribbon SBC SWeCore when deployed with the Asterisk PBX. This document is intended for Ribbon's C20, AS, and SBC technical staff and any individual tasked with the integration of Ribbon's C20-AS-SBC with the SIP Trunking solution with the Asterisk PBX on the enterprise side.

This guide contains the following configuration sections:

Section-A : Ribbon SBC SWe Lite Configuration

- This section provides a sample of the Ribbon SBC SWe Lite configuration used during the interoperability testing. The commands and configurations provided are only for reference; other configurations are also possible based on the customer's requirement.

Section-B : Ribbon SBC SWe Core Configuration

- This section provides a sample of the Ribbon SBC SWe Core configuration used during the interoperability testing. The commands and configurations provided are only for reference; other configurations are also possible based on the customer's requirement.

Section-C : Ribbon C20-AS Configuration

- This section provides a sample of the Ribbon C20-AS configuration used during the interoperability testing. The commands and configurations provided are only for reference; other configurations are also possible based on the customer's requirement.

Section-D : Asterisk PBX Configuration

- This section provides the procedure for configuring the Asterisk to support connectivity to the Ribbon C20-AS SIP Trunking solution through the SBC. This section requires you to have knowledge of using, configuring, and supporting the Asterisk and experience of working with the product platform. The following sections show you how to configure the Asterisk. You use the config files with root login credentials to configure the Asterisk.

Scope/ Non-Goals

It is not the goal of this guide to provide detailed configurations that will meet the requirements of every customer deployment. Use this guide as a starting point and build Ribbon SBC SWe Lite configurations in consultation with network design and deployment engineers.

Audience

This is a technical document intended for telecommunications engineers for configuring Ribbon SBC SWe Lite and Asterisk PBX. Steps will require navigating the third-party product as well as the Ribbon product, using a graphical user interface (GUI) or a command line interface (CLI).

You must have an understanding of the basic concepts of TCP/UDP/TLS, IP/Routing, and SIP/RTP/SRTP to complete the configuration and to perform any necessary troubleshooting.

Pre-Requisites

Before proceeding with the configuration of the Ribbon SBC SWe Lite, Ribbon SBC SWe Core, and Ribbon C20-AS & Asterisk PBX products, make sure you have the completed the following pre-requisites:

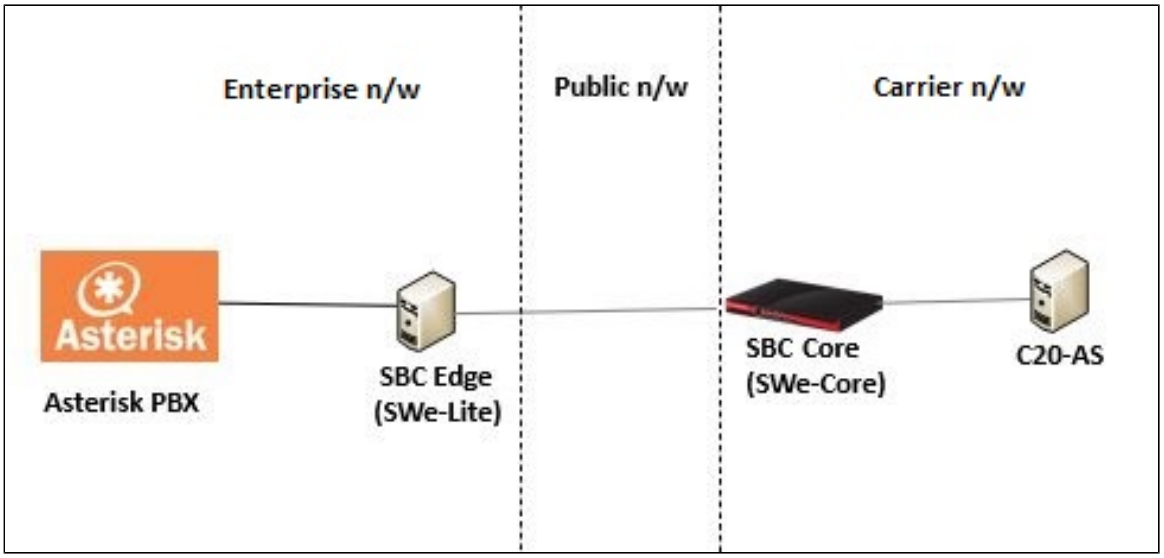
- Ribbon SWe Lite is installed with all required licenses and is running in the network.
- Ribbon SWe Core is installed with all required licenses and is running in the network.
- Ribbon C20-AS is installed with all required licenses and is running in the network.
- Asterisk PBX is installed with all required licenses and is running in the network.

Product and Device Details

	Equipment/ Product	Software Version
Ribbon Communications	SWe Lite	8.3.5 build 19
	SWe Core	7.0.0R0
	C20	R19
	AS	AS12.1 (MCP_19.0.20.2_2017-11-28-1107)
Third-Party Products	Asterisk PBX	1.18

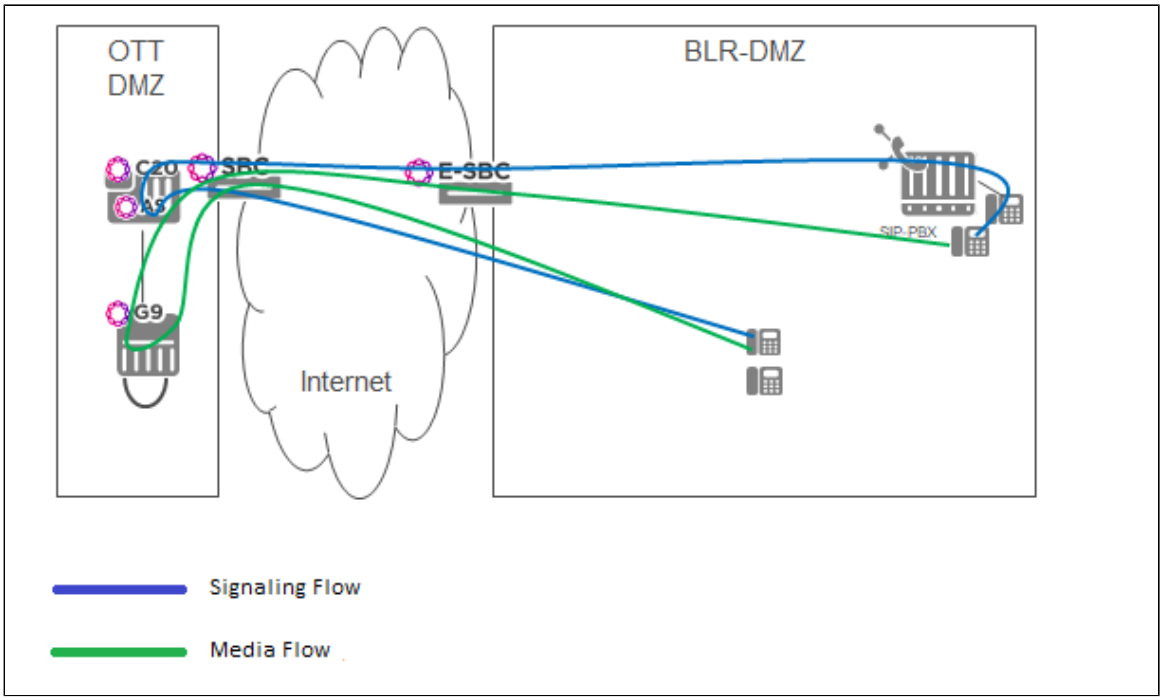
Network Topology Diagram

Deployment Topology



Interoperability Test Lab Topology

The following figure depicts the high-level architecture of the IOT, covering call flow and its overall topology.



Section-A : Ribbon SBC SWe Lite Configuration

You configure Ribbon SBC SWe Lite through the Ribbon SBC SWe Lite's integrated web server. This guide assumes that the operator has already completed the initial configuration that positions the Ribbon SBC SWe Lite on the IP network. To start the configuration, use a standard web browser to connect to the IP or FQDN address of the SBC. Supply the username and password to complete the login process.

Asterisk PBX Side Configuration

Use the Initial Task configuration procedure to position the Ribbon SBC Swe Lite between the Ribbon SBC Core and Asterisk PBX. This task creates SIP components and call routing basics. Create profiles with a specific set of characteristics that correspond to the Asterisk PBX. This profile creation includes configuring the following entities on the SBC Edge 1000:

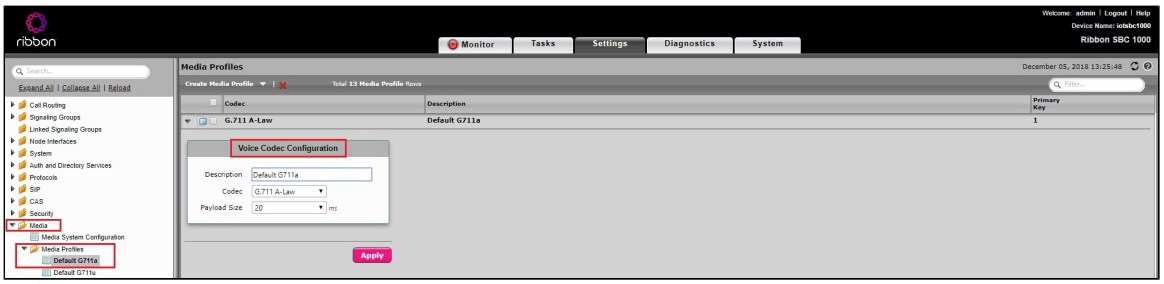
- Media Profile
- Media List
- SIP Profile
- SIP Server Table
- Signaling Group
- Call Routing Table

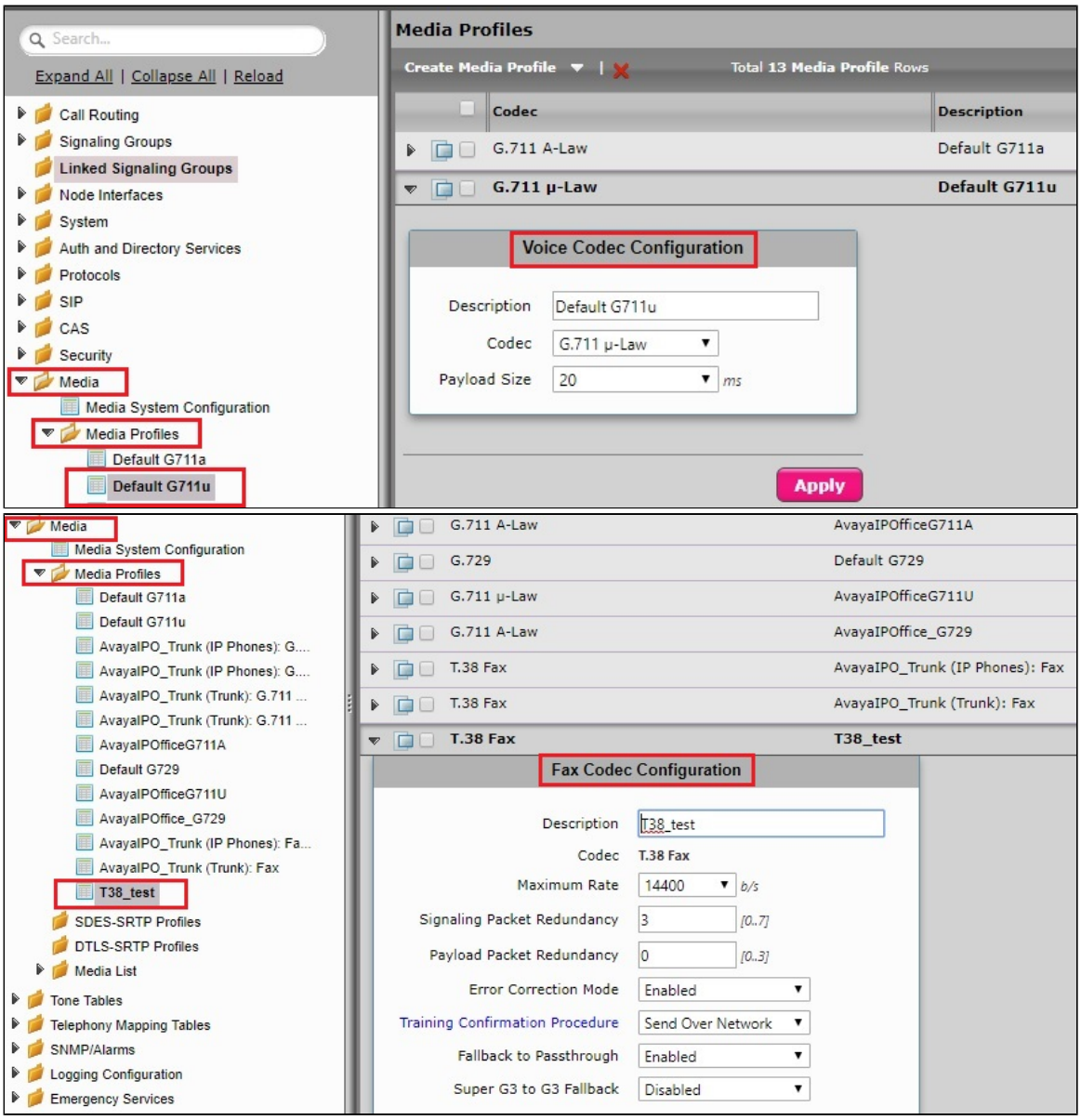
Media Profile

Create a Media Profile towards the Interactive Intelligence side with G711a law as the first codec, G711u law as the second codec, and T.38 Fax as the third codec.

To create or modify a Media Profile:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **Media > Media Profile**.



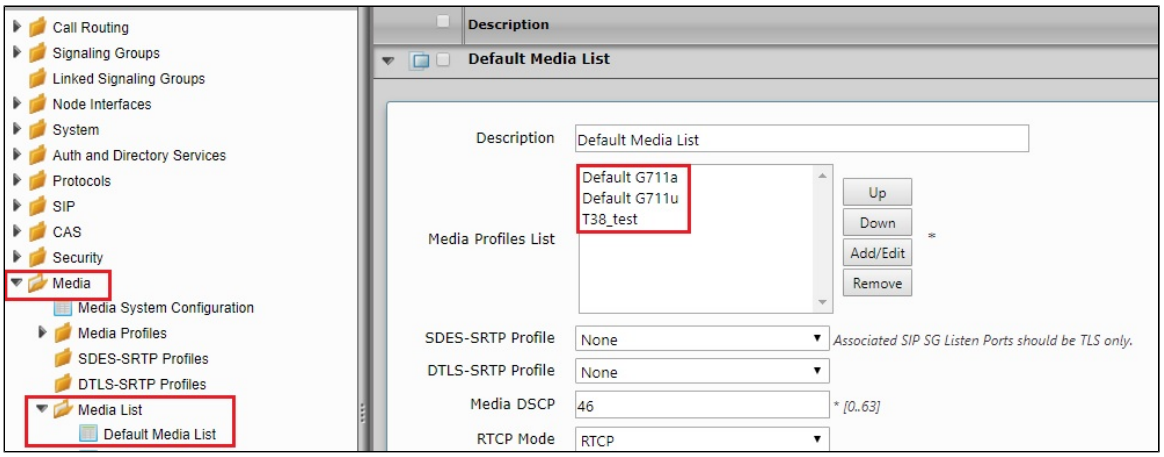


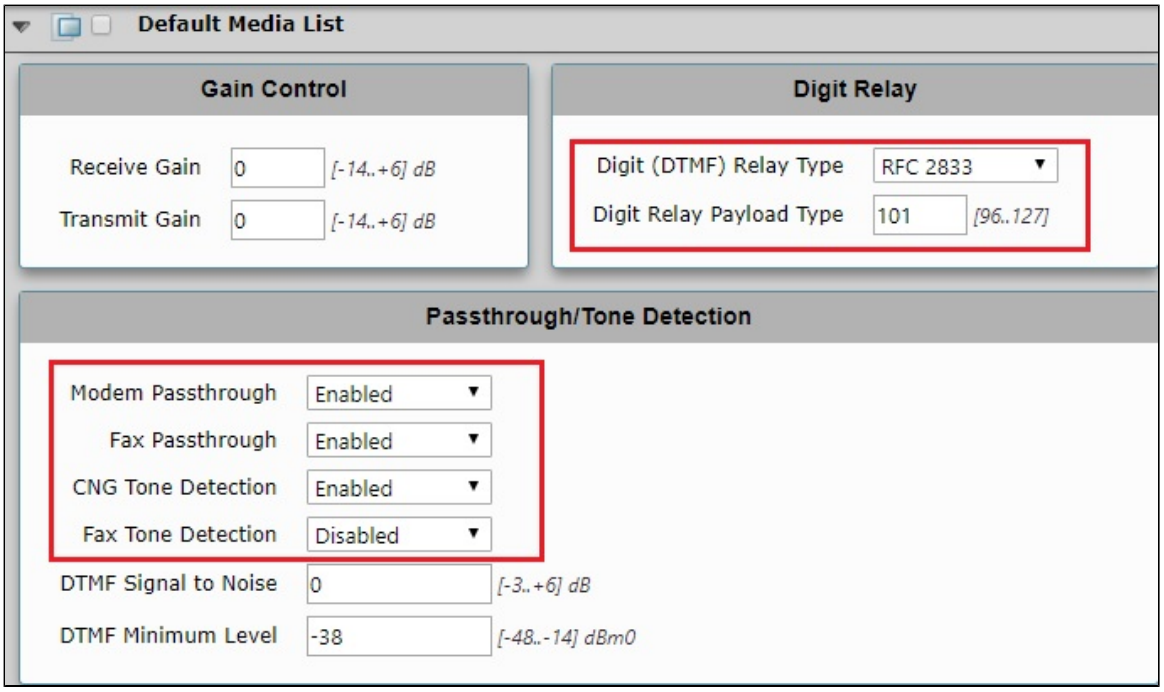
Media List

Media List allows you to specify a set of codecs and fax profiles that are allowed on a given SIP Signaling Group. They contain one or more Media Profiles, which you must first define in Media List. These lists allow you to accommodate specific transmission requirements and SIP devices that only implement a subset of the available voice codecs.

To create or modify a Media List:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **Media > Media List**.

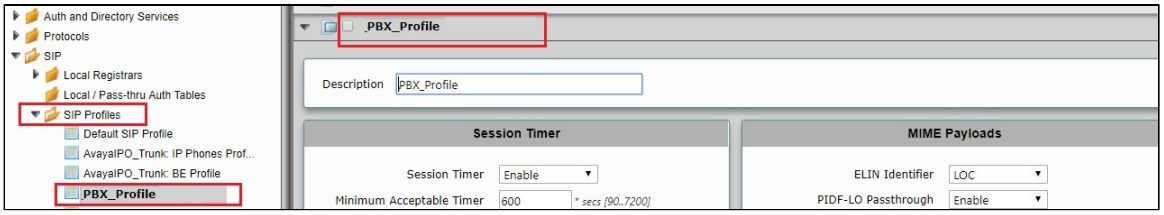


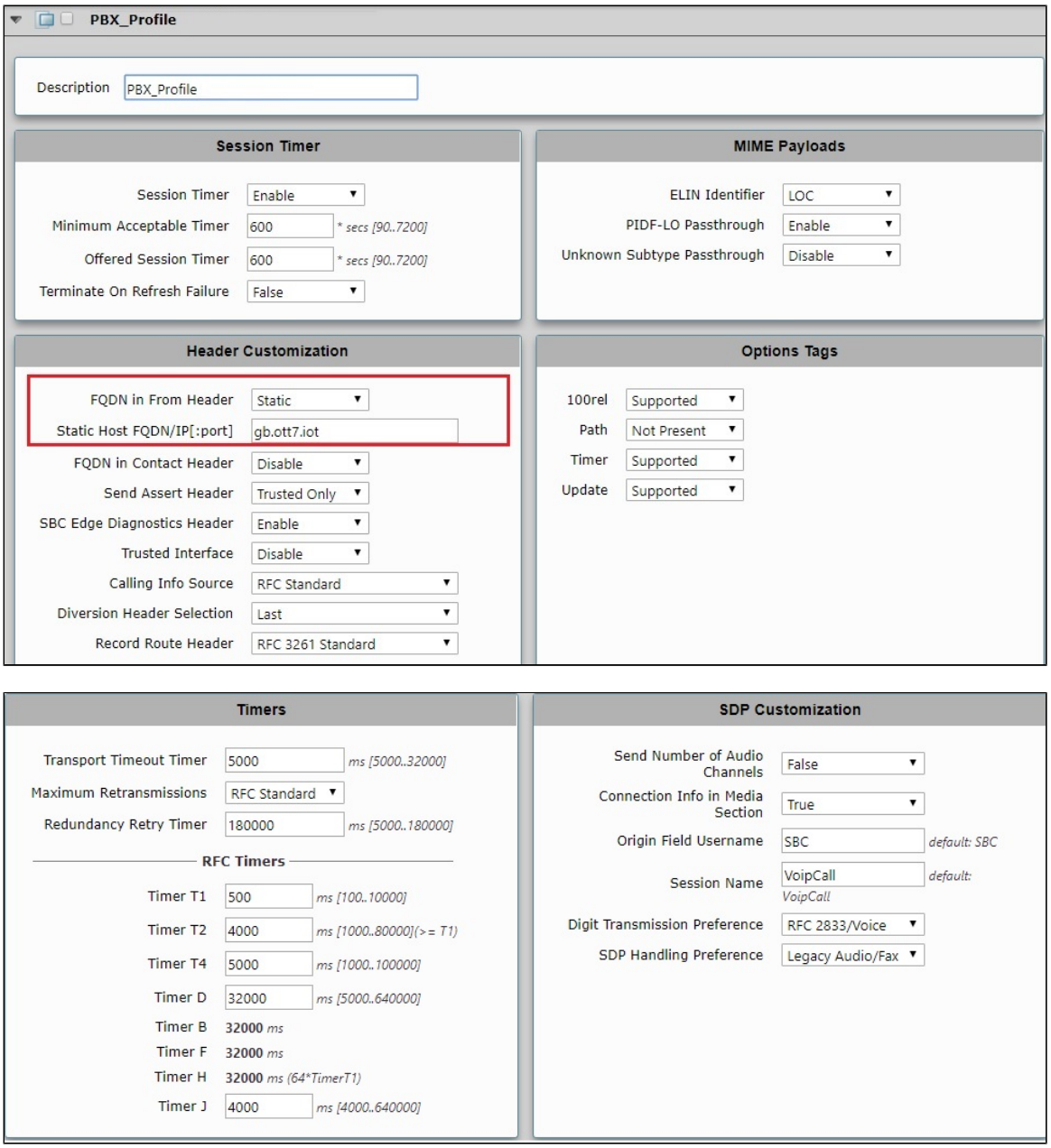


SIP Profile

To create or modify an existing SIP Profile:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **SIP > SIP Profiles**.



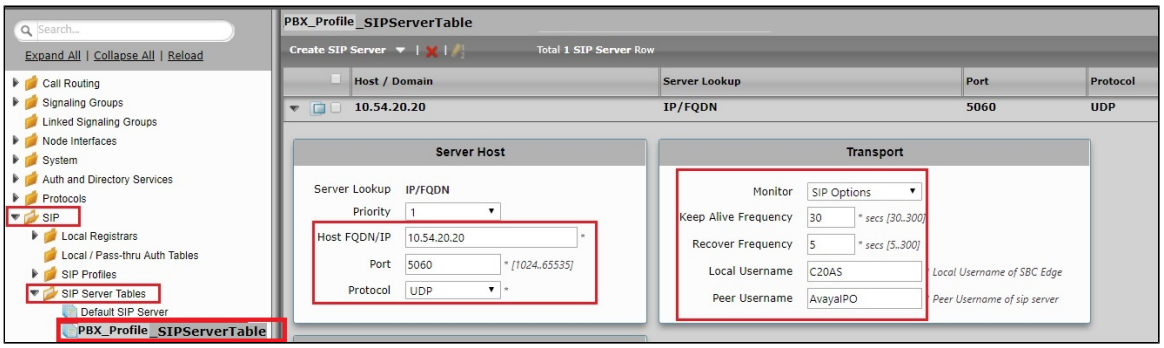


SIP Server Table

SIP Server Tables contain information about the SIP devices connected to the SBC Edge 1000. The entries in the tables provide information about the IP Addresses, ports, and transport protocols used to communicate with each server. The table entries also contain links to counters that are useful for troubleshooting.

To create or modify an existing SIP Server Table:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **SIP > SIP Server Tables**.



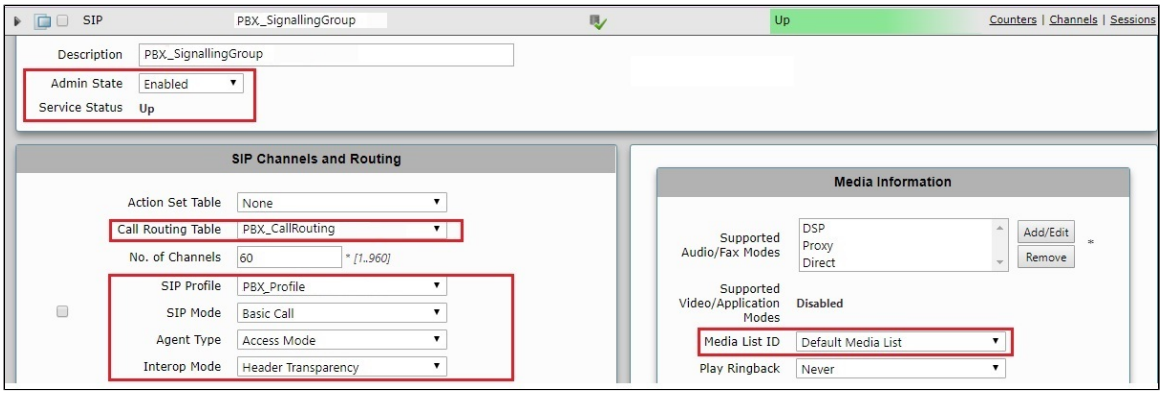
Signaling Group

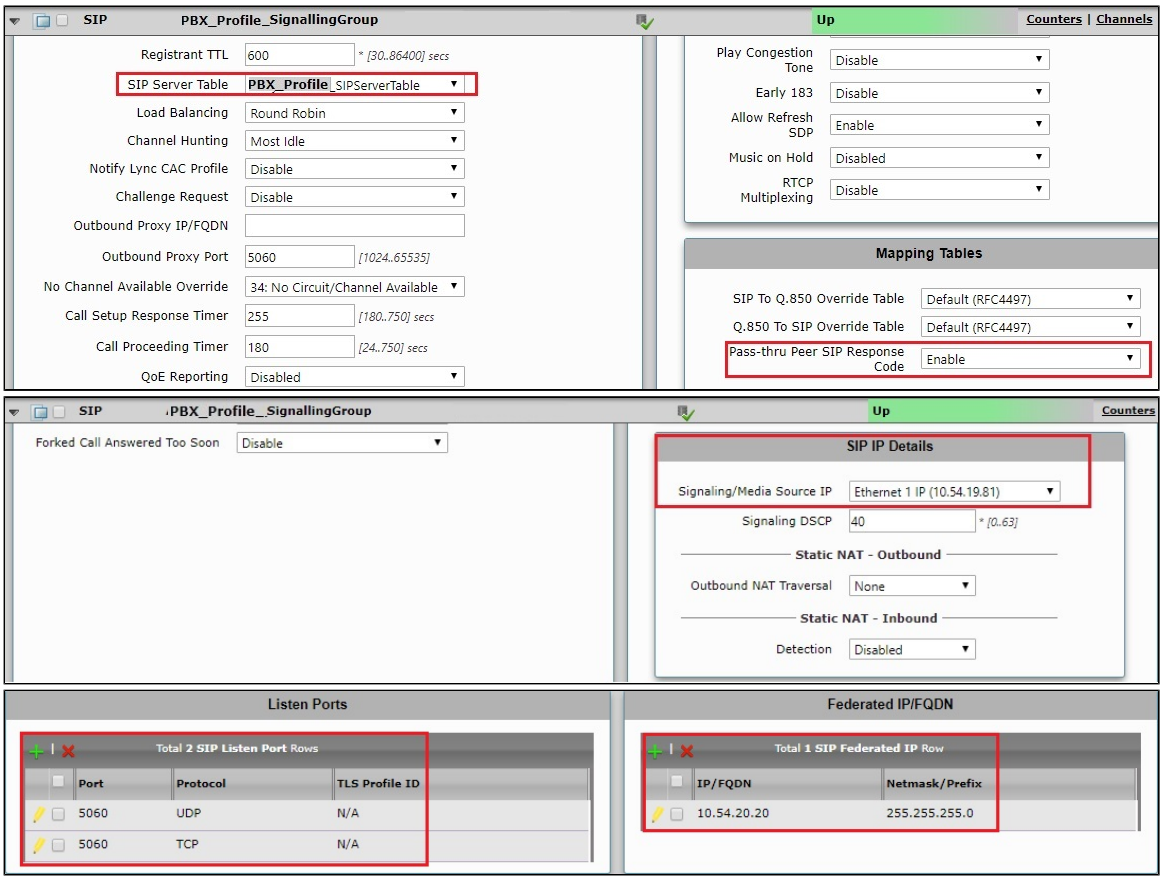
Signaling groups allow grouping telephony channels for routing and shared configuration. They are used for

- routing calls and selecting Call Routes
- selecting Tone Tables and Action Sets
- specifying protocol settings and links to server, media, and mapping tables for SIP

To create or modify an existing Signaling Group:

- In the WebUI, click the **Settings** tab.
- In the left navigation pane, click **Signaling Groups**.
- From the Create Signaling Group drop-down box, click **SIP Signaling Group**.





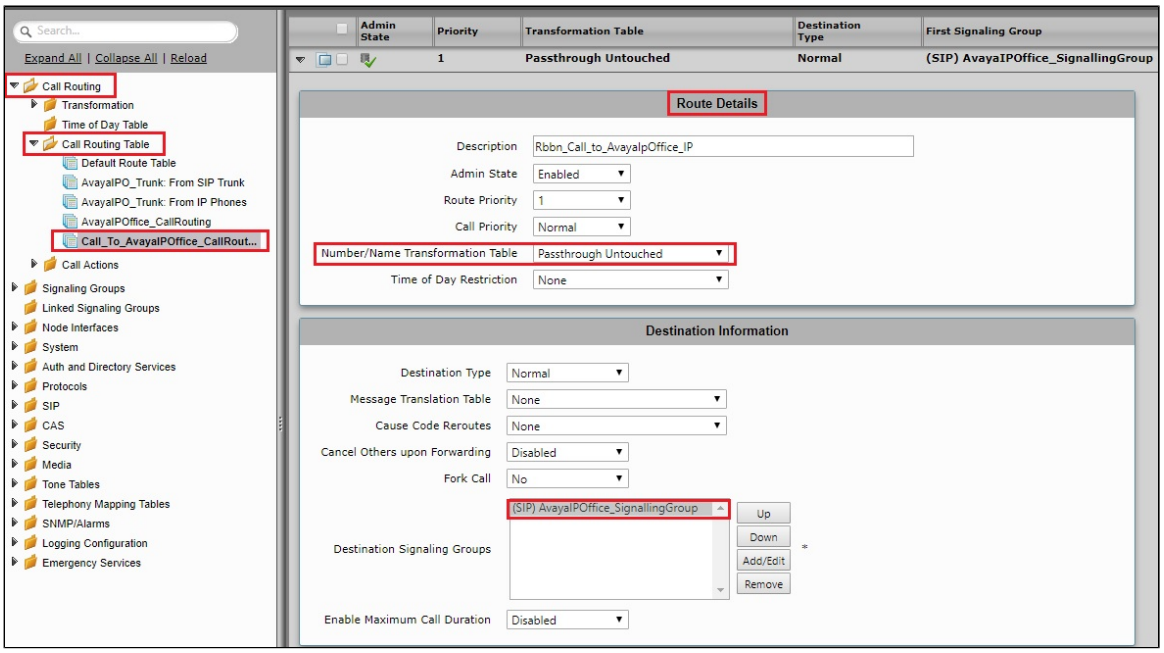
IPAddress/FQDN in Federated IP/FQDN: Specifies the IP Address or Fully Qualified Domain Name of a server from which the SBC Edge 1000 accepts SIP messages. Make sure that it contains the IP address of Asterisk PBX.

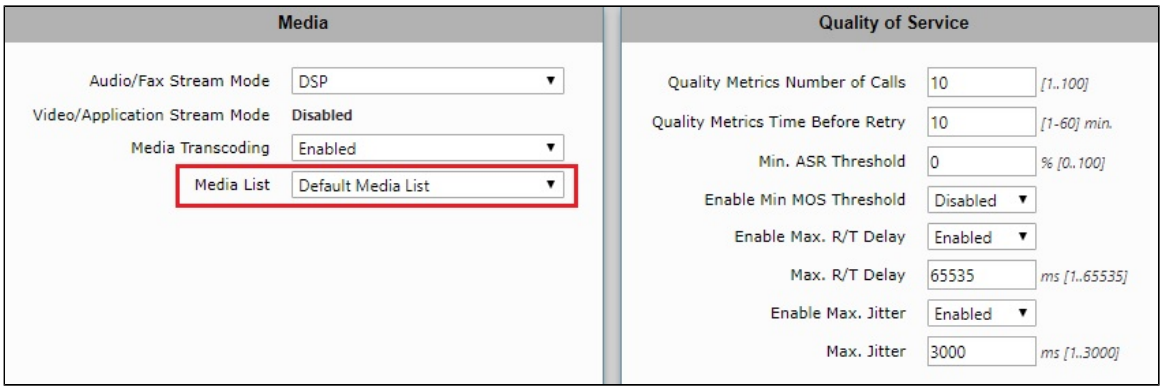
Call Routing Table

Call Routing tables allow you to configure flexible routes for transferring calls between Signaling Groups and for translating the calls. They allow c all transfers between ports and protocols, such as ISDN to SIP. These tables serve as one of the central connection points of the system, linking Transformation Tables, Message Translations, Cause Code Reroute Tables, Media Lists, and the three types of Signaling Groups (ISDN, SIP, and CAS).

To create or modify an existing Call Routing Table:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, click **Call Routing Table**.





The following list defines the fields in the Call Routing Table:

- **Destination Signaling Groups**: Specifies the Signaling Groups used as the destination of calls.
- **Audio/Fax Stream Mode**: Set the Media Mode value to RTP DSP.
- **Media Transcoding:** Enable this field for the Ribbon SBC SWe Lite for transcoding.
- **Media List:** Specifies the Media List used for this call route.

SBC Core Side Configuration

Create profiles with a specific set of characteristics that correspond to the SBC Core. It includes configuring the following entities on the SBC Edge 1000:

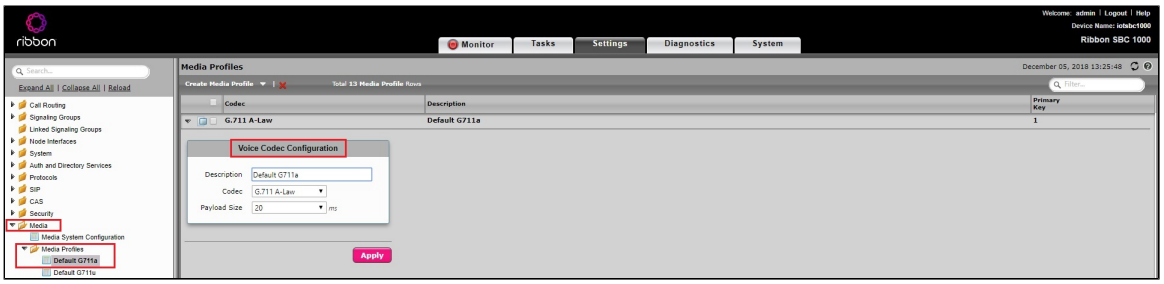
- Media Profile
- Media List
- SIP Profile
- SIP Server Table
- Signaling Group
- Call Routing Table

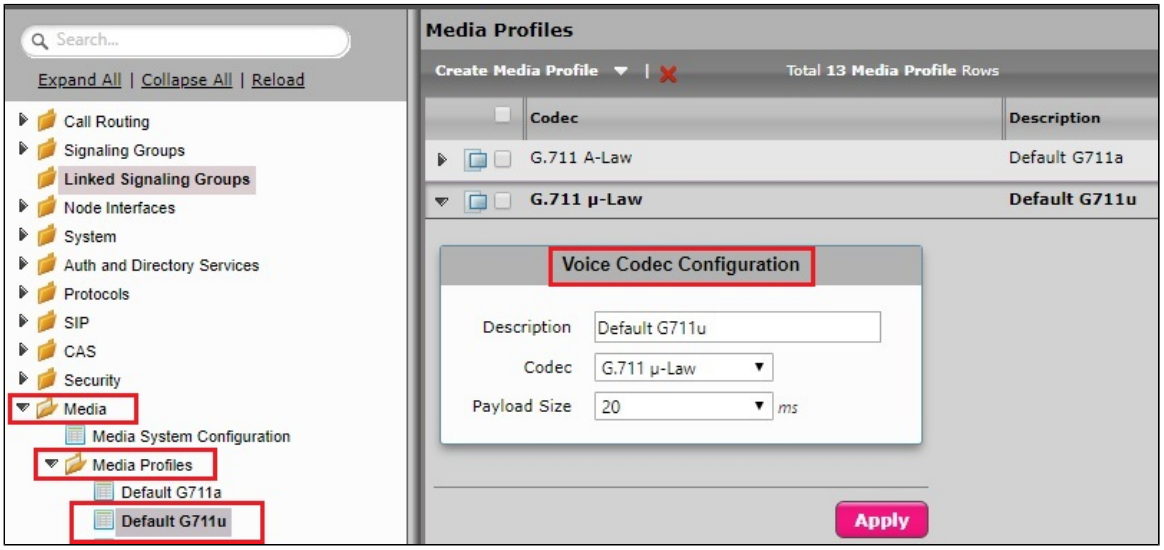
Media Profile

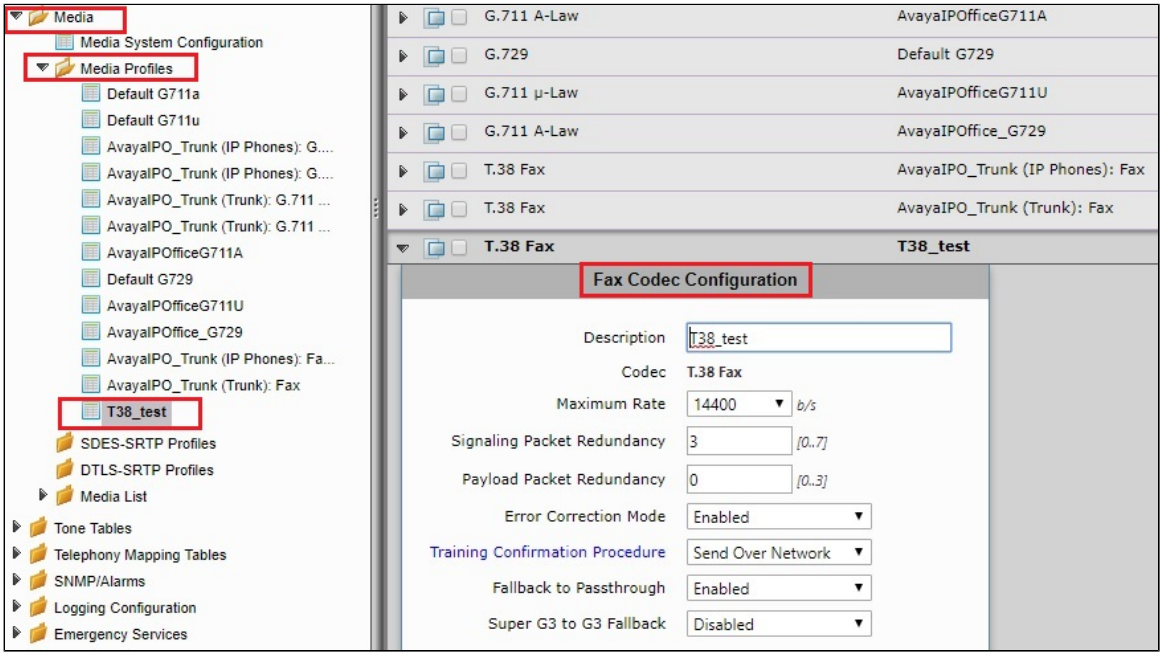
Create a Media Profile for the Interactive Intelligence side with G711a law as the first codec, G711u law as the second codec, and T.38 Fax as the third codec.

To create or modify a Media Profile:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **Media > Media Profile**.





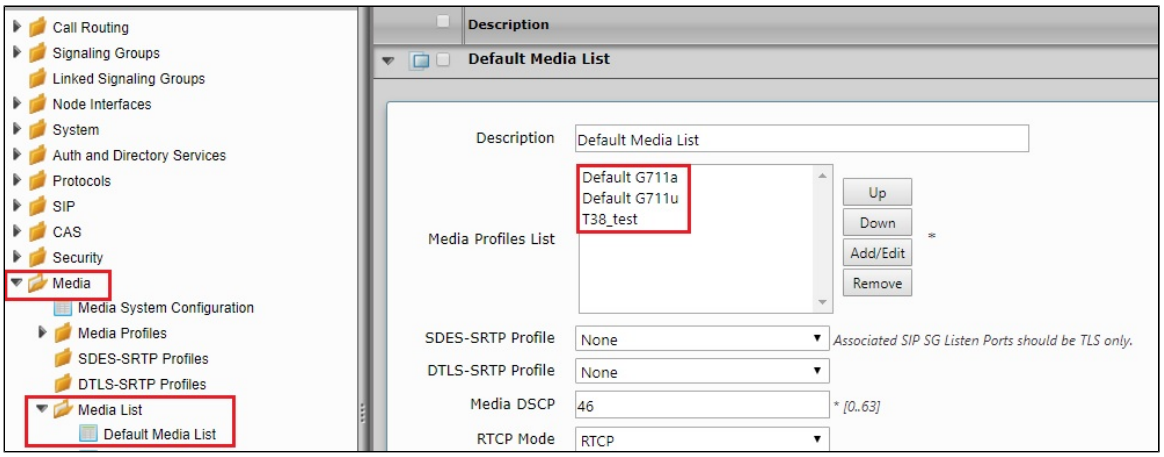


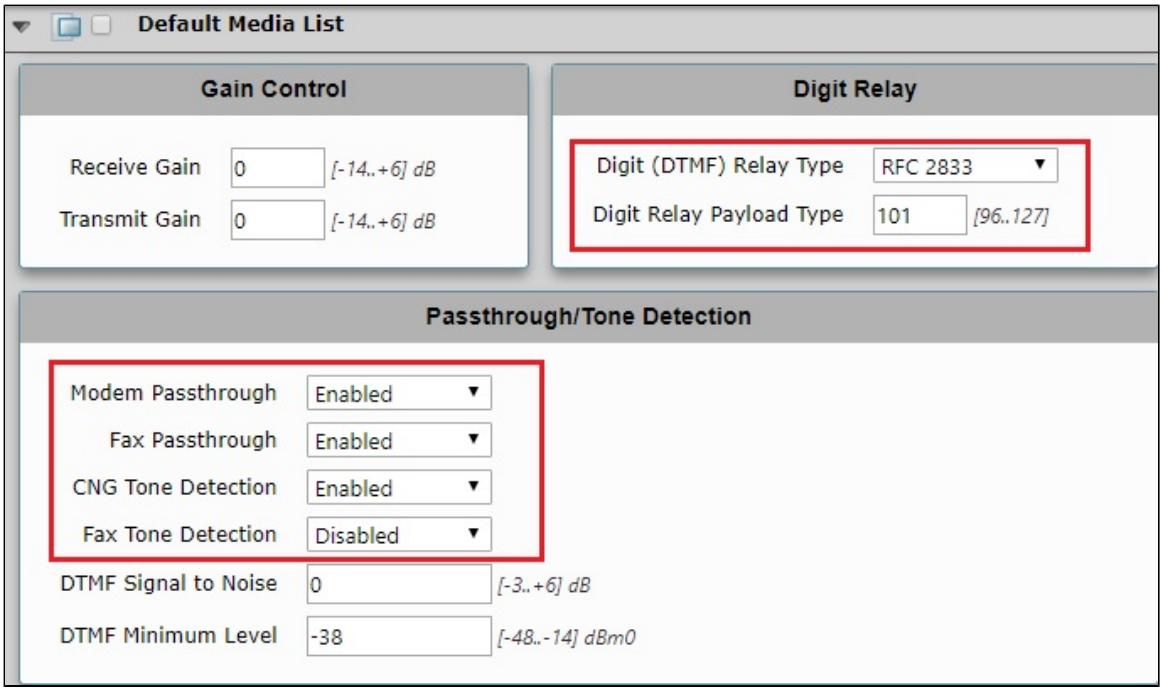
Media List

Media List allows you to specify a set of codecs and fax profiles that you can configure on a specifi SIP Signaling Group. They contain one or more Media Profiles, which you must first define in the Media List. These lists allow you to accommodate specific transmission requirements and SIP devices that only implement a subset of the available voice codecs.

To create or modify a Media List:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **Media > Media List**.

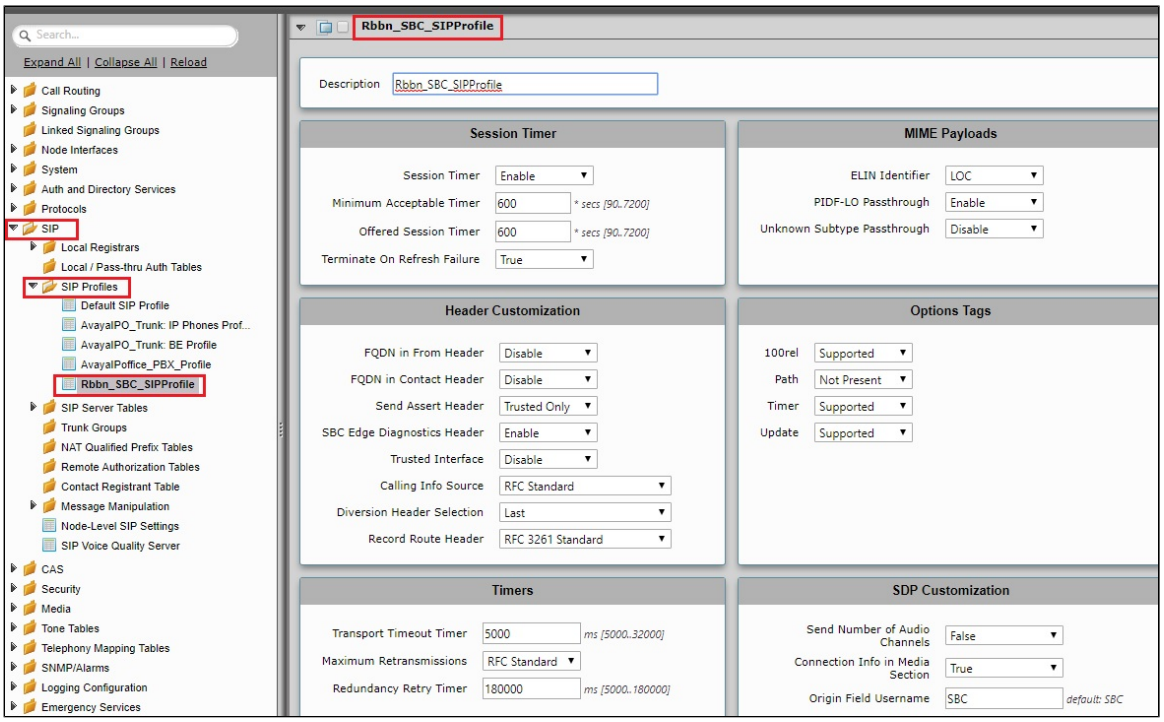


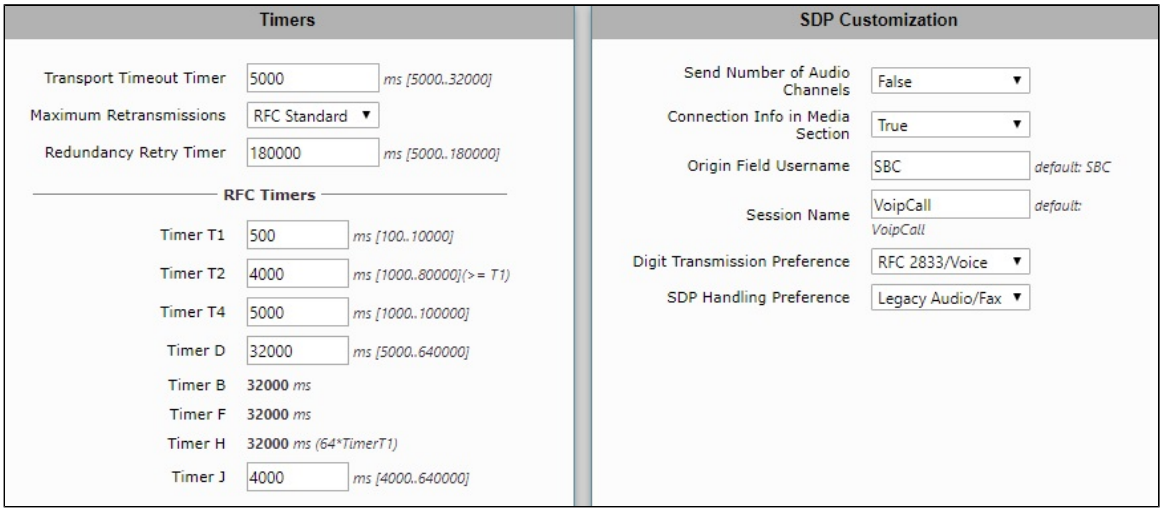


SIP Profile

To create or modify an existing SIP Profile:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select**SIP > SIP Profiles**.



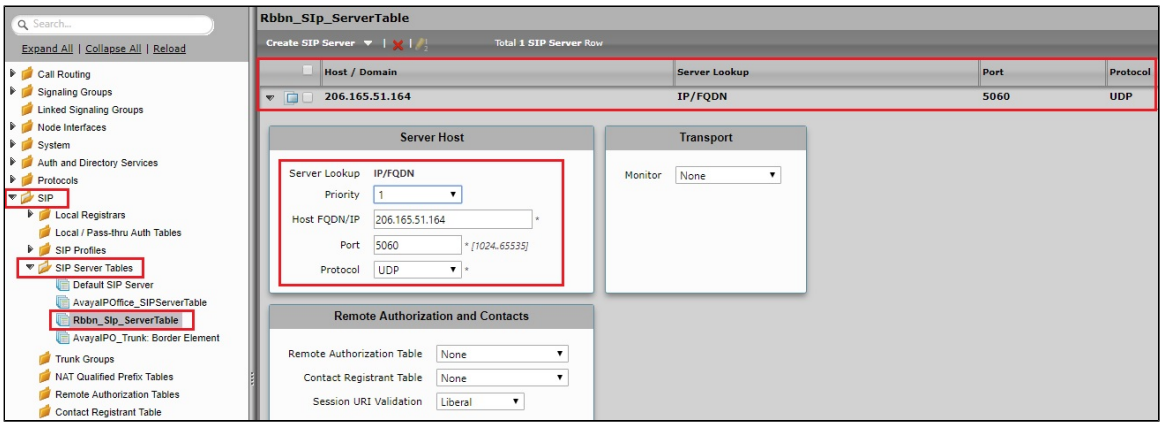


SIP Server Table

SIP Server Tables contain information about the SIP devices connected to the SBC Edge 1000. The entries in the tables provide information about the IP Addresses, ports, and transport protocols used to communicate with each server. The table entries also contain links to counters that are useful for troubleshooting.

To create or modify an existing SIP Server Table:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, select **SIP > SIP Server Tables**.



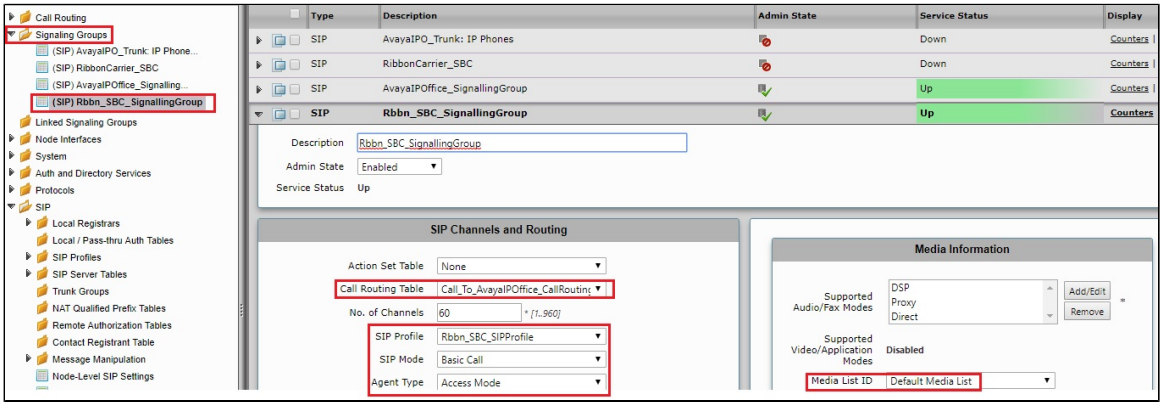
Signaling Group

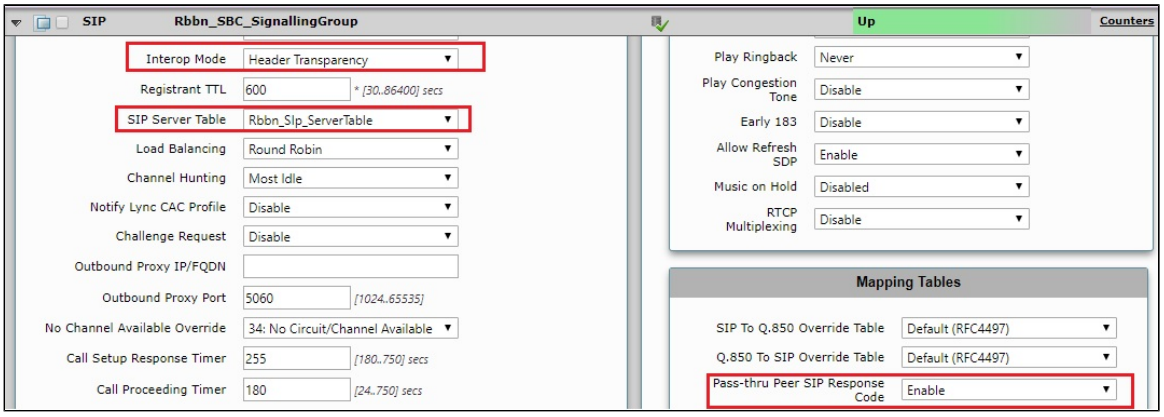
Signaling groups allow grouping telephony channels for routing and shared configuration. They are used for

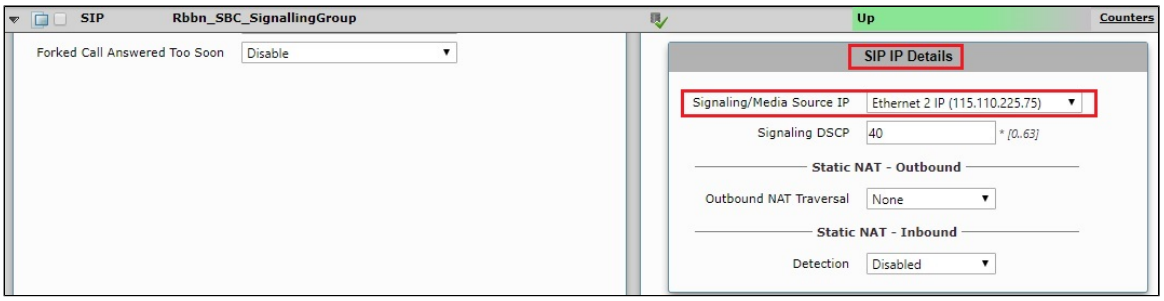
- routing calls and selecting Call Routes
- selecting Tone Tables and Action Sets
- specifying protocol settings and link to server, media, and mapping tables for SIP

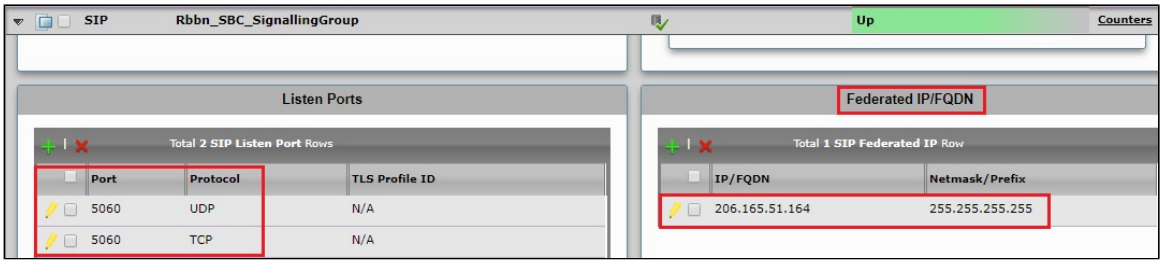
To create or modify an existing Signaling Group:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, click **Signaling Groups**.
3. From the Create Signaling Group drop-down box, select **SIP Signaling Group**.









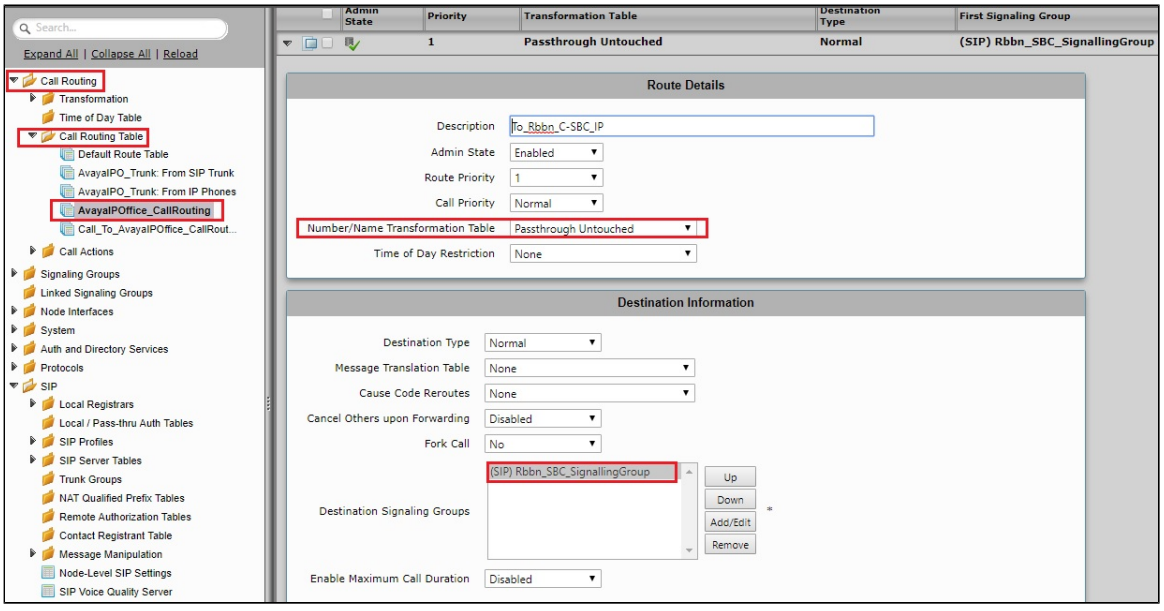
IP Address/FQDN in Federated IP/FQDN: Specifies the IP Address or Fully Qualified Domain Name of a server from which the SBC Edge 1000 accepts SIP messages. Make sure that it contains the IP address of the Asterisk PBX.

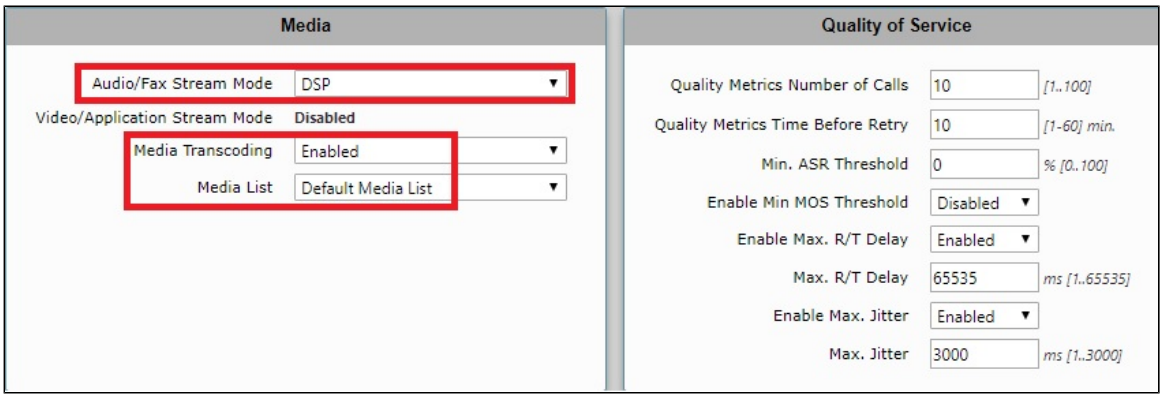
Call Routing Table

Call Routing tables allow you to configure flexible routes for transferring calls between Signaling Groups and for translating the calls. They allow c all transfers between ports and protocols, such as ISDN to SIP. These tables serve as one of the central connection points of the system, linking Transformation Tables, Message Translations, Cause Code Reroute Tables, Media Lists, and the three types of Signaling Groups (ISDN, SIP, and CAS).

To create or modify an existing Call Routing Table:

1. In the WebUI, click the **Settings** tab.
2. In the left navigation pane, click **Call Routing Table**.





The following list defines the fields in the Call Routing Table:

- **Destination Signaling Groups**: Specifies the Signaling Groups used as the destination of calls.
- **Audio/Fax Stream Mode**: Set the Media Mode field value to RTP DSP.
- **Media Transcoding:** Enable this field for the Ribbon SBC SWe Lite for transcoding.
- **Media List:** Specifies the Media List used for this call route.

Dynamic Registration SIP PBX Setup

Currently, the Ribbon SBC Edge does not support the Dynamic Registration feature. Therefore, dynamic PBX registration related configuration is out of scope in this configuration guide.

Section-B : Ribbon SBC SWe Core Configuration

This section provides a sample of the Ribbon SBC SWe configuration used during the interoperability testing. The following commands and configurations are only for reference; other configurations are also possible based on the customer's requirements.

Global Configuration

Codec Entry

Create a Codec Entry with the supported codec in the network.

```
set profiles media codecEntry G729A-IOT-TEST dtmf relay rfc2833 set profiles media codecEntry G729A-IOT-TEST
packetSize 20 commit   set profiles media codecEntry G711_ALAW_PTIME_20 dtmf relay rfc2833 set profiles media
codecEntry G711_ALAW_PTIME_20 packetSize 20 commit
```

RTCP

Configure the RTCP interval.

```
set system media mediaRtcpControl senderReportInterval 5
commit
```

DSP Resource Allocation

This configuration only applies if the SBC is deployed with the (hardware) DSP resources. If it is not, executing this configuration doe not have any negative impact.

> The subsequent configuration section (Packet Service Profiles) does not attempt transcoding, so the lack of compression resources does not impact the overall SBC configuration in this document.

```
set system mediaProfile compression 75 tone 25
commit
```

SBC Configuration for C20/AS Side

Packet Service Profile (PSP)

Create a Packet Service Profile (PSP) for the C20/Application Server (AS). Specify the PSP within the SIP trunk group configuration.

```
set profiles media packetServiceProfile CORE_PSP codec codecEntry1 G729A-IOT-TEST
set profiles media packetServiceProfile CORE_PSP codec codecEntry2 G711_ALAW_PTIME_20
set profiles media packetServiceProfile CORE_PSP packetToPacketControl transcode transcoderFreeTransparency
set profiles media packetServiceProfile CORE_PSP packetToPacketControl codecsAllowedForTranscoding thisLeg ""
set profiles media packetServiceProfile CORE_PSP packetToPacketControl codecsAllowedForTranscoding otherLeg ""
set profiles media packetServiceProfile CORE_PSP rtcpOptions rtcp enable
set profiles media packetServiceProfile CORE_PSP preferredRtpPayloadTypeForDtmfRelay 101
commit
```

IP Signaling Profile (IPSP)

Create an IP signaling profile for the C20/AS side. Specify the PSP within the SIP trunk group configuration.

```
set profiles signaling ipSignalingProfile CORE_IPSP
set profiles signaling ipSignalingProfile CORE_IPSP ipProtocolType sipOnly
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes flags
includeTransportTypeInContactHeader enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes flags
minimizeRelayingOfMediaChangesFromOtherCallLegAll enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes flags
relayDataPathModeChangeFromOtherCallLeg enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes flags noPortNumber5060 enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes relayFlags dialogEventPackage enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes relayFlags info enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes relayFlags notify enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes relayFlags refer enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes relayFlags statusCode4xx6xx enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes relayFlags updateWithoutSdp enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes transparencyFlags authcodeHeaders enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes transparencyFlags mwiBody enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes transparencyFlags referredByHeader
enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes transparencyFlags sipfragBody enable
set profiles signaling ipSignalingProfile CORE_IPSP commonIpAttributes transparencyFlags unknownBody enable
set profiles signaling ipSignalingProfile CORE_IPSP egressIpAttributes flags disable2806Compliance enable
set profiles signaling ipSignalingProfile CORE_IPSP egressIpAttributes privacy transparency enable
set profiles signaling ipSignalingProfile CORE_IPSP ingressIpAttributes flags sendUpdatedSDPin200Ok enable
set profiles signaling ipSignalingProfile CORE_IPSP egressIpAttributes transport type1 udp
set profiles signaling ipSignalingProfile CORE_IPSP egressIpAttributes transport type2 tcp commit
```

IP Interface Group

Create an IP interface group and assign its interface and IP address.

```
set addressContext default ipInterfaceGroup CORE_LIF ipInterface IPIF2 ceName IOTSBC1 portName pkt0
set addressContext default ipInterfaceGroup CORE_LIF ipInterface IPIF2 ipAddress 172.28.xxx.xx
set addressContext default ipInterfaceGroup CORE_LIF ipInterface IPIF2 prefix xx
set addressContext default ipInterfaceGroup CORE_LIF ipInterface IPIF2 mode inService state enabled
commit
```

Zone

Zone groups the set of objects that communicate to the Session Manager (SESM) AS.

```
set addressContext default zone CORE id 3
set addressContext default zone CORE remoteDeviceType appServer
commit
```

SIP Signaling Port

A SIP signaling port is a logical address that sends and receives SIP call signaling packets and is permanently bound to a specific zone.

```
set addressContext default zone CORE sipSigPort 3 ipInterfaceGroupName CORE_LIF ipAddressV4 172.28.xxx.xx
portNumber 5060 transportProtocolsAllowed sip-udp,sip-tcp
set addressContext default zone CORE sipSigPort 3 state enabled mode inService
commit
```

DNS Group

DNS Groups set DNS objects that you may use for DNS resolution within a particular Zone.

```
set addressContext default dnsGroup EXT_DNS
set addressContext default dnsGroup EXT_DNS type ip interface IPIF2 server DNS2 ipAddress x.x.x.x state enabled
set addressContext default zone CORE dnsGroup EXT_DNS
commit
```

SIP Trunk Group

Create a SIP Trunk Group for the SESM-AS side and assign the IPSP and PSP that you configured above. For `ingressIpPrefix`, replace `x.x.x.x` with the IP address prefix that you want to allow. You can configure multiple SESM IP addresses

```
set addressContext default zone CORE sipTrunkGroup CORE_STG media mediaIpInterfaceGroupName CORE_LIF
set addressContext default zone CORE sipTrunkGroup CORE_STG signaling honorMaddrParam enabled
set addressContext default zone CORE sipTrunkGroup CORE_STG policy media packetServiceProfile CORE_PSP
set addressContext default zone CORE sipTrunkGroup CORE_STG policy signaling ipSignalingProfile CORE_IPSP
set addressContext default zone CORE sipTrunkGroup CORE_STG services dnsSupportType a-srv-naptr
set addressContext default zone CORE sipTrunkGroup CORE_STG ingressIpPrefix x.x.x.x x
set addressContext default zone CORE sipTrunkGroup CORE_STG signaling relayNonInviteRequest enabled
set addressContext default zone CORE sipTrunkGroup CORE_STG media sdpAttributesSelectiveRelay enabled
set addressContext default zone CORE sipTrunkGroup CORE_STG mode inService state enabled
commit
```

IP Static Route

Create a default route to the subnet's next hop IP for the interface and IP Interface Group.

```
set addressContext default staticRoute X.X.X.X X x.X.X.X CORE_LIF IPIF2 preference 100
commit
```

IP Peer

Create an IP Peer with the AS IP address and assign it to the CORE zone.

```
set addressContext default zone CORE ipPeer CORE_PEER ipAddress x.x.x.x ipPort 5060
commit
```

SBC Configuration for the Enterprise Ribbon SBC SWe Lite side

Packet Service Profile (PSP)

Create a Packet Service Profile (PSP) for the SBC Edge side. Specify the PSP within the SIP trunk group configuration.

```
set profiles media packetServiceProfile ACCESS_PSP codec codecEntry1 G729A-IOT-TEST
set profiles media packetServiceProfile ACCESS_PSP codec codecEntry2 G711_ALAW_PTIME_20
set profiles media packetServiceProfile ACCESS_PSP packetToPacketControl transcode transcoderFreeTransparency
set profiles media packetServiceProfile ACCESS_PSP packetToPacketControl codecsAllowedForTranscoding thisLeg ""
set profiles media packetServiceProfile ACCESS_PSP packetToPacketControl codecsAllowedForTranscoding otherLeg
""
set profiles media packetServiceProfile ACCESS_PSP rtcpOptions rtcp enable
set profiles media packetServiceProfile ACCESS_PSP preferredRtpPayloadTypeForDtmfRelay 101
commit
```

IP Signaling Profile (IPSP)

Create an IP Signaling Profile (IPSP) for the SBC Edge side. Specify the IPSP within the SIP trunk group configuration.

```
set profiles signaling ipSignalingProfile ACCESS_IPSP set profiles signaling ipSignalingProfile ACCESS_IPSP
ipProtocolType sipOnly
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes flags
includeTransportTypeInContactHeader enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes flags
minimizeRelayingOfMediaChangesFromOtherCallLegAll enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes flags
relayDataPathModeChangeFromOtherCallLeg enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes flags noPortNumber5060 enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes relayFlags dialogEventPackage enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes relayFlags info enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes relayFlags notify enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes relayFlags refer enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes relayFlags statusCode4xx6xx enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes relayFlags updateWithoutSdp enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes transparencyFlags authcodeHeaders
enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes transparencyFlags mwiBody enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes transparencyFlags referredByHeader
enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes transparencyFlags sipfragBody enable
set profiles signaling ipSignalingProfile ACCESS_IPSP commonIpAttributes transparencyFlags unknownBody enable
set profiles signaling ipSignalingProfile ACCESS_IPSP egressIpAttributes flags disable2806Compliance enable
set profiles signaling ipSignalingProfile ACCESS_IPSP egressIpAttributes privacy transparency enable
set profiles signaling ipSignalingProfile ACCESS_IPSP ingressIpAttributes flags sendUpdatedSDPin200Ok enable
set profiles signaling ipSignalingProfile ACCESS_IPSP egressIpAttributes transport type1 udp
set profiles signaling ipSignalingProfile ACCESS_IPSP egressIpAttributes transport type2 tcp
commit
```

IP Interface Group

Create an IP Interface Group and assign its interface and IP address for connecting to the SBC Edge.

```
set addressContext default ipInterfaceGroup ACCESS_LIF ipInterface IPIF0 ceName IOTSBC1 portName pkt1
set addressContext default ipInterfaceGroup ACCESS_LIF ipInterface IPIF0 ipAddress xxx.xx.xxx.xx
set addressContext default ipInterfaceGroup ACCESS_LIF ipInterface IPIF0 prefix xx
set addressContext default ipInterfaceGroup ACCESS_LIF ipInterface IPIF0 mode inService state enabled
commit
```

Zone

A Zone groups the set of objects that communicate with the SBC Edge 1000.

```
set addressContext default zone ACCESS id 2
commit
```

SIP Signaling Port

A SIP Signaling port is a logical address permanently bound to a specific zone that sends and receives SIP call signaling packets.

```
set addressContext default zone ACCESS id 2 sipSigPort 2 ipInterfaceGroupName ACCESS_LIF ipAddressV4 XXX.XXX.
XXX.XXX portNumber 5060 transportProtocolsAllowed sip-tcp,sip-udp
set addressContext default zone ACCESS id 2 sipSigPort 2 mode inService state enabled
commit
```

SIP Trunk Group

Create a SIP Trunk Group on the Enterprise SBC Edge 1000 side and assign the PSP and IPSP that you configured above.

```
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG media mediaIpInterfaceGroupName ACCESS_LIF
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG signaling honorMaddrParam enabled
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG policy media packetServiceProfile ACCESS_PSP
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG policy signaling ipSignalingProfile ACCESS_IPSP
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG services dnsSupportType a-srv-naptr
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG ingressIpPrefix x.x.x.x x
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG signaling relayNonInviteRequest enabled
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG media sdpAttributesSelectiveRelay enabled
set addressContext default zone ACCESS sipTrunkGroup ACCESS_STG mode inService state enabled
commit
```

IP Peer

Create an IP Peer with the Fully-Qualified Domain Name (FQDN) or IP address of the endpoint and assign it to the PSTN Side.

```
set addressContext default zone ACCESS ipPeer ACCESS_PEER ipAddress x.x.x.x ipPort 5060
commit
```

IP Static Route

Create a default route to the subnet's next hop IP for the interface and IP Interface Group.

```
set addressContext default staticRoute X.X.X.X X X.X.X.1 ACCESS_LIF IPIF0 preference 100
commit
```

Routing Label

Create a Routing Label with a single Routing Label Route to bind the egress Trunk Group for the C20-AS and the Enterprise SBC Edge 1000 IP Peer.

```
set global callRouting routingLabel ACCESS_RL routingLabelRoute 1 trunkGroup ACCESS_STG ipPeer ACCESS_PEER
inService inService
set global callRouting routingLabel CORE_RL routingLabelRoute 1 trunkGroup CORE_STG ipPeer CORE_PEER inService
inService
commit
```

Routing

Routing allows you send calls to the correct destination. You can use routing options based on your requirements. Configure the standard and specific routes (with usernames) to ensure that no matter how the called party is addressed (a number or username), the SBC routes the message to the Core. Create Route entries for standard Trunk Group routing with Matching Criteria and a Routing Label destination.

```
set global callRouting route trunkGroup ACCESS_STG IOTSBC1 standard Sonus_NULL Sonus_NULL all all ALL none
Sonus_NULL routingLabel CORE_RL
set global callRouting route trunkGroup ACCESS_STG IOTSBC1 username Sonus_NULL Sonus_NULL all all ALL none
Sonus_NULL routingLabel CORE_RL
set global callRouting route trunkGroup CORE_STG IOTSBC1 standard Sonus_NULL Sonus_NULL all all ALL none
Sonus_NULL routingLabel ACCESS_RL
commit
```

Section-C : Ribbon C20-AS Configuration

The following documentation was used to configure the Application Server: C20-EXPERiUS SIP Trunking Solution Guide (630-01981-01_02.01_1.2).

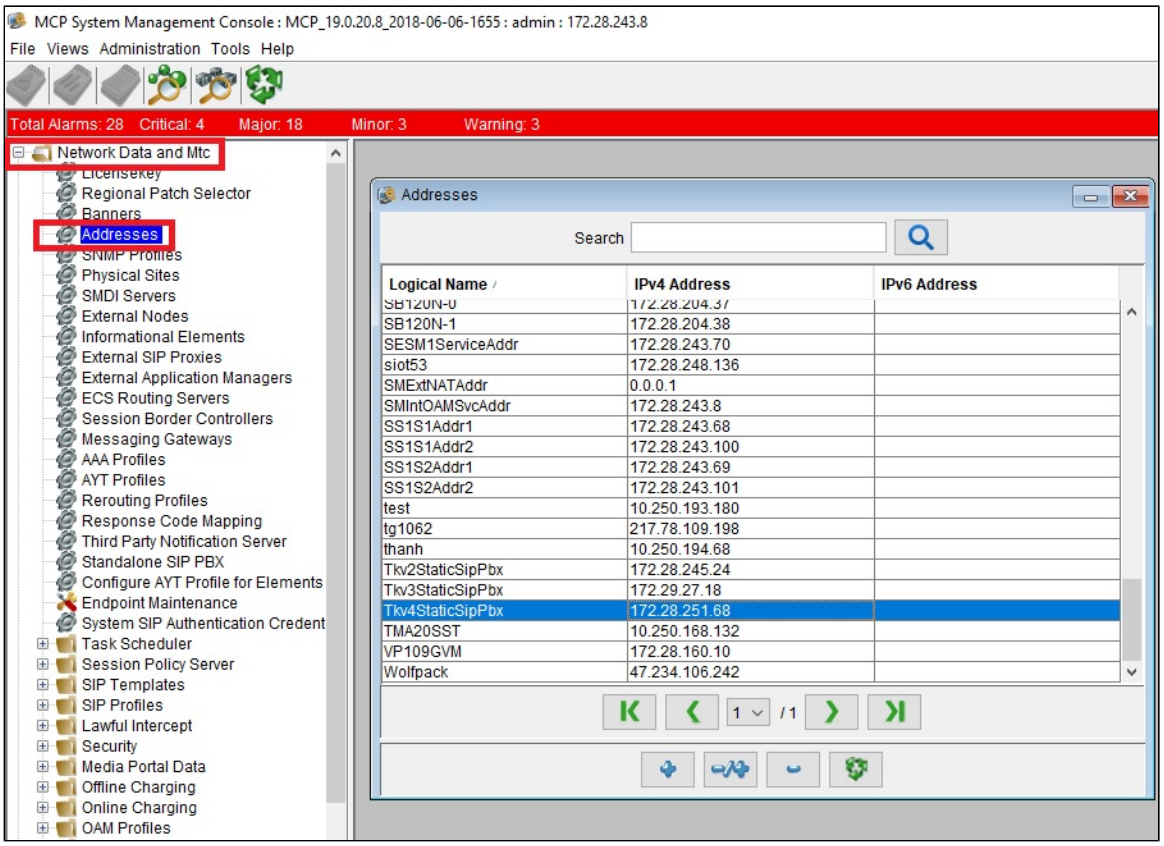
Configure Ribbon AS

System Management Console Configuration

To configure the Static SIP PBX, you will need access to the Ribbon C20-AS MCP System Management Console.

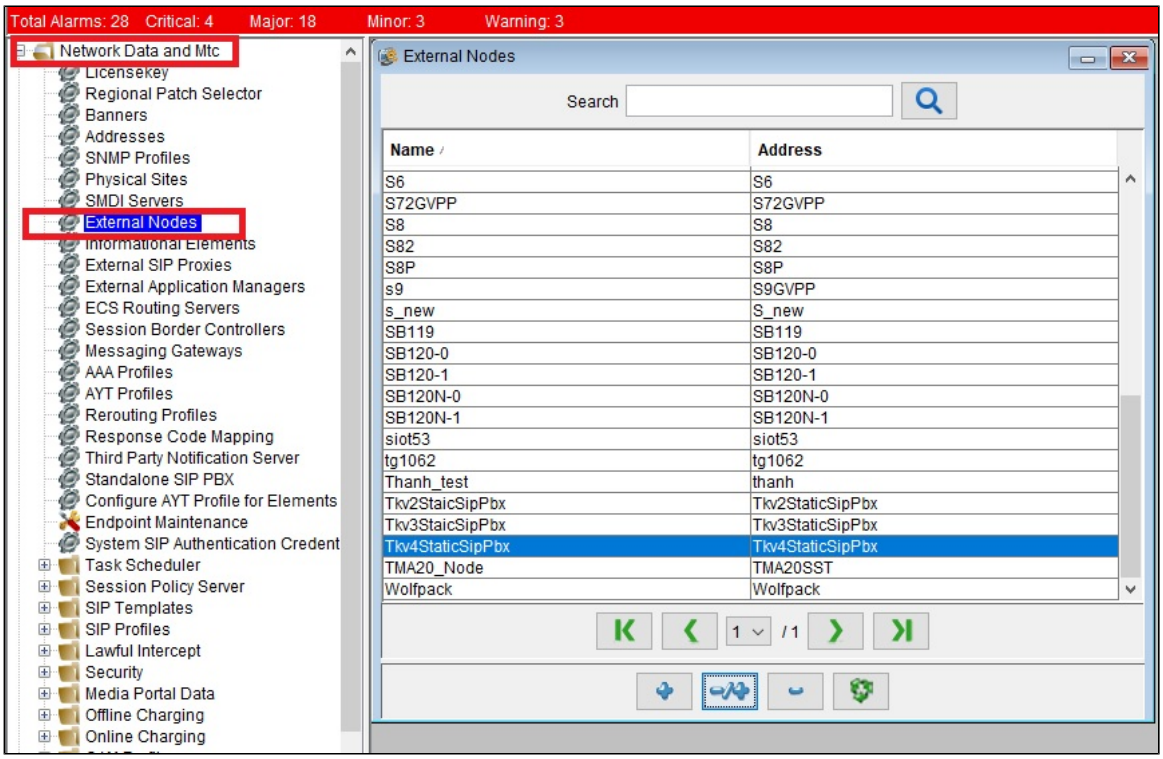
Static SIP PBX Addresses

The Static SIP PBX does not send SIP REGISTER messages to notify the network of its signaling address, therefore, the address of the SIP PBX must be provisioned in the network. Configure the address of the private side of the SBC that represents the static SIP PBX. Select **Network Data and Mtc > Addresses**. Add an address as shown in the following example.



Static SIP PBX External Nodes

Configure an external node by mapping the Static SIP PBX address to the node name. Select **Network Data and Mtc > External Nodes**. Add an External Node as shown in the example below.

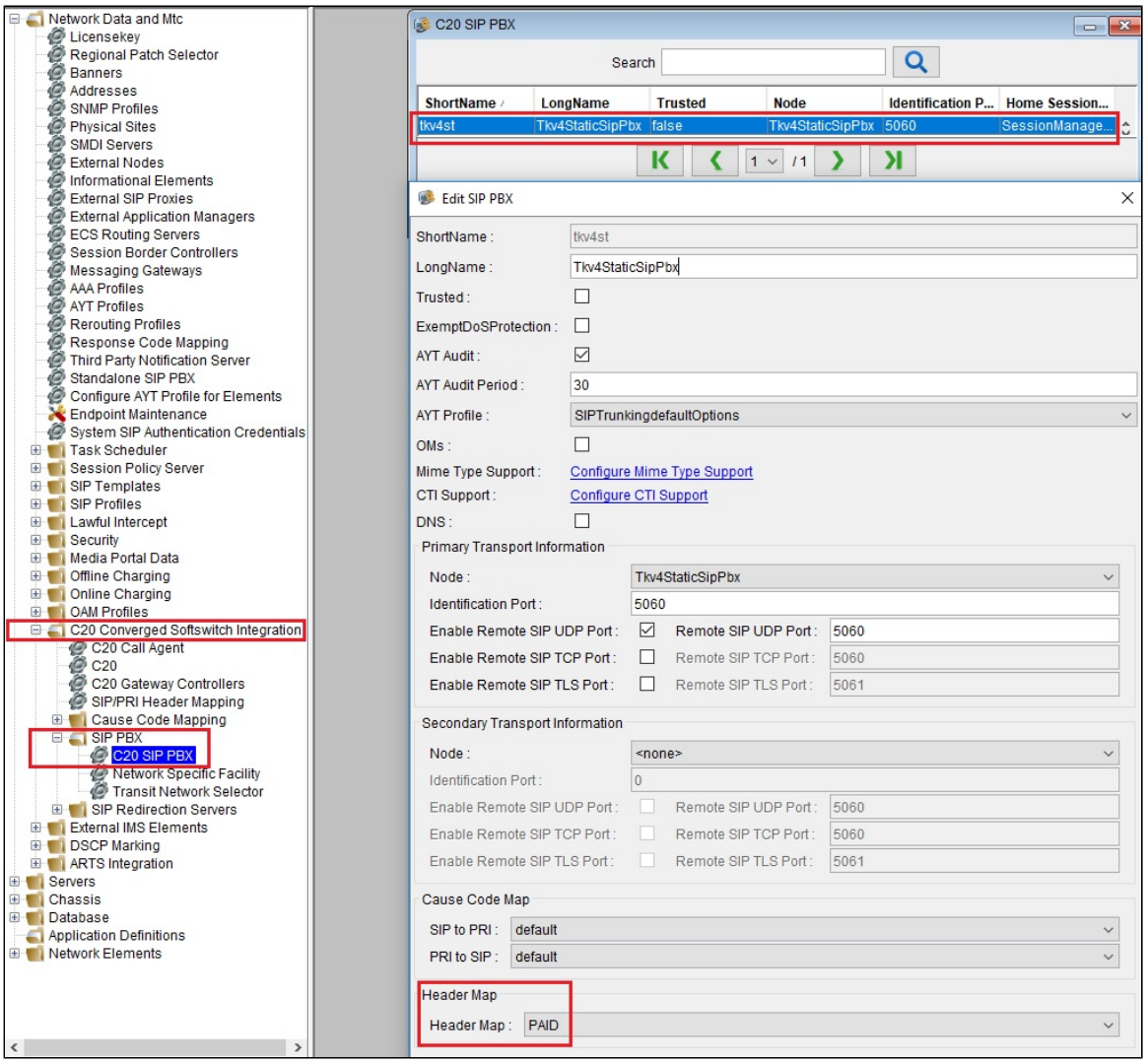


C20 SIP PBX

From **Network Data and Mtc**, select **C20 Converged Softswitch Integration**, **SIP PBX**, and then **C20 SIP PBX**.

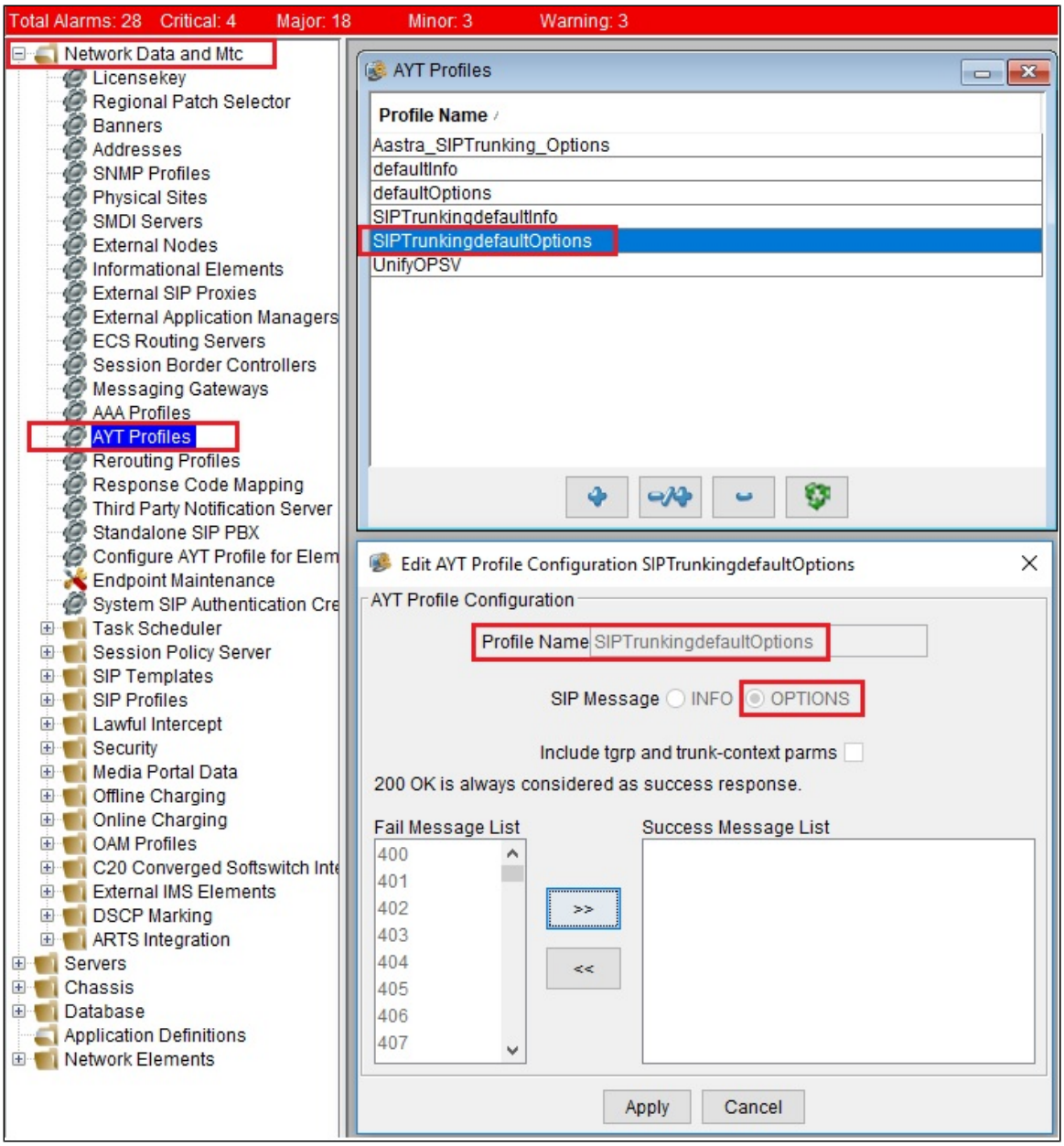
Add a C20 SIP PBX as referenced below using the C20 CIP PBX “tkv4st.”

> AS13.0 and AS12.1 loads support the "transport node" solution as shown in the snapshot below.
>
> For AS14.0 and higher loads, C20 SIPPBX uses the "transport link" solution.



AYT Profiles

The AYT Profile is mandatory if the AYT Audit was selected in the C20 SIP PBX setup as shown in the preceding screen capture. Here, we will either select SIP OPTIONS to send OPTIONS message or SIP INFO to send INFO messages (to probe SIP PBX availability) and also, we may select valid responses for the Success Message List apart from default 200 OK response to the AYT audit. The AYT audits (either SIP OPTIONS or SIP INFO messages) will be sent by the C20/AS to monitor the connectivity to the SIP PBX. From **Network Data and Mtc** select **AYT Profiles**.

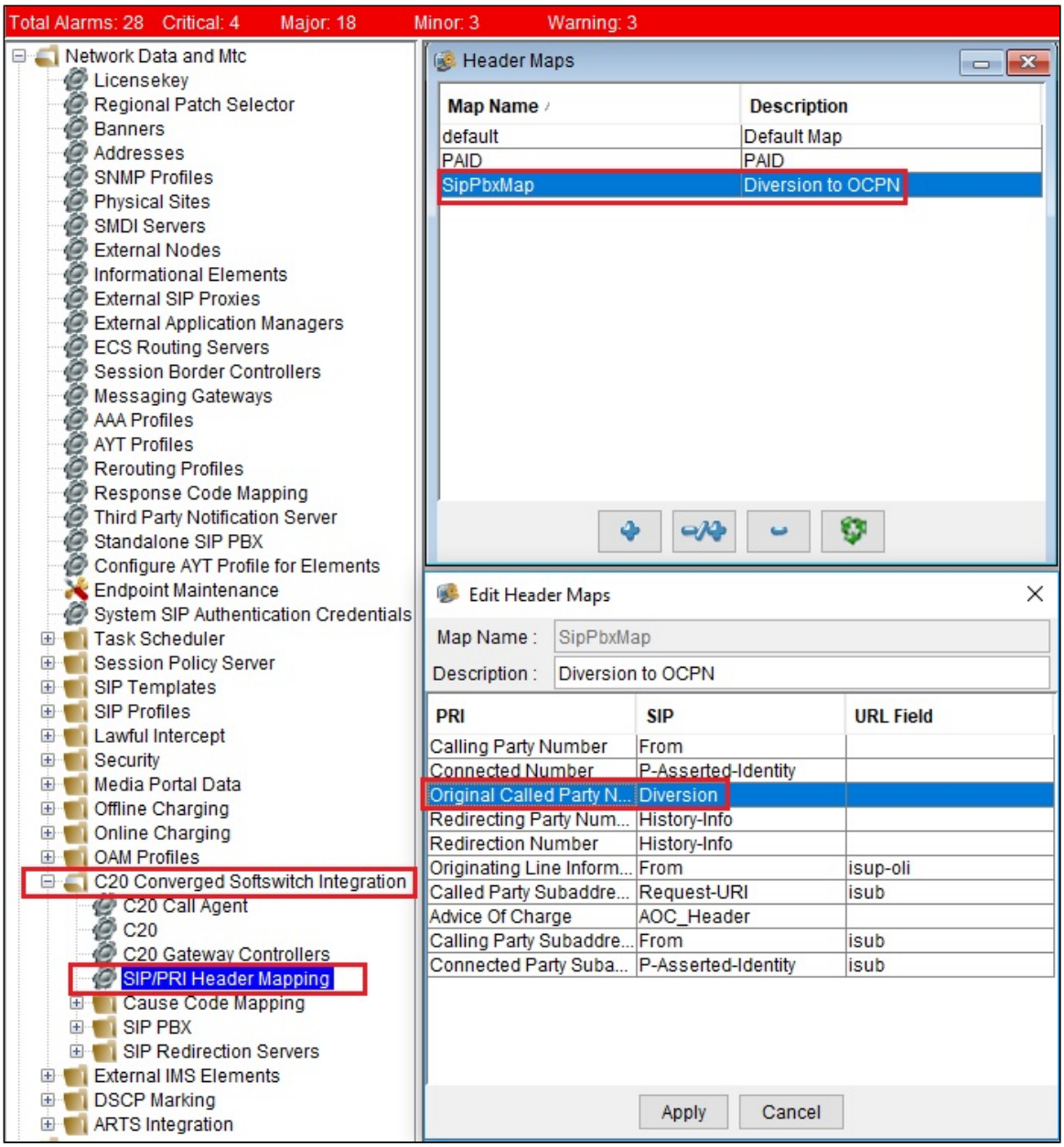


Note
SIPTrunkingdefaultOptions was used during testing.

SIP/PRI Header Mapping

A new Header Map was created for this test. This maps specific SIP messages to specific PRI messages. The default Header Map was modified (see the following screen capture).

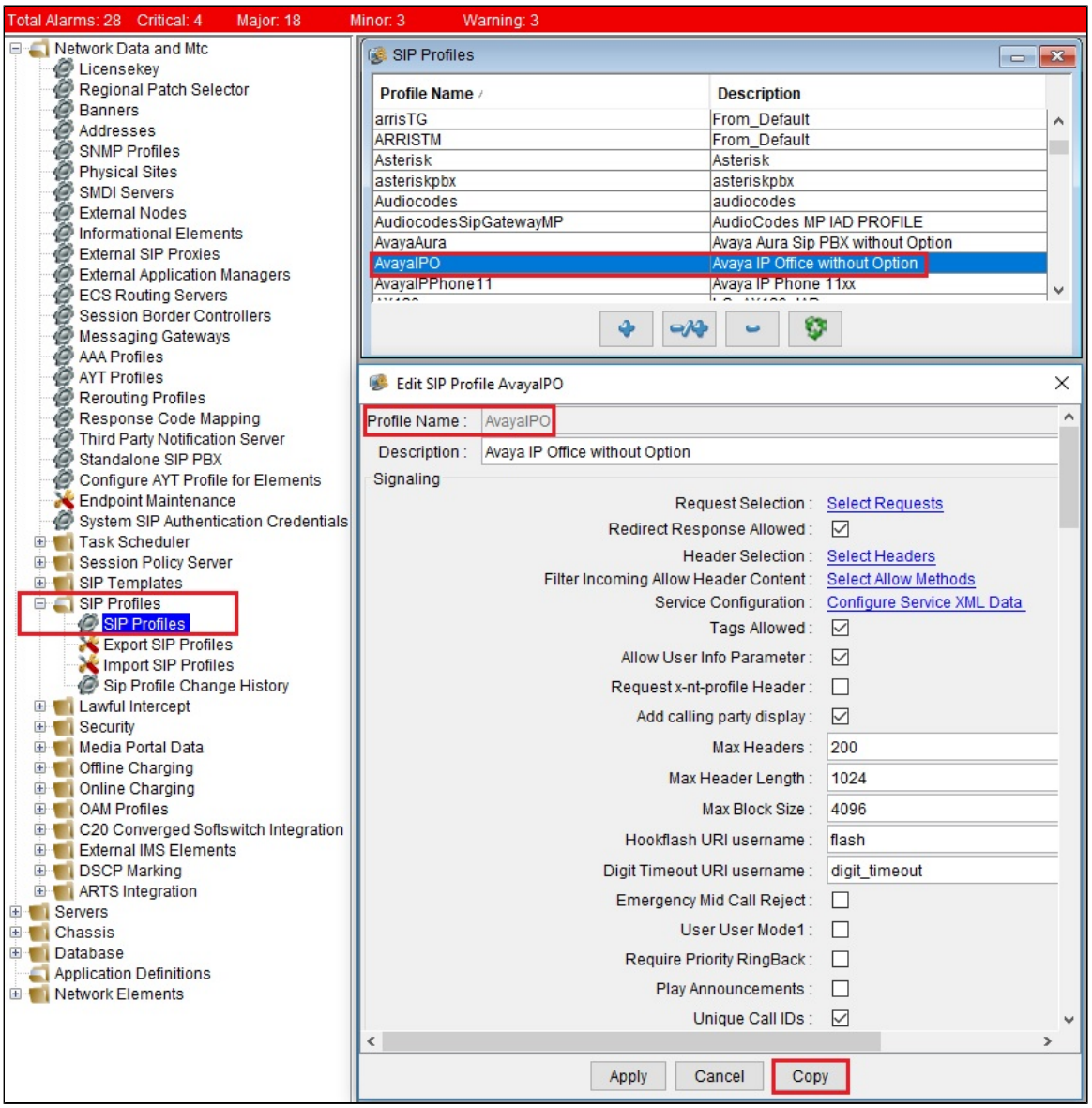
From **Network Data and Mtc** select the **C20 Converged Softswitch Integration** folder and then **SIP/PRI Header Mapping**.

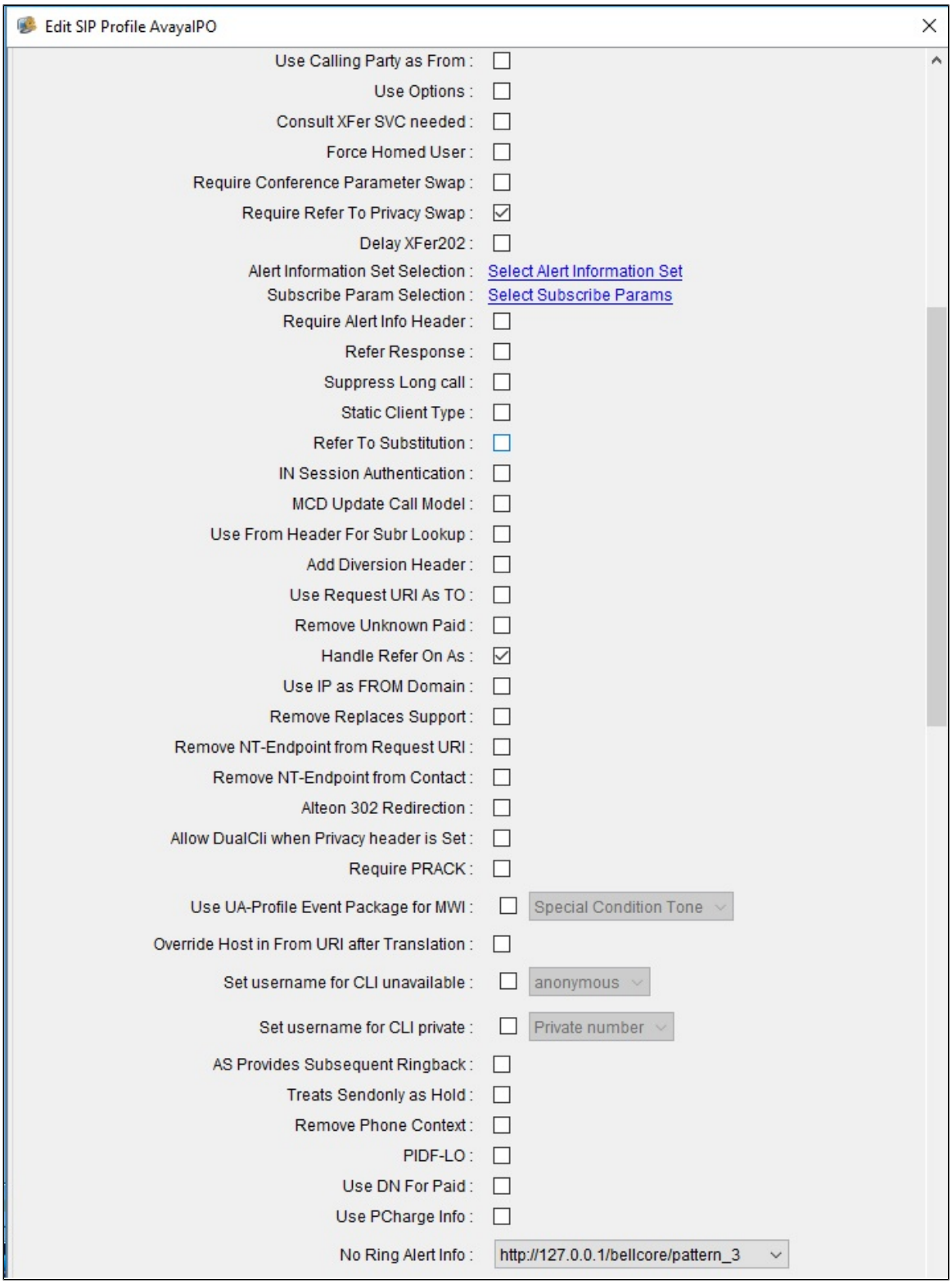


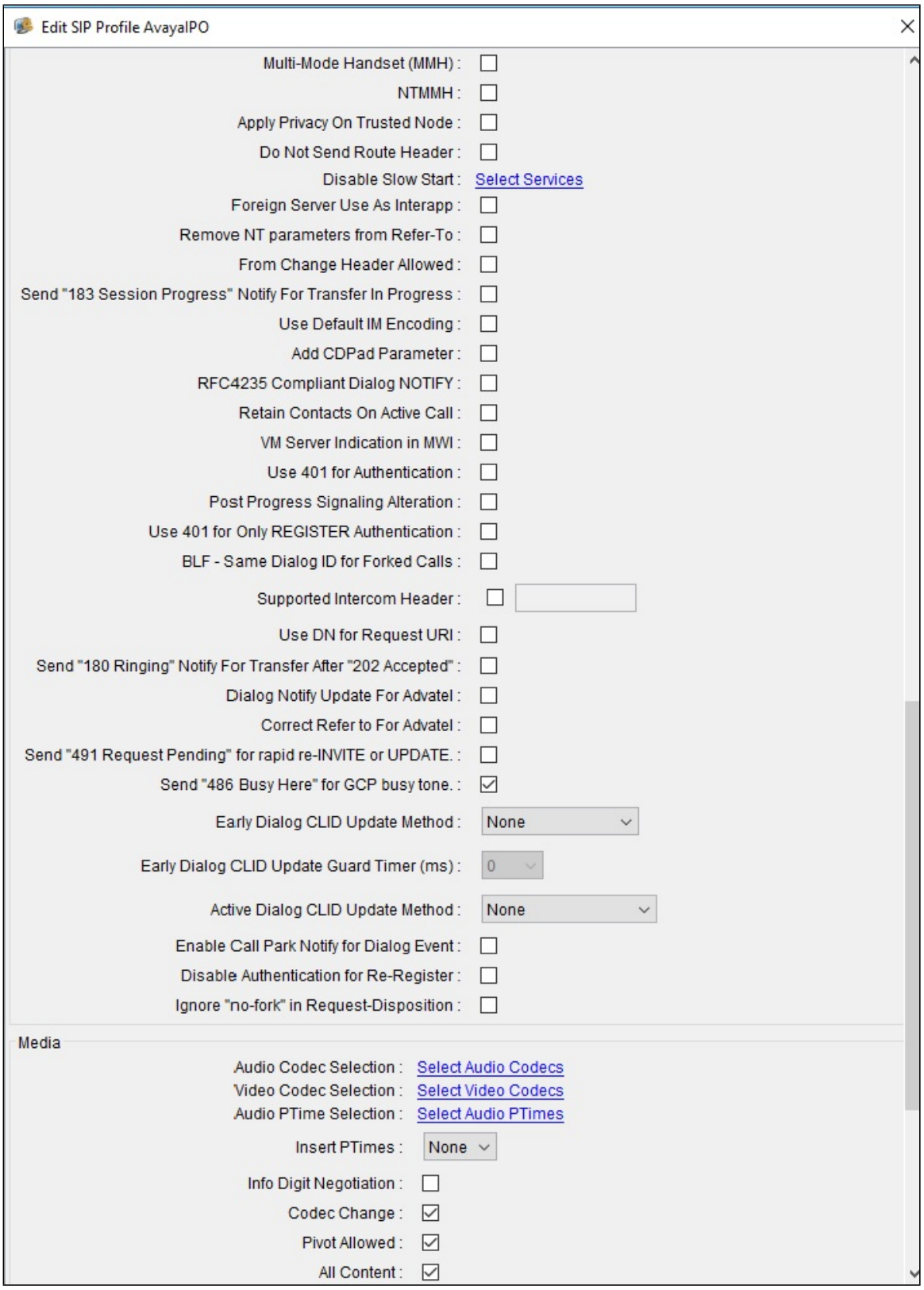
Note
The PRI header Original Called Party Number was changed from the History-Info header to the Diversion header. This instructs the AS to use the Diversion header containing the Original Called party Number. When the Call is sent to the CIM Voice Mail server, the Diversion header is in the INVITE message.

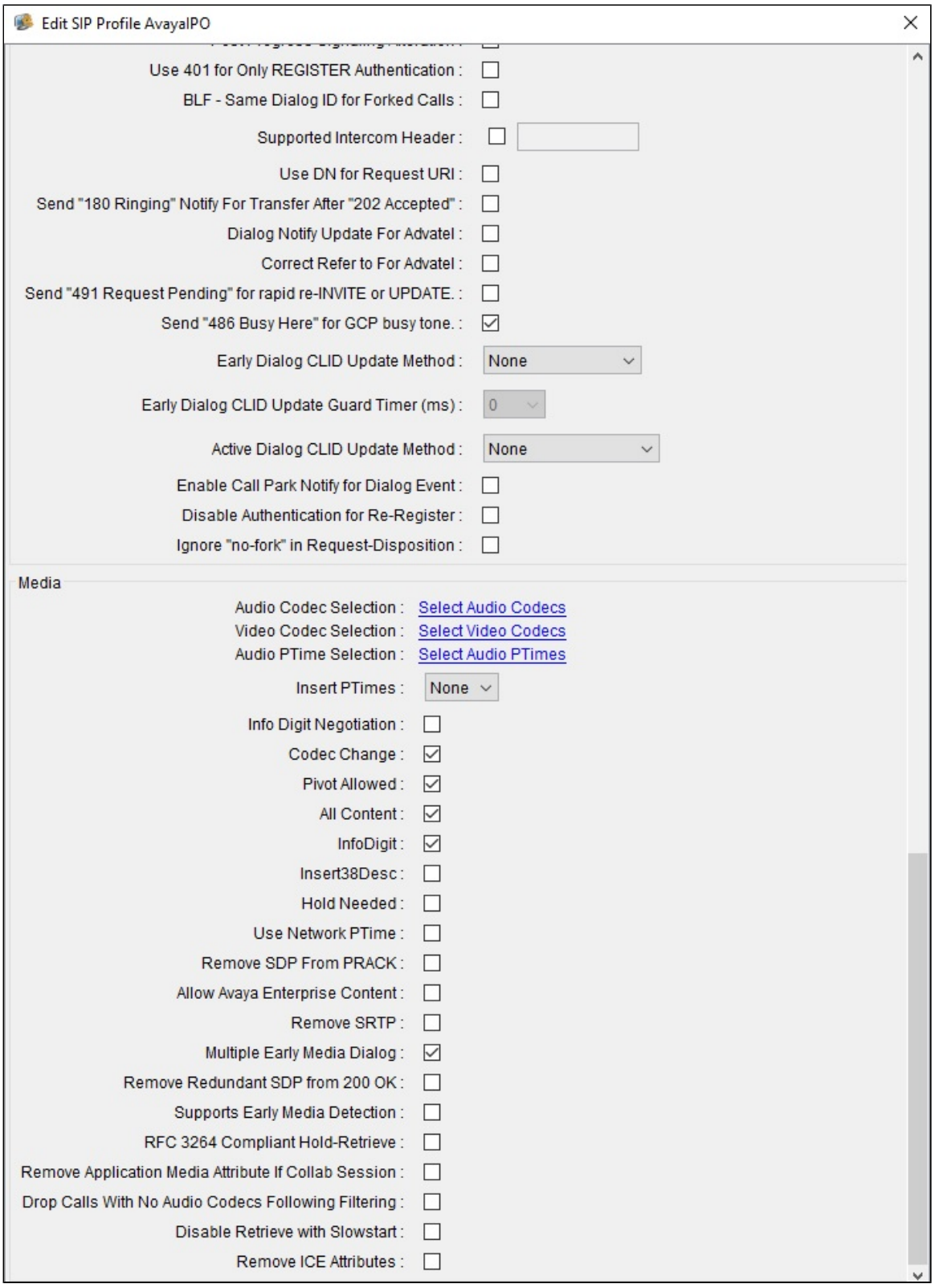
SIP Profiles

A custom SIP Profile may be required to support different types of SIP PBX. In this case, Ribbon recommends you copy the SipPBX SIP Profile and only change the settings required to support the particular SIP PBX. From **Network Data and Mtc**, select the **SIP Profiles** folder and then **SIP Profiles**. Then give the profile a name and description.



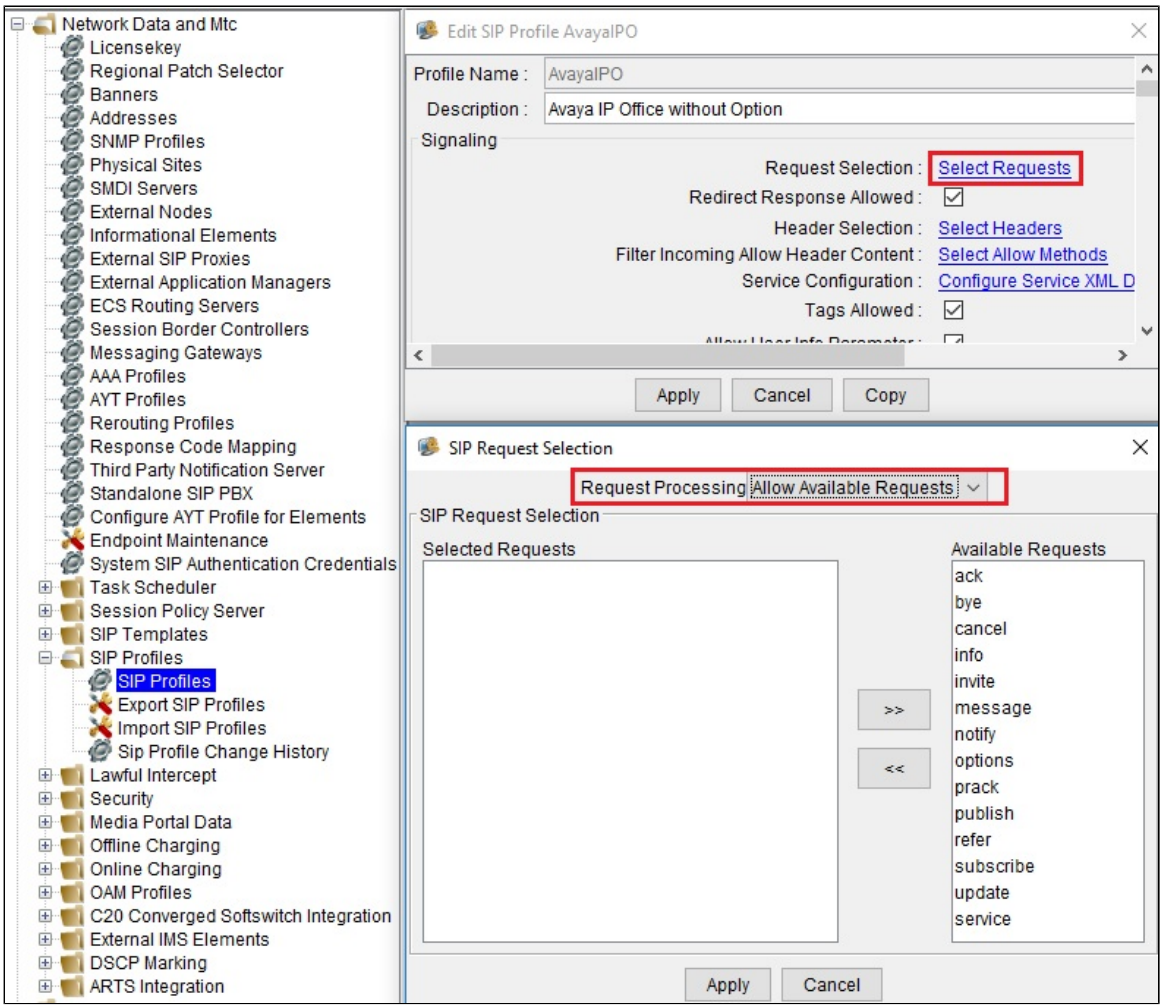






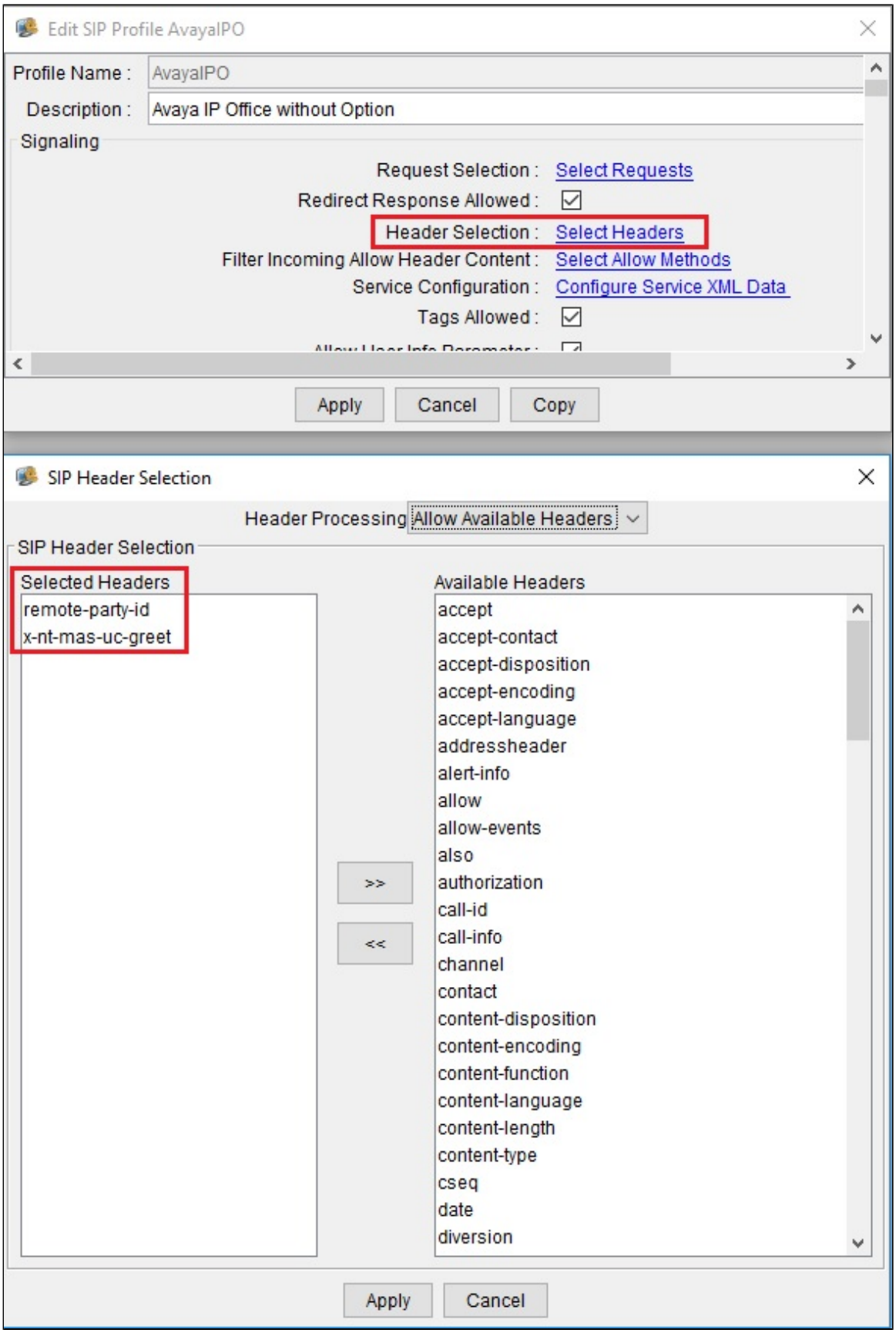
Select Requests

The following default settings are viewed when clicking the **Select Requests** link.



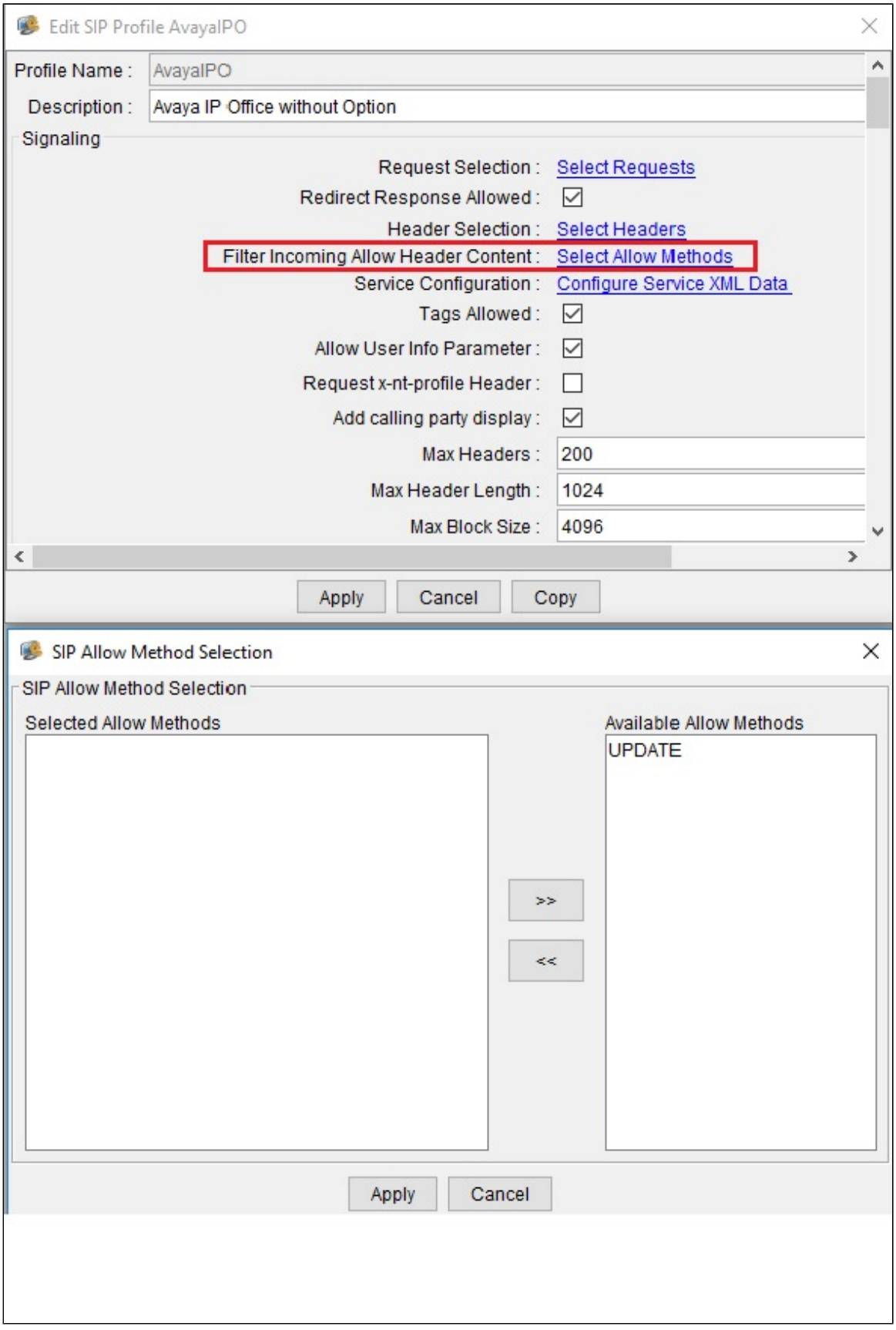
Select Headers

The following default settings are viewed when clicking the **Select Headers** link.



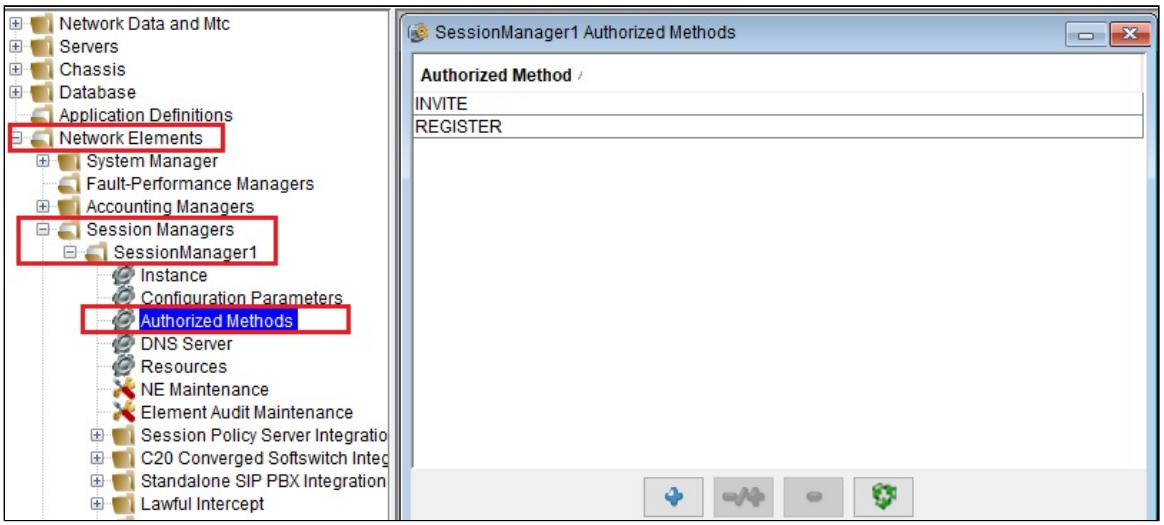
Select Allow Methods

The following default settings are viewed when clicking the **Select Allow Methods** link.



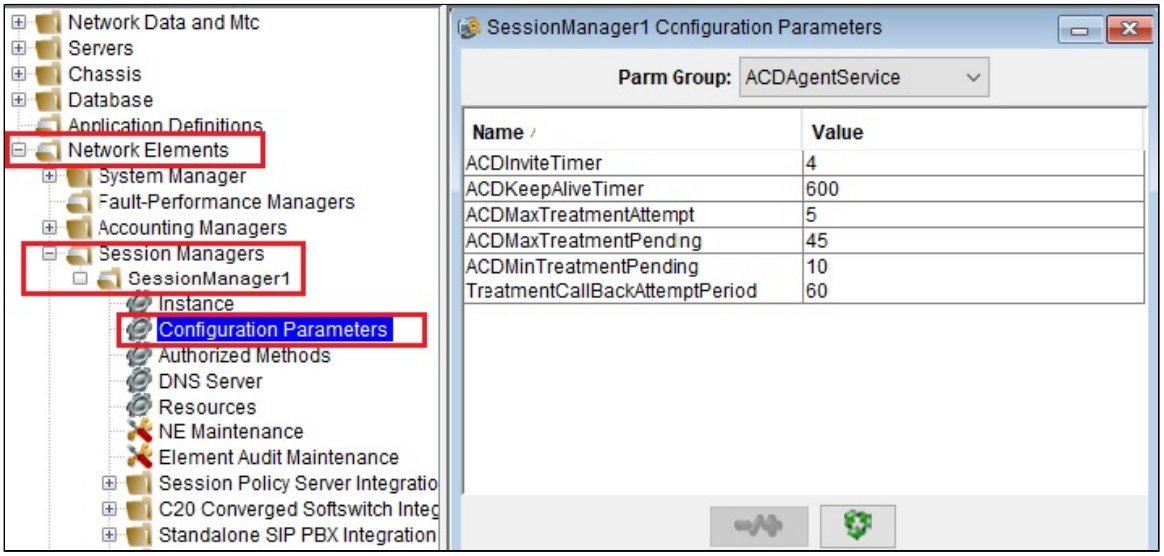
SIP Authorized Methods

Configure the AS to require authentication for both SIP INVITE and REGISTER transactions. From **Network Elements**, select **Session Managers**, **SessionManagerX**, and then **Authorized Methods**.



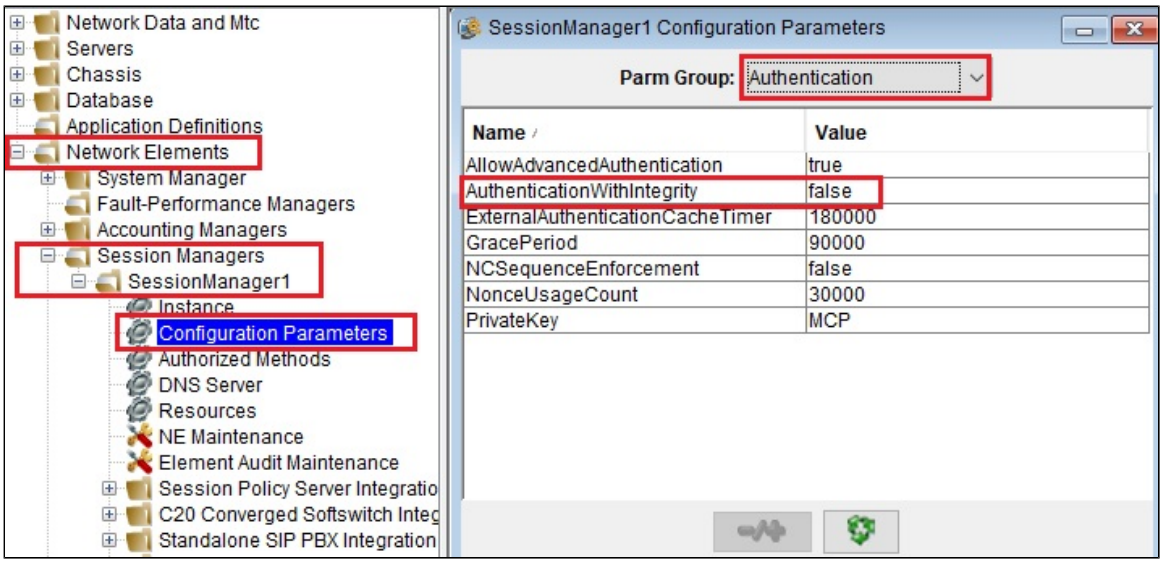
Session Manager Configuration Parameters

Session Manager (SESM) Configuration Parameters and associated values are set globally within the AS and are not unique to the SIP PBX. From **Network Elements**, select **Session Managers**, **Session Manager x (x can be from number 1)**, and then **Configuration Parameters**.



Authentication With Integrity

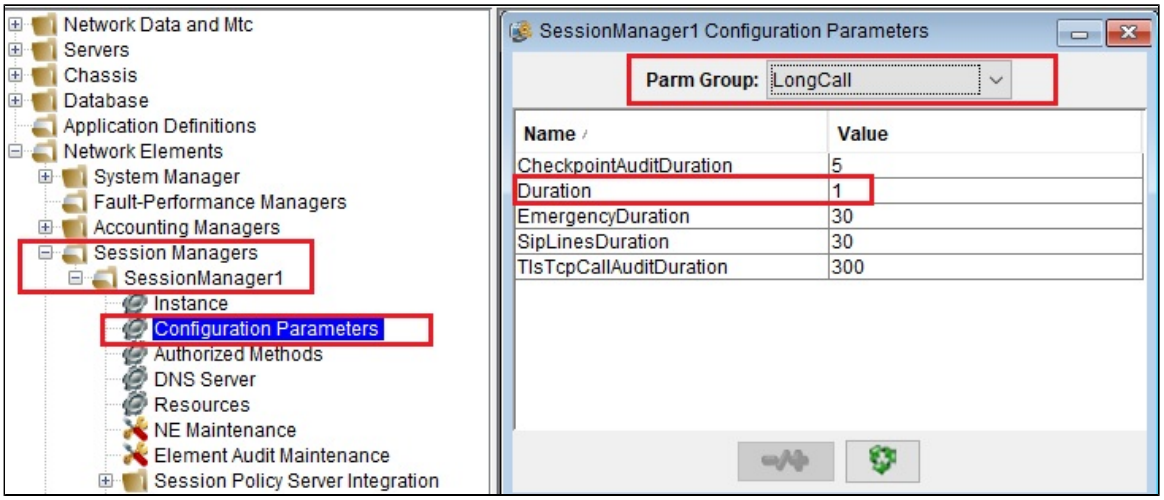
For each AS SESM, set the **AuthenticationWithIntegrity** parameter to **false** and Parm Group = **Authentication**.



Configuration Parameters for Long Call

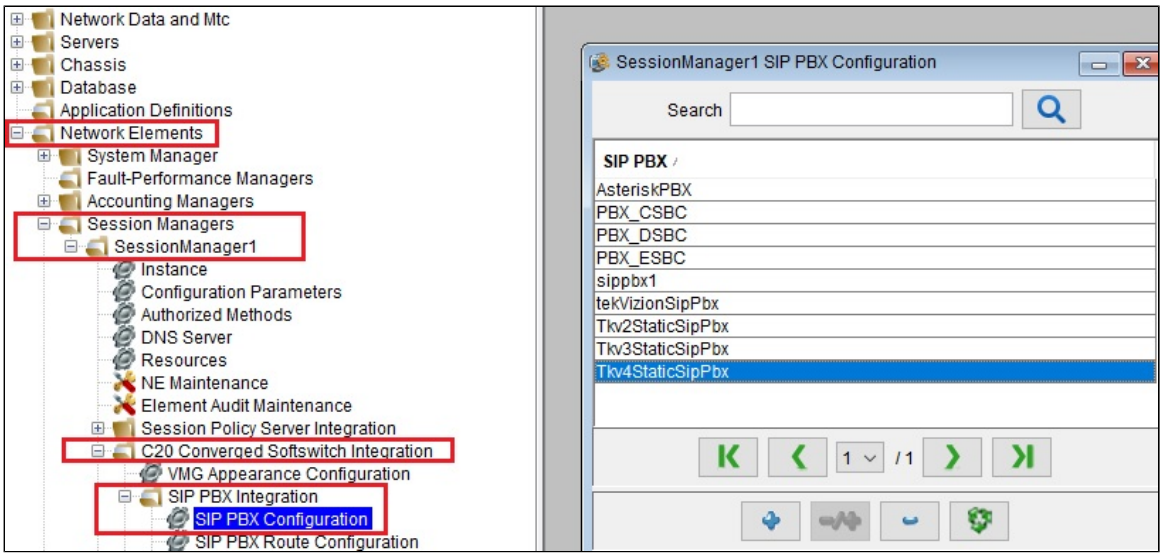
This parameter determines the length of time, in minutes, between endpoint audits. Duration is used to detect abandoned calls. A value of zero deactivates the duration parameter.

> **Note**
> Make sure Parm Group = LongCall.



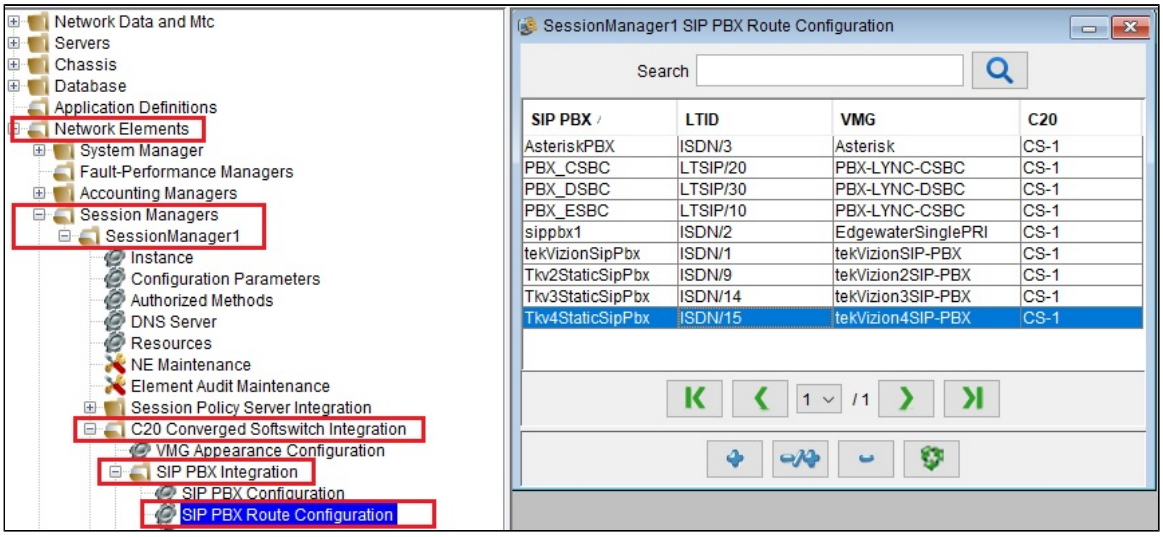
SIP PBX Configuration

Defines a name for the SIP PBX to use in other associated provisioning entities. From **Network Elements**, select **Session Managers > SessionManager1 > C20 Converged Softswitch Integration > SIP PBX Integration > SIP PBX Configuration**. The SIP PBX Configuration used for our setup was Tkv4StaticSipPbx.



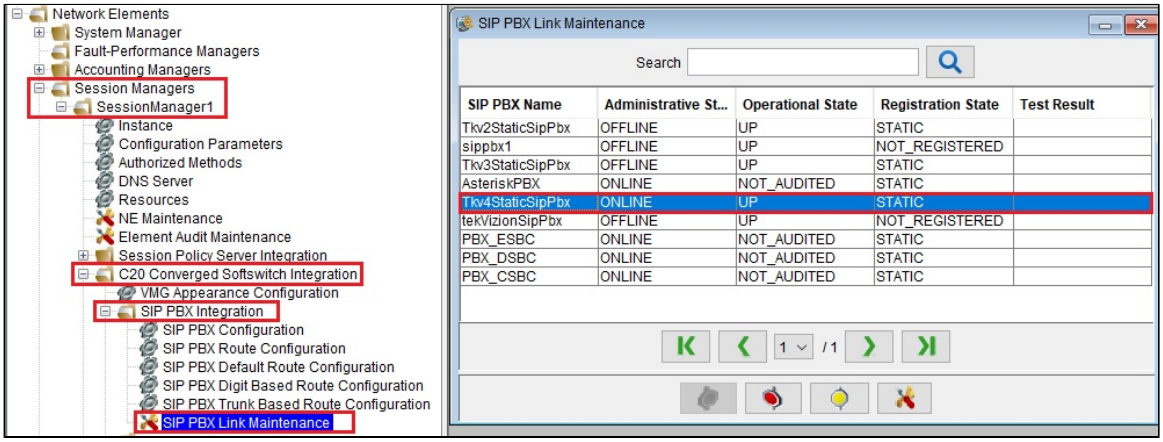
SIP PBX Route Configuration

Establishes links between the AS SIP PBX entity and the C20 by associating the SIP PBX name, defined in the AS SIP PBX Configuration, with an ISDN Logical Terminal ID (LTID) and Virtual Media Gateway defined in the C20.



SIP PBX Link Maintenance

After configuring the SIP PBX in the Provisioning Manager shown in the following screen capture, we will be able to bring up the link.



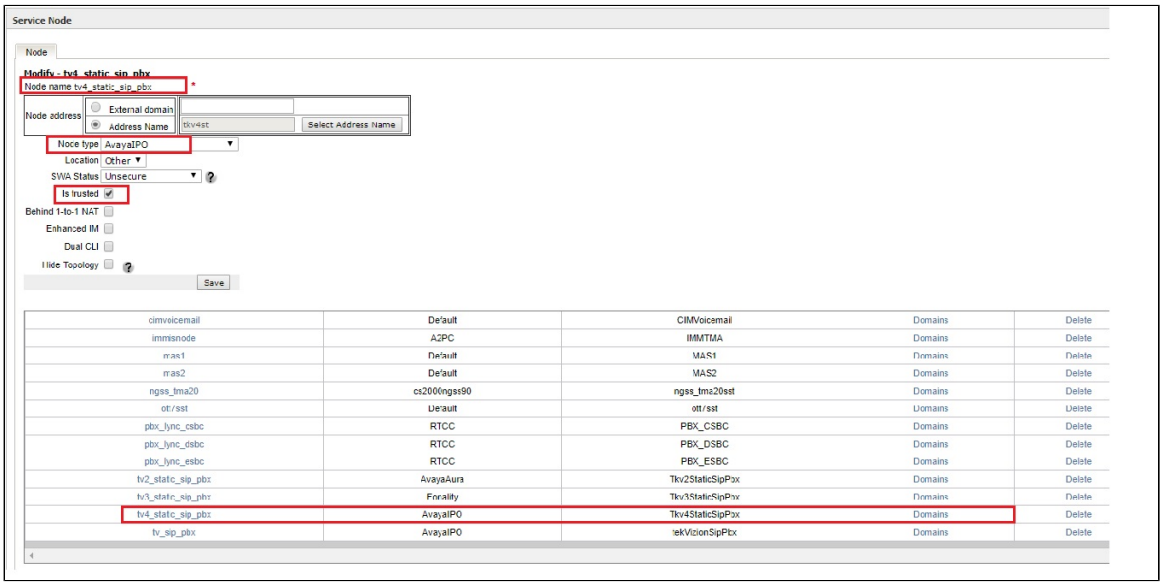
AS Provisioning Manager

The AS Provisioning Manager is the AS web-based interface which is accessible using http as shown in the example below.

Example: http://<provisioning client IP address>:8443/prov

Service Node for SIP PBX

Defines the SIP PBX Entity, established through the AS System Manager, as an AS Service Node. In the Translations tab, select **Service Node**.

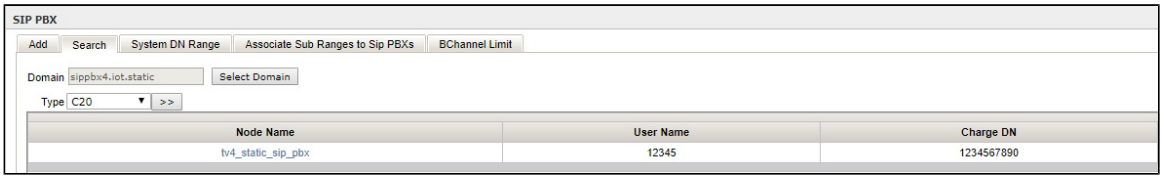


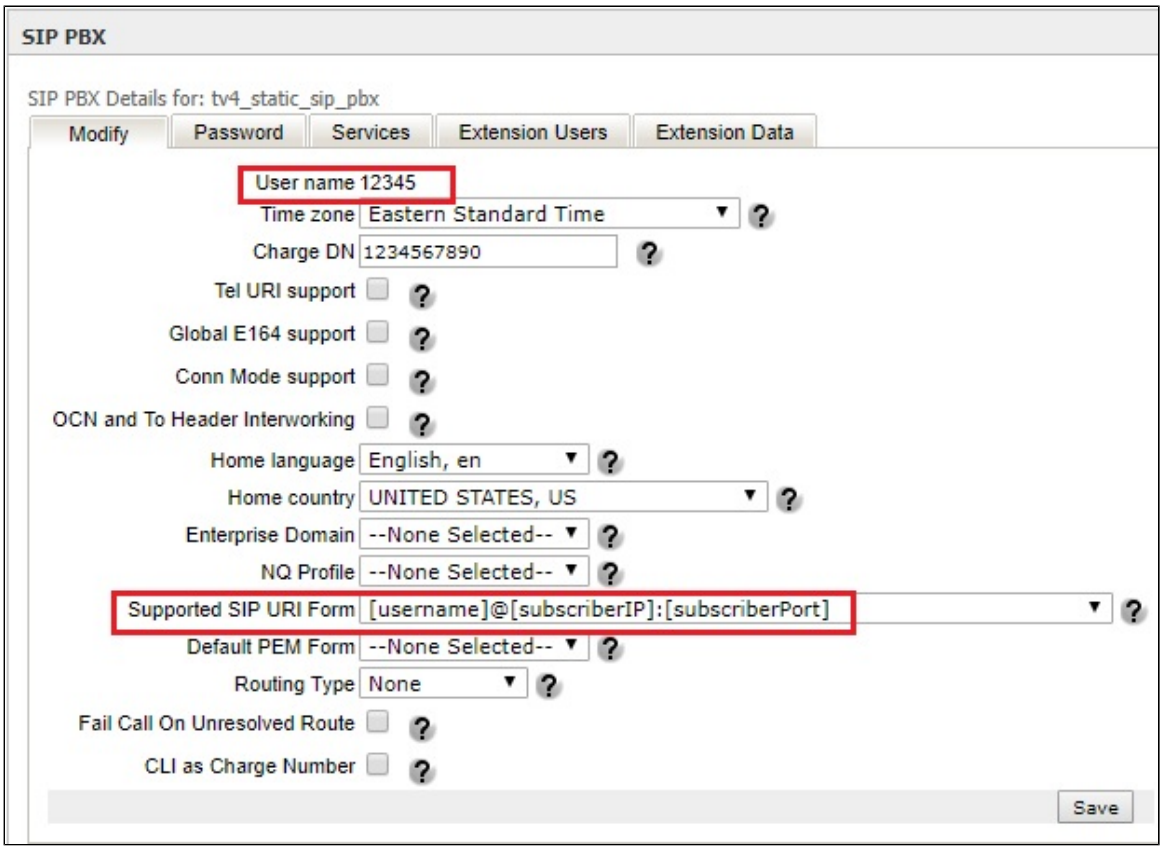
The following list defines the fields in the Service Node screen:

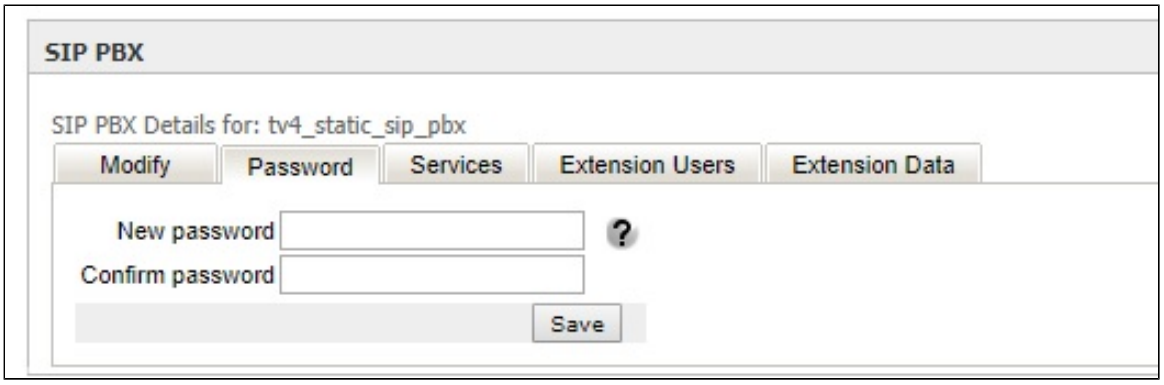
- **Node type:** Select a node type based on the SIP Profile that has been established for this SIP PBX.
- **Select Address Name:** Select the short name of the SIP PBX.
- **Location:** Choose **Other** unless another location is required for the SIP.
- PBX Location choices are limited to those available within the SIP PBX's assigned domain "Is trusted": When selected, the AS will send the P-Asserted-Identity header to the SIP PBX.
- For the Node name, assign the Domain to the Node by clicking the Blue Domain link.

SIP PBX

From the Solution tab, select **SIP PBX**, then create a SIP PBX. Specify a username and password for the SIPPBX that will be used to authenticate associated transactions specified in the Authorized Methods.The following is a SIP PBX example.







Configure Ribbon C20

Gateway

Add a Gateway of type VOIP_VPN associated with the SIP PBX Entity (in this case tekVizion4SIP-PBX). The IP address must be the service address of the AS SESM on which the SIP PBX is configured.

Gateway List

Name	Domain	IP Address	MGC Sec...	GW ...	Profile	Max Terms	Res Terms	Protocol
Asterisk		172.28.243.70			VOIP_VPN	2047	24	gcp
EdgewaterSinglePRI		172.28.243.70			VOIP_VPN	2047	24	gcp
PBX-LYNC-CSBC		172.28.243.70			VOIP_VPN	2047	24	gcp
PBX-LYNC-DSBC		172.28.243.70			VOIP_VPN	2047	24	gcp
PBX-LYNC-ESBC		172.28.243.70			VOIP_VPN	2047	24	gcp
tekVizion2SIP-PBX		172.28.243.70			VOIP_VPN	2047	24	gcp
tekVizion3SIP-PBX		172.28.243.70			VOIP_VPN	2047	24	gcp
tekVizion4SIP-PBX		172.28.243.70			VOIP_VPN	2047	24	gcp
tekVizionSIP-PBX		172.28.243.70			VOIP_VPN	2047	24	gcp
testpbx		172.28.243.70			VOIP_VPN	2047	5	gcp

Carriers

Define Carrier(s) to assign the ISDN PRI trunks in the C20 associated with the SIP PBX Entity. In this example, carrier EPG_002 for (SIP PBX) gateway tekVizion4SIP-PBX has 23 channels assigned starting at 2. These will be used as bearer channels. Carrier EPG_001 for tekVizion4SIP-PBX has a single channel (1) that will serve as the ISDN PRI “D channel” for signaling.

> **Note**
> The node number is (92).

Carrier List

Name	Gateway	GW Domain	Node Num	Start Term	Num Ports	NFAS/D...	IUA ID
EPG_002	Asterisk		92	2	23	0	
EPG_001	Asterisk		92	1	1	0	
EPG_002	EdgewaterSinglePRI		92	26	23	0	
EPG_001	EdgewaterSinglePRI		92	25	1	0	
EPG_002	tekVizionSIP-PBX		92	50	23	0	
EPG_001	tekVizionSIP-PBX		92	49	1	0	
EPG_002	tekVizion2SIP-PBX		92	74	23	0	
EPG_001	tekVizion2SIP-PBX		92	73	1	0	
EPG_002	tekVizion3SIP PBX		92	98	23	0	
EPG_001	tekVizion3SIP-PBX		92	97	1	0	
EPG_002	tekVizion4SIP-PBX		92	122	23	0	
EPG_001	tekVizion4SIP-PBX		92	121	1	0	
EPG_001	testpbx		92	300	1	0	
EPG_001	PBX-LYNC-ESBC		92	500	10	0	
EPG_001	PBX-LYNC-CSBC		92	600	10	0	
EPG_001	PBX-LYNC-DSBC		92	650	10	0	

C20 Call Agent

Table CLLI

Define the Trunk Group name and size.

```
TABLE: CLLI
>list
CLLI ADNUM TRKGRSIZ ADMININF
---------------------------
TEKVIZION_4 241 48 TEKVIZION_4_SIP_PBX_IOT
```

Table TRKGRP

Define an ISDN PRI trunk group supporting the SIP PBX. GRPKEY is derived from table CLLI. GRPTYP = PRA is required for ISDN PRI trunk group. LTID is derived from table LTDEF and cannot be datafilled manually. Therefore, enter $ for this field when initially adding the SIP PBX trunk group.

```
TABLE: TRKGRP
>list
GRPKEY GRPTYP TRAFSNO PADGRP NCCLS GRPINFO
-----------------------------------------
TEKVIZION_4 PRA 0 NPDGP NCRT MIDL N (ISDN 15) $ $
```

Table TRKSGRP

Define a trunk subgroup entry for the SIP PBX ISDN PRI trunk group. SGRPKEY value TEKVIZION is derived from table CLLI; 0 means it is the 0th trunk subgroup for TEKVIZION. SGRPVAR ISDN is required to indicate ISDN PRI. PMTYPE, GWCNO, GWCNODENO, and GWCTRMNO are all values based on the SIP PBX gateway defined by the Gateway Controller provisioning.

> **Note**
> The value for GWCTRMNO (92), defined in Gateway Controller Carrier provisioning, is the D channel for the SIP PBX ISDN PRI trunk group.

```
TABLE: TRKSGRP
>list
SGRPKEY CARDCODE SGRPVAR SGRPVAR
------------------------------
TEKVIZION_4 0 DS1SIG ISDN 20 20 87Q931 2 N STAND NETWORK PT_PT USER N UNEQ
160 N DEFAULT GWC 10 92 121 64K HDLC $ $
```

Table TRKMEM

Define trunk members for the SIP PBX trunk group using the bearer path carrier channels (total 23) defined at the Gateway Controller.

```
TABLE: TRKMEM
>
>list 30
CLLI EXTRKNM SGRP MEMVAR
------------------------
TEKVIZION_4 0 0 GWC 10 92 122
TEKVIZION_4 1 0 GWC 10 92 123
TEKVIZION_4 2 0 GWC 10 92 124
TEKVIZION_4 3 0 GWC 10 92 125
TEKVIZION_4 4 0 GWC 10 92 126
TEKVIZION_4 5 0 GWC 10 92 127
TEKVIZION_4 6 0 GWC 10 92 128
TEKVIZION_4 7 0 GWC 10 92 129
TEKVIZION_4 8 0 GWC 10 92 130
TEKVIZION_4 9 0 GWC 10 92 131
TEKVIZION_4 10 0 GWC 10 92 132
TEKVIZION_4 11 0 GWC 10 92 133
TEKVIZION_4 12 0 GWC 10 92 134
TEKVIZION_4 13 0 GWC 10 92 135
TEKVIZION_4 14 0 GWC 10 92 136
TEKVIZION_4 15 0 GWC 10 92 137
TEKVIZION_4 16 0 GWC 10 92 138
TEKVIZION_4 17 0 GWC 10 92 139
TEKVIZION_4 18 0 GWC 10 92 140
TEKVIZION_4 19 0 GWC 10 92 141
TEKVIZION_4 20 0 GWC 10 92 142
TEKVIZION_4 21 0 GWC 10 92 143
TEKVIZION_4 22 0 GWC 10 92 144
```

Table LTDEF

Define the Logical Terminal ID ISDN 15 for use with the SIP PBX trunk group.

```
TABLE: LTDEF
>
>pos ISDN 15
ISDN 15 B PRA 23 NTNAPRI V1 NIL (NOPMD ) $
```

Table LTMAP

Map the Logical Terminal ID, defined in table LTDEF, to the SIP PBX trunk group.

(TEKVIZION_4).

- LTKEY is the LTID defined in table LTDEF.
- CLLI is the CLLI of the SIP PBX trunk group.
- All remaining values are DEFAULT.

```
>table ltmap
TABLE: LTMAP
LTKEY MAPTYPE OPTION
--------------------
ISDN 15 CLLI TEKVIZION_4 (TEI 0) $
```

Table LTCALLS

Define translation paths for LTID ISDN 15. Public routes use XLARTE (translation route selector) XLALEC.

```
>table ltcalls
TABLE: LTCALLS
ISDN 15 PUB XLALEC 80 IOTPLAN IOTAREA $
```

Table LTDATA

Define service data associated with LTID ISDN 15. OPTION PRI_IP_PROT is used to define the IP protocol for ISDN 14 as SIP (required for SIP PBX trunks).

```
>table ltdata
ISDN 15 SERV SERV Y Y ALWAYS ALWAYS (NET_RINGBACK_ON ) (PRI_IP_PROT SIP )
(EMCT ) $
```

Table MSGRTE

If using centralized C20 voice mail, table MSGRTE routes message-waiting indications back to the mailbox subscriber. In this example, any MWI for DNs in this range are sent to the PRA trunk group TEKVIZION_4 (LTID ISDN 15).

```
>table msgrte
TABLE: MSGRTE
PUBLIC 211 212
(PRA TEKVIZION_4 0 N ( LTID ISDN 15) $)$
```

Routing

PSTN to Asterisk PBX call

An example of the translation flow to route calls from PSTN line to SIP PBX trunk members is shown in the TRAVER outputs below.

```
>traver l 9920055 9192210250 b
TABLE IBNLINES
SL5 00 0 05 55 0 DT STN IBN 9920055 IOTLAB 0 0 919 $
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
TABLE IBNFEAT
TUPLE NOT FOUND
TABLE CUSTSTN
IOTLAB AIN AIN TIID
TABLE OFCVAR
AIN_OFFICE_TRIGGRP TIID
AIN Orig Attempt TDP: no subscribed trigger.
TABLE NCOS
IOTLAB 0 0 0 IOT0 ( XLAS RESXLA FEATXLA RESDC) ( OCTXLA FEATXLA)$
TABLE CUSTHEAD: CUSTGRP, PRELIMXLA, CUSTXLA, FEATXLA, VACTRMT, AND DIGCOL
IOTLAB NXLA RESXLA FEATXLA 0 RESDC
TABLE DIGCOL
RESDC 9 RES
TABLE IBNXLA: XLANAME RESXLA
TUPLE NOT FOUND
Default from table XLANAME:
RESXLA
    (NET N N 0 N NDGT N N GEN ( LATTR 80 IOTPLAN IOTAREA) $ $)$ 9
TABLE DIGCOL
NDGT specified: digits collected individually
TABLE LINEATTR
80 IBN NONE NT 0 0 NILSFC 0 NIL NIL 00 613_PKDK_80 L613_NILLA_0 $
LCABILL OFF - BILLING DONE ON BASIS OF CALLTYPE
TABLE XLAPLAN
IOTPLAN NSCR 919 IOTL RTE1 Y IOTLAB 0 0 $ $
TABLE RATEAREA
IOTAREA NLCA NIL LATA1 IOTLAB
TABLE STDPRTCT
IOTL ( 1) ( 1) 4
 . SUBTABLE STDPRT
 . KEY NOT FOUND
 . DEFAULT VALUE IS:   N NP 0 NA
 . SUBTABLE AMAPRT
 . 9 9 NONE OVRDALL N
 . REGULAR TOLL BILLING WILL BE SUPRESSED BUT
 . LOCAL CALLS WILL CREATE TOLL BILLING RECORDS
TABLE HPCPATTN
TUPLE NOT FOUND
TABLE HNPACONT
919 Y 999 500 ( 64) ( 1) ( 0) ( 0) ( 0) 3 $
 . SUBTABLE HNPACODE
 . 919 919 HNPA 0
 . 221 221 DN 919 221
TABLE TOFCNAME
919 221 $
TABLE DNINV
919 221 0250 T OFR4 750
TABLE DNFEAT
TUPLE NOT FOUND
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
 . TABLE OFR4
 .  750 N D TEKVIZION_4 7 $ N
 . EXIT TABLE OFR4
```

```
LNP Info: Called DN is resident.
LNP Info: Called DN has native NPANXX.
LNP Info: HNPA results are used.
AIN Info Collected TDP: no subscribed trigger.
AIN Info Analyzed TDP: no subscribed trigger.
AIN Term Attempt TDP: no subscribed trigger.

+++ TRAVER: SUCCESSFUL CALL TRACE +++


DIGIT TRANSLATION ROUTES

1 TEKVIZION_4                N CDN  E164  L  250 NIL_NSF   BC SPEECH
Terminating LTID is ISDN   15

TREATMENT ROUTES.  TREATMENT IS: GNCT
1 T120

+++ TRAVER: SUCCESSFUL CALL TRACE +++
```

Asterisk PBX to PSTN Call

An example of the translation flow to route calls from SIP PBX trunk members to PSTN line is shown in the TRAVER outputs below.

```
>traver tr TEKVIZION_4 9199920055
>b
TABLE TRKGRP
TEKVIZION_4 PRA 0 NPDGP NCRT MIDL N (ISDN 15) $ $
TABLE LTCALLS
ISDN 15 PUB XLALEC 80 IOTPLAN IOTAREA $
TABLE CUSTSTN
TUPLE NOT FOUND
TABLE OFCVAR
AIN_OFFICE_TRIGGRP TIID
TABLE LINEATTR
80 IBN NONE NT 0 0 NILSFC 0 NIL NIL 00 613_PKDK_80 L613_NILLA_0 $
LCABILL OFF - BILLING DONE ON BASIS OF CALLTYPE
TABLE XLAPLAN
IOTPLAN NSCR 919 IOTL RTE1 Y IOTLAB 0 0 $ $
TABLE RATEAREA
IOTAREA NLCA NIL LATA1 IOTLAB
TABLE STDPRTCT
IOTL ( 1) ( 1) 4
 . SUBTABLE STDPRT
WARNING: CHANGES IN TABLE STDPRT MAY ALTER OFFICE
BILLING.  CALL TYPE DEFAULT IS NP.  PLEASE REFER TO
DOCUMENTATION.
 . KEY NOT FOUND
 . DEFAULT VALUE IS:   N NP 0 NA
 . SUBTABLE AMAPRT
 . 9 9 NONE OVRDALL N
 . REGULAR TOLL BILLING WILL BE SUPRESSED BUT
 . LOCAL CALLS WILL CREATE TOLL BILLING RECORDS
TABLE HPCPATTN
TUPLE NOT FOUND
TABLE HNPACONT
919 Y 999 500 ( 64) ( 1) ( 0) ( 0) ( 0) 3 $
 . SUBTABLE HNPACODE
 . 919 919 HNPA 0
 . 992 992 DN 919 992
TABLE TOFCNAME
919 992 $
TABLE DNINV
919 992 0055 ILC SL5 00 0 05 55
TABLE DNFEAT
TUPLE NOT FOUND
TABLE DNATTRS
TUPLE NOT FOUND
TABLE DNGRPS
TUPLE NOT FOUND
LNP Info: Called DN is resident.
LNP Info: Called DN has native NPANXX.
LNP Info: HNPA results are used.
AIN Info Collected TDP: no subscribed trigger.
AIN Info Analyzed TDP: no subscribed trigger.
AIN Term Attempt TDP: no subscribed trigger.

+++ TRAVER: SUCCESSFUL CALL TRACE +++


DIGIT TRANSLATION ROUTES

1 LINE                  9199920055                    ST
```

```
TREATMENT ROUTES.  TREATMENT IS: GNCT
1 GNCTANN
```

Section-D : Asterisk PBX Configuration

This section provides the procedure for configuring the Asterisk to support connectivity to the Ribbon C20-AS SIP Trunking solution through the SBC. This section requires you to have the knowledge of using, configuring, and supporting the Asterisk and experience of working with the product platform. The following sections show you how to configure the Asterisk. You use the config files with root login credentials to configure the Asterisk.

Accessing Asterisk

Log in to the Asterisk ssh session with root credentials, and access the /etc/asterisk/ folder.

Asterisk User & Peer Configuration

For configuring Asterisk PBX users & outbound / inbound Peers, refer to the sip.conf file and configure per specific requirement.

```
vi /etc/asterisk/sip.conf

; SIP Configuration example for Asterisk
;
; Note: Please read the security documentation for Asterisk in order to
;         understand the risks of installing Asterisk with the sample
;        configuration. If your Asterisk is installed on a public
;        IP address connected to the Internet, you will want to learn
;        about the various security settings BEFORE you start
;        Asterisk.
;
;        Especially note the following settings:
;                - allowguest (default enabled)
;                - permit/deny - IP address filters
;                - contactpermit/contactdeny - IP address filters for registrations
;                - context - Which set of services you offer various users
;
; SIP dial strings
;------------------------------------------------------------
; In the dialplan (extensions.conf) you can use several
; syntaxes for dialing SIP devices.
;        SIP/devicename
;        SIP/username@domain   (SIP uri)
;        SIP/username[:password[:md5secret[:authname[:transport]]]]@host[:port]
;        SIP/devicename/extension
;        SIP/devicename/extension/IPorHost
;        SIP/username@domain//IPorHost
;
;
; Devicename
;        devicename is defined as a peer in a section below.
;
; username@domain
;        Call any SIP user on the Internet
;        (Don't forget to enable DNS SRV records if you want to use this)
;
; devicename/extension
;        If you define a SIP proxy as a peer below, you may call
;        SIP/proxyhostname/user or SIP/user@proxyhostname
;        where the proxyhostname is defined in a section below
;        This syntax also works with ATA's with FXO ports
;
; SIP/username[:password[:md5secret[:authname]]]@host[:port]
;        This form allows you to specify password or md5secret and authname
;        without altering any authentication data in config.
;        Examples:
;
;        SIP/*98@mysipproxy
;        SIP/sales:topsecret::account02@domain.com:5062
;        SIP/12345678::bc53f0ba8ceb1ded2b70e05c3f91de4f:myname@192.168.0.1
```

```
;
; IPorHost
;       The next server for this call regardless of domain/peer
;
; All of these dial strings specify the SIP request URI.
; In addition, you can specify a specific To: header by adding an
; exclamation mark after the dial string, like
;
;        SIP/sales@mysipproxy!sales@edvina.net
;
; A new feature for 1.8 allows one to specify a host or IP address to use
; when routing the call. This is typically used in tandem with func_srv if
; multiple methods of reaching the same domain exist. The host or IP address
; is specified after the third slash in the dialstring. Examples:
;
; SIP/devicename/extension/IPorHost
; SIP/username@domain//IPorHost
;
; CLI Commands
; ----------------------------------------------------------
; Useful CLI commands to check peers/users:
;   sip show peers              Show all SIP peers (including friends)
;   sip show registry           Show status of hosts we register with
;
;   sip set debug on            Show all SIP messages
;
;   sip reload                  Reload configuration file
;   sip show settings           Show the current channel configuration
;
;------- Naming devices -------------------------------------------------
;
; When naming devices, make sure you understand how Asterisk matches calls
; that come in.
;       1. Asterisk checks the SIP From: address username and matches against
;          names of devices with type=user
;          The name is the text between square brackets [name]
;       2. Asterisk checks the From: addres and matches the list of devices
;          with a type=peer
;       3. Asterisk checks the IP address (and port number) that the INVITE
;          was sent from and matches against any devices with type=peer
;
; Don't mix extensions with the names of the devices. Devices need a unique
; name. The device name is *not* used as phone numbers. Phone numbers are
; anything you declare as an extension in the dialplan (extensions.conf).
;
; When setting up trunks, make sure there's no risk that any From: username
; (caller ID) will match any of your device names, because then Asterisk
; might match the wrong device.
;
; Note: The parameter "username" is not the username and in most cases is
;       not needed at all. Check below. In later releases, it's renamed
;       to "defaultuser" which is a better name, since it is used in
;       combination with the "defaultip" setting.
;-----------------------------------------------------------------------------

; ** Old configuration options **
; The "call-limit" configuation option is considered old is replaced
; by new functionality. To enable callcounters, you use the new
; "callcounter" setting (for extension states in queue and subscriptions)
; You are encouraged to use the dialplan groupcount functionality
; to enforce call limits instead of using this channel-specific method.
; You can still set limits per device in sip.conf or in a database by using
; "setvar" to set variables that can be used in the dialplan for various limits.

[general]
context=default                 ; Default context for incoming calls
;allowguest=no                  ; Allow or reject guest calls (default is yes)
                                ; If your Asterisk is connected to the Internet
                                ; and you have allowguest=yes
                                ; you want to check which services you offer everyone
                                ; out there, by enabling them in the default context (see below).
;match_auth_username=yes        ; if available, match user entry using the
```

```
                              ; 'username' field from the authentication line
                              ; instead of the From: field.
allowoverlap=no               ; Disable overlap dialing support. (Default is yes)
;allowoverlap=yes             ; Enable RFC3578 overlap dialing support.
                              ; Can use the Incomplete application to collect the
                              ; needed digits from an ambiguous dialplan match.
;allowoverlap=dtmf            ; Enable overlap dialing support using DTMF delivery
                              ; methods (inband, RFC2833, SIP INFO) in the early
                              ; media phase.  Uses the Incomplete application to
                              ; collect the needed digits.
;allowtransfer=no             ; Disable all transfers (unless enabled in peers or users)
                              ; Default is enabled. The Dial() options 't' and 'T' are not
                              ; related as to whether SIP transfers are allowed or not.
;realm=mydomain.tld           ; Realm for digest authentication
                              ; defaults to "asterisk". If you set a system name in
                              ; asterisk.conf, it defaults to that system name
                              ; Realms MUST be globally unique according to RFC 3261
                              ; Set this to your host name or domain name
;domainsasrealm=no            ; Use domains list as realms
                              ; You can serve multiple Realms specifying several
                              ; 'domain=...' directives (see below).
                              ; In this case Realm will be based on request 'From'/'To' header
                              ; and should match one of domain names.
                              ; Otherwise default 'realm=...' will be used.

; With the current situation, you can do one of four things:
;  a) Listen on a specific IPv4 address.      Example: bindaddr=192.0.2.1
;  b) Listen on a specific IPv6 address.      Example: bindaddr=2001:db8::1
;  c) Listen on the IPv4 wildcard.            Example: bindaddr=0.0.0.0
;  d) Listen on the IPv4 and IPv6 wildcards.  Example: bindaddr=::
; (You can choose independently for UDP, TCP, and TLS, by specifying different values for
; "udpbindaddr", "tcpbindaddr", and "tlsbindaddr".)
; (Note that using bindaddr=:: will show only a single IPv6 socket in netstat.
;  IPv4 is supported at the same time using IPv4-mapped IPv6 addresses.)
;
; Using bindaddr will only enable UDP support in order to be backwards compatible with those systems
; that were upgraded prior to TCP support. Use udpbindaddr and tcpbindaddr to bind to UDP and TCP
; independently.
;
; You may optionally add a port number. (The default is port 5060 for UDP and TCP, 5061
; for TLS).
;   IPv4 example: bindaddr=0.0.0.0:5062
;   IPv6 example: bindaddr=[::]:5062
;
; The address family of the bound UDP address is used to determine how Asterisk performs
; DNS lookups. In cases a) and c) above, only A records are considered. In case b), only
; AAAA records are considered. In case d), both A and AAAA records are considered. Note,
; however, that Asterisk ignores all records except the first one. In case d), when both A
; and AAAA records are available, either an A or AAAA record will be first, and which one
; depends on the operating system. On systems using glibc, AAAA records are given
; priority.

udpbindaddr=172.16.104.100:5060 ; IP address to bind UDP listen socket to (0.0.0.0 binds to all)
                              ; Optionally add a port number, 192.168.1.1:5062 (default is port 5060)

; When a dialog is started with another SIP endpoint, the other endpoint
; should include an Allow header telling us what SIP methods the endpoint
; implements. However, some endpoints either do not include an Allow header
; or lie about what methods they implement. In the former case, Asterisk
; makes the assumption that the endpoint supports all known SIP methods.
; If you know that your SIP endpoint does not provide support for a specific
; method, then you may provide a comma-separated list of methods that your
; endpoint does not implement in the disallowed_methods option. Note that
; if your endpoint is truthful with its Allow header, then there is no need
; to set this option. This option may be set in the general section or may
; be set per endpoint. If this option is set both in the general section and
; in a peer section, then the peer setting completely overrides the general
; setting (i.e. the result is *not* the union of the two options).
;
; Note also that while Asterisk currently will parse an Allow header to learn
; what methods an endpoint supports, the only actual use for this currently
; is for determining if Asterisk may send connected line UPDATE requests and
```

```
; MESSAGE requests. Its use may be expanded in the future.
;
; disallowed_methods = UPDATE

;
; Note that the TCP and TLS support for chan_sip is currently considered
; experimental.  Since it is new, all of the related configuration options are
; subject to change in any release.  If they are changed, the changes will
; be reflected in this sample configuration file, as well as in the UPGRADE.txt file.
;
tcpenable=yes                    ; Enable server for incoming TCP connections (default is no)
tcpbindaddr=172.16.104.100:5060            ; IP address for TCP server to bind to (0.0.0.0 binds to all
interfaces)
                                 ; Optionally add a port number, 192.168.1.1:5062 (default is port 5060)

;tlsenable=no                    ; Enable server for incoming TLS (secure) connections (default is no)
;tlsbindaddr=0.0.0.0             ; IP address for TLS server to bind to (0.0.0.0) binds to all interfaces)
                                 ; Optionally add a port number, 192.168.1.1:5063 (default is port 5061)
                                 ; Remember that the IP address must match the common name (hostname) in the
                                 ; certificate, so you don't want to bind a TLS socket to multiple IP addresses.
                                 ; For details how to construct a certificate for SIP see
                                 ; http://tools.ietf.org/html/draft-ietf-sip-domain-certs

;tcpauthtimeout = 30             ; tcpauthtimeout specifies the maximum number
                                 ; of seconds a client has to authenticate.  If
                                 ; the client does not authenticate beofre this
                                 ; timeout expires, the client will be
                                 ; disconnected. (default: 30 seconds)

;tcpauthlimit = 100              ; tcpauthlimit specifies the maximum number of
                                 ; unauthenticated sessions that will be allowed
                                 ; to connect at any given time. (default: 100)

transport=udp                    ; Set the default transports.  The order determines the primary default
transport.
                                 ; If tcpenable=no and the transport set is tcp, we will fallback to UDP.

srvlookup=yes                    ; Enable DNS SRV lookups on outbound calls
                                 ; Note: Asterisk only uses the first host
                                 ; in SRV records
                                 ; Disabling DNS SRV lookups disables the
                                 ; ability to place SIP calls based on domain
                                 ; names to some other SIP users on the Internet
                                 ; Specifying a port in a SIP peer definition or
                                 ; when dialing outbound calls will supress SRV
                                 ; lookups for that peer or call.

;pedantic=yes                    ; Enable checking of tags in headers,
                                 ; international character conversions in URIs
                                 ; and multiline formatted headers for strict
                                 ; SIP compatibility (defaults to "yes")

; See https://wiki.asterisk.org/wiki/display/AST/IP+Quality+of+Service for a description of these parameters.
;tos_sip=cs3                     ; Sets TOS for SIP packets.
;tos_audio=ef                    ; Sets TOS for RTP audio packets.
;tos_video=af41                  ; Sets TOS for RTP video packets.
;tos_text=af41                   ; Sets TOS for RTP text packets.

;cos_sip=3                       ; Sets 802.1p priority for SIP packets.
;cos_audio=5                     ; Sets 802.1p priority for RTP audio packets.
;cos_video=4                     ; Sets 802.1p priority for RTP video packets.
;cos_text=3                      ; Sets 802.1p priority for RTP text packets.

;maxexpiry=3600                  ; Maximum allowed time of incoming registrations
                                 ; and subscriptions (seconds)
;minexpiry=60                    ; Minimum length of registrations/subscriptions (default 60)
;defaultexpiry=120               ; Default length of incoming/outgoing registration
;mwiexpiry=3600                  ; Expiry time for outgoing MWI subscriptions
;maxforwards=70                        ; Setting for the SIP Max-Forwards: header (loop prevention)
                                 ; Default value is 70
;qualifyfreq=60                  ; Qualification: How often to check for the host to be up in seconds
                                 ; and reported in milliseconds with sip show settings.
```

```
                              ; Set to low value if you use low timeout for NAT of UDP sessions
                              ; Default: 60
;qualifygap=100                     ; Number of milliseconds between each group of peers being qualified
                              ; Default: 100
;qualifypeers=1                      ; Number of peers in a group to be qualified at the same time
                              ; Default: 1
;notifymimetype=text/plain    ; Allow overriding of mime type in MWI NOTIFY
;buggymwi=no                  ; Cisco SIP firmware doesn't support the MWI RFC
                              ; fully. Enable this option to not get error messages
                              ; when sending MWI to phones with this bug.
;mwi_from=asterisk            ; When sending MWI NOTIFY requests, use this setting in
                              ; the From: header as the "name" portion. Also fill the
                              ; "user" portion of the URI in the From: header with this
                              ; value if no fromuser is set
                              ; Default: empty
;vmexten=voicemail            ; dialplan extension to reach mailbox sets the
                              ; Message-Account in the MWI notify message
                              ; defaults to "asterisk"

; Codec negotiation
;
; When Asterisk is receiving a call, the codec will initially be set to the
; first codec in the allowed codecs defined for the user receiving the call
; that the caller also indicates that it supports. But, after the caller
; starts sending RTP, Asterisk will switch to using whatever codec the caller
; is sending.
;
; When Asterisk is placing a call, the codec used will be the first codec in
; the allowed codecs that the callee indicates that it supports. Asterisk will
; *not* switch to whatever codec the callee is sending.
;
;preferred_codec_only=yes     ; Respond to a SIP invite with the single most preferred codec
                              ; rather than advertising all joint codec capabilities. This
                              ; limits the other side's codec choice to exactly what we prefer.

;disallow=all                 ; First disallow all codecs
;allow=ulaw                   ; Allow codecs in order of preference
;allow=ilbc                   ; see https://wiki.asterisk.org/wiki/display/AST/RTP+Packetization
                              ; for framing options
;
; This option specifies a preference for which music on hold class this channel
; should listen to when put on hold if the music class has not been set on the
; channel with Set(CHANNEL(musicclass)=whatever) in the dialplan, and the peer
; channel putting this one on hold did not suggest a music class.
;
; This option may be specified globally, or on a per-user or per-peer basis.
;
;mohinterpret=default
;
; This option specifies which music on hold class to suggest to the peer channel
; when this channel places the peer on hold. It may be specified globally or on
; a per-user or per-peer basis.
;
;mohsuggest=default
;
;parkinglot=plaza             ; Sets the default parking lot for call parking
                              ; This may also be set for individual users/peers
                              ; Parkinglots are configured in features.conf
;language=en                  ; Default language setting for all users/peers
                              ; This may also be set for individual users/peers
;relaxdtmf=yes                ; Relax dtmf handling
;trustrpid = no               ; If Remote-Party-ID should be trusted
;sendrpid = yes               ; If Remote-Party-ID should be sent (defaults to no)
;sendrpid = rpid              ; Use the "Remote-Party-ID" header
                              ; to send the identity of the remote party
                              ; This is identical to sendrpid=yes
;sendrpid = pai               ; Use the "P-Asserted-Identity" header
                              ; to send the identity of the remote party
;rpid_update = no             ; In certain cases, the only method by which a connected line
                              ; change may be immediately transmitted is with a SIP UPDATE request.
                              ; If communicating with another Asterisk server, and you wish to be able
                              ; transmit such UPDATE messages to it, then you must enable this option.
```

```
                              ; Otherwise, we will have to wait until we can send a reinvite to
                              ; transmit the information.
;prematuremedia=no            ; Some ISDN links send empty media frames before
                              ; the call is in ringing or progress state. The SIP
                              ; channel will then send 183 indicating early media
                              ; which will be empty - thus users get no ring signal.
                              ; Setting this to "yes" will stop any media before we have
                              ; call progress (meaning the SIP channel will not send 183 Session
                              ; Progress for early media). Default is "yes". Also make sure that
                              ; the SIP peer is configured with progressinband=never.
                              ;
                              ; In order for "noanswer" applications to work, you need to run
                              ; the progress() application in the priority before the app.

;progressinband=never         ; If we should generate in-band ringing always
                              ; use 'never' to never use in-band signalling, even in cases
                              ; where some buggy devices might not render it
                              ; Valid values: yes, no, never Default: never
;useragent=Asterisk PBX       ; Allows you to change the user agent string
                              ; The default user agent string also contains the Asterisk
                              ; version. If you don't want to expose this, change the
                              ; useragent string.
;promiscredir = no            ; If yes, allows 302 or REDIR to non-local SIP address
                              ; Note that promiscredir when redirects are made to the
                              ; local system will cause loops since Asterisk is incapable
                              ; of performing a "hairpin" call.
;usereqphone = no             ; If yes, ";user=phone" is added to uri that contains
                              ; a valid phone number
dtmfmode = rfc2833           ; Set default dtmfmode for sending DTMF. Default: rfc2833
                              ; Other options:
                              ; info : SIP INFO messages (application/dtmf-relay)
                              ; shortinfo : SIP INFO messages (application/dtmf)
                              ; inband : Inband audio (requires 64 kbit codec -alaw, ulaw)
                              ; auto : Use rfc2833 if offered, inband otherwise

;compactheaders = yes         ; send compact sip headers.
;
;videosupport=yes             ; Turn on support for SIP video. You need to turn this
                              ; on in this section to get any video support at all.
                              ; You can turn it off on a per peer basis if the general
                              ; video support is enabled, but you can't enable it for
                              ; one peer only without enabling in the general section.
                              ; If you set videosupport to "always", then RTP ports will
                              ; always be set up for video, even on clients that don't
                              ; support it.  This assists callfile-derived calls and
                              ; certain transferred calls to use always use video when
                              ; available. [yes|NO|always]

;maxcallbitrate=384           ; Maximum bitrate for video calls (default 384 kb/s)
                              ; Videosupport and maxcallbitrate is settable
                              ; for peers and users as well
;callevents=no                ; generate manager events when sip ua
                              ; performs events (e.g. hold)
;authfailureevents=no         ; generate manager "peerstatus" events when peer can't
                              ; authenticate with Asterisk. Peerstatus will be "rejected".
;alwaysauthreject = yes       ; When an incoming INVITE or REGISTER is to be rejected,
                              ; for any reason, always reject with an identical response
                              ; equivalent to valid username and invalid password/hash
                              ; instead of letting the requester know whether there was
                              ; a matching user or peer for their request.  This reduces
                              ; the ability of an attacker to scan for valid SIP usernames.
                              ; This option is set to "yes" by default.

;auth_options_requests = yes  ; Enabling this option will authenticate OPTIONS requests just like
                              ; INVITE requests are.  By default this option is disabled.

;g726nonstandard = yes        ; If the peer negotiates G726-32 audio, use AAL2 packing
                              ; order instead of RFC3551 packing order (this is required
                              ; for Sipura and Grandstream ATAs, among others). This is
                              ; contrary to the RFC3551 specification, the peer _should_
                              ; be negotiating AAL2-G726-32 instead :-(
;outboundproxy=proxy.provider.domain           ; send outbound signaling to this proxy, not directly to the
```

```
devices
;outboundproxy=proxy.provider.domain:8080       ; send outbound signaling to this proxy, not directly to the
devices
;outboundproxy=proxy.provider.domain,force      ; Send ALL outbound signalling to proxy, ignoring route: headers
;outboundproxy=tls://proxy.provider.domain      ; same as '=proxy.provider.domain' except we try to connect
with tls
;outboundproxy=192.0.2.1                        ; IPv4 address literal (default port is 5060)
;outboundproxy=2001:db8::1                      ; IPv6 address literal (default port is 5060)
;outboundproxy=192.168.0.2.1:5062               ; IPv4 address literal with explicit port
;outboundproxy=[2001:db8::1]:5062               ; IPv6 address literal with explicit port
;                                               ; (could also be tcp,udp) - defining transports on the proxy
line only
;                                               ; applies for the global proxy, otherwise use the transport=
option
;matchexternaddrlocally = yes     ; Only substitute the externaddr or externhost setting if it matches
                                  ; your localnet setting. Unless you have some sort of strange network
                                  ; setup you will not need to enable this.

;dynamic_exclude_static = yes   ; Disallow all dynamic hosts from registering
                                  ; as any IP address used for staticly defined
                                  ; hosts.  This helps avoid the configuration
                                  ; error of allowing your users to register at
                                  ; the same address as a SIP provider.

;contactdeny=0.0.0.0/0.0.0.0           ; Use contactpermit and contactdeny to
;contactpermit=172.16.0.0/255.255.0.0  ; restrict at what IPs your users may
                                       ; register their phones.

;rtp_engine=asterisk            ; RTP engine to use when communicating with the device

;
; If regcontext is specified, Asterisk will dynamically create and destroy a
; NoOp priority 1 extension for a given peer who registers or unregisters with
; us and have a "regexten=" configuration item.
; Multiple contexts may be specified by separating them with '&'. The
; actual extension is the 'regexten' parameter of the registering peer or its
; name if 'regexten' is not provided.  If more than one context is provided,
; the context must be specified within regexten by appending the desired
; context after '@'.  More than one regexten may be supplied if they are
; separated by '&'.  Patterns may be used in regexten.
;
;regcontext=sipregistrations
;regextenonqualify=yes          ; Default "no"
                                ; If you have qualify on and the peer becomes unreachable
                                ; this setting will enforce inactivation of the regexten
                                ; extension for the peer
;legacy_useroption_parsing=yes        ; Default "no"      ; If you have this option enabled and there are
semicolons
                                                         ; in the user field of a sip URI, the field be truncated
                                                         ; at the first semicolon seen. This effectively makes
                                                         ; semicolon a non-usable character for peer names,
extensions,
                                                         ; and maybe other, less tested things.  This can be useful
                                                         ; for improving compatability with devices that like to use
                                                         ; user options for whatever reason.  The behavior is
similar to
                                                         ; how SIP URI's were typically handled in 1.6.2, hence the
name.

; The shrinkcallerid function removes '(', ' ', ')', non-trailing '.', and '-' not
; in square brackets.  For example, the caller id value 555.5555 becomes 5555555
; when this option is enabled.  Disabling this option results in no modification
; of the caller id value, which is necessary when the caller id represents something
; that must be preserved.  This option can only be used in the [general] section.
; By default this option is on.
;
;shrinkcallerid=yes     ; on by default


;use_q850_reason = no ; Default "no"
                      ; Set to yes add Reason header and use Reason header if it is available.
;
```

```
;----------------------- TLS settings ------------------------------------------------------------
;tlscertfile=</path/to/certificate.pem> ; Certificate file (*.pem format only) to use for TLS connections
                                        ; default is to look for "asterisk.pem" in current directory

;tlsprivatekey=</path/to/private.pem> ; Private key file (*.pem format only) for TLS connections.
                                      ; If no tlsprivatekey is specified, tlscertfile is searched for
                                      ; for both public and private key.

;tlscafile=</path/to/certificate>
;        If the server your connecting to uses a self signed certificate
;        you should have their certificate installed here so the code can
;        verify the authenticity of their certificate.

;tlscapath=</path/to/ca/dir>
;        A directory full of CA certificates.  The files must be named with
;        the CA subject name hash value.
;        (see man SSL_CTX_load_verify_locations for more info)

;tlsdontverifyserver=[yes|no]
;        If set to yes, don't verify the servers certificate when acting as
;        a client.  If you don't have the server's CA certificate you can
;        set this and it will connect without requiring tlscafile to be set.
;        Default is no.

;tlscipher=<SSL cipher string>
;        A string specifying which SSL ciphers to use or not use
;        A list of valid SSL cipher strings can be found at:
;                http://www.openssl.org/docs/apps/ciphers.html#CIPHER_STRINGS
;
;tlsclientmethod=tlsv1     ; values include tlsv1, sslv3, sslv2.
                           ; Specify protocol for outbound client connections.
                           ; If left unspecified, the default is sslv2.
;
;--------------------------- SIP timers -----------------------------------------------------
; These timers are used primarily in INVITE transactions.
; The default for Timer T1 is 500 ms or the measured run-trip time between
; Asterisk and the device if you have qualify=yes for the device.
;
;t1min=100                      ; Minimum roundtrip time for messages to monitored hosts
                                ; Defaults to 100 ms
;timert1=500                    ; Default T1 timer
                                ; Defaults to 500 ms or the measured round-trip
                                ; time to a peer (qualify=yes).
;timerb=32000                   ; Call setup timer. If a provisional response is not received
                                ; in this amount of time, the call will autocongest
                                ; Defaults to 64*timert1

;--------------------------- RTP timers -----------------------------------------------------
; These timers are currently used for both audio and video streams. The RTP timeouts
; are only applied to the audio channel.
; The settings are settable in the global section as well as per device
;
;rtptimeout=60                  ; Terminate call if 60 seconds of no RTP or RTCP activity
                                ; on the audio channel
                                ; when we're not on hold. This is to be able to hangup
                                ; a call in the case of a phone disappearing from the net,
                                ; like a powerloss or grandma tripping over a cable.
;rtpholdtimeout=300             ; Terminate call if 300 seconds of no RTP or RTCP activity
                                ; on the audio channel
                                ; when we're on hold (must be > rtptimeout)
;rtpkeepalive=<secs>            ; Send keepalives in the RTP stream to keep NAT open
                                ; (default is off - zero)

;--------------------------- SIP Session-Timers (RFC 4028)----------------------------------
; SIP Session-Timers provide an end-to-end keep-alive mechanism for active SIP sessions.
; This mechanism can detect and reclaim SIP channels that do not terminate through normal
; signaling procedures. Session-Timers can be configured globally or at a user/peer level.
; The operation of Session-Timers is driven by the following configuration parameters:
;
; * session-timers    - Session-Timers feature operates in the following three modes:
;                         originate : Request and run session-timers always
;                         accept    : Run session-timers only when requested by other UA
```

```
;                          refuse    : Do not run session timers in any case
;                       The default mode of operation is 'accept'.
; * session-expires   - Maximum session refresh interval in seconds. Defaults to 1800 secs.
; * session-minse     - Minimum session refresh interval in seconds. Defualts to 90 secs.
; * session-refresher - The session refresher (uac|uas). Defaults to 'uas'.
;                          uac - Default to the caller initially refreshing when possible
;                          uas - Default to the callee initially refreshing when possible
;
; Note that, due to recommendations in RFC 4028, Asterisk will always honor the other
; endpoint's preference for who will handle refreshes. Asterisk will never override the
; preferences of the other endpoint. Doing so could result in Asterisk and the endpoint
; fighting over who sends the refreshes. This holds true for the initiation of session
; timers and subsequent re-INVITE requests whether Asterisk is the caller or callee, or
; whether Asterisk is currently the refresher or not.
;
session-timers=originate
session-expires=900
;session-minse=90
session-refresher=uac
;
;--------------------------- SIP DEBUGGING ------------------------------------------------
;sipdebug = yes                ; Turn on SIP debugging by default, from
                               ; the moment the channel loads this configuration
;recordhistory=yes             ; Record SIP history by default
                               ; (see sip history / sip no history)
;dumphistory=yes               ; Dump SIP history at end of SIP dialogue
                               ; SIP history is output to the DEBUG logging channel


;--------------------------- STATUS NOTIFICATIONS (SUBSCRIPTIONS) ---------------------------
; You can subscribe to the status of extensions with a "hint" priority
; (See extensions.conf.sample for examples)
; chan_sip support two major formats for notifications: dialog-info and SIMPLE
;
; You will get more detailed reports (busy etc) if you have a call counter enabled
; for a device.
;
; If you set the busylevel, we will indicate busy when we have a number of calls that
; matches the busylevel treshold.
;
; For queues, you will need this level of detail in status reporting, regardless
; if you use SIP subscriptions. Queues and manager use the same internal interface
; for reading status information.
;
; Note: Subscriptions does not work if you have a realtime dialplan and use the
; realtime switch.
;
;allowsubscribe=no             ; Disable support for subscriptions. (Default is yes)
;subscribecontext = default    ; Set a specific context for SUBSCRIBE requests
                               ; Useful to limit subscriptions to local extensions
                               ; Settable per peer/user also
;notifyringing = no            ; Control whether subscriptions already INUSE get sent
                               ; RINGING when another call is sent (default: yes)
;notifyhold = yes              ; Notify subscriptions on HOLD state (default: no)
                               ; Turning on notifyringing and notifyhold will add a lot
                               ; more database transactions if you are using realtime.
;notifycid = yes               ; Control whether caller ID information is sent along with
                               ; dialog-info+xml notifications (supported by snom phones).
                               ; Note that this feature will only work properly when the
                               ; incoming call is using the same extension and context that
                               ; is being used as the hint for the called extension.  This means
                               ; that it won't work when using subscribecontext for your sip
                               ; user or peer (if subscribecontext is different than context).
                               ; This is also limited to a single caller, meaning that if an
                               ; extension is ringing because multiple calls are incoming,
                               ; only one will be used as the source of caller ID.  Specify
                               ; 'ignore-context' to ignore the called context when looking
                               ; for the caller's channel.  The default value is 'no.' Setting
                               ; notifycid to 'ignore-context' also causes call-pickups attempted
                               ; via SNOM's NOTIFY mechanism to set the context for the call pickup
                               ; to PICKUPMARK.
;callcounter = yes             ; Enable call counters on devices. This can be set per
```

```
                                ; device too.

;---------------------------------------- T.38 FAX SUPPORT ---------------------------------
;
; This setting is available in the [general] section as well as in device configurations.
; Setting this to yes enables T.38 FAX (UDPTL) on SIP calls; it defaults to off.
;
; t38pt_udptl = yes            ; Enables T.38 with FEC error correction.
; t38pt_udptl = yes,fec        ; Enables T.38 with FEC error correction.
; t38pt_udptl = yes,redundancy ; Enables T.38 with redundancy error correction.
; t38pt_udptl = yes,none       ; Enables T.38 with no error correction.
;
; In some cases, T.38 endpoints will provide a T38FaxMaxDatagram value (during T.38 setup) that
; is based on an incorrect interpretation of the T.38 recommendation, and results in failures
; because Asterisk does not believe it can send T.38 packets of a reasonable size to that
; endpoint (Cisco media gateways are one example of this situation). In these cases, during a
; T.38 call you will see warning messages on the console/in the logs from the Asterisk UDPTL
; stack complaining about lack of buffer space to send T.38 FAX packets. If this occurs, you
; can set an override (globally, or on a per-device basis) to make Asterisk ignore the
; T38FaxMaxDatagram value specified by the other endpoint, and use a configured value instead.
; This can be done by appending 'maxdatagram=<value>' to the t38pt_udptl configuration option,
; like this:
;
; t38pt_udptl = yes,fec,maxdatagram=400 ; Enables T.38 with FEC error correction and overrides
;                                       ; the other endpoint's provided value to assume we can
;                                       ; send 400 byte T.38 FAX packets to it.
;
; FAX detection will cause the SIP channel to jump to the 'fax' extension (if it exists)
; based one or more events being detected. The events that can be detected are an incoming
; CNG tone or an incoming T.38 re-INVITE request.
;
; faxdetect = yes                ; Default 'no', 'yes' enables both CNG and T.38 detection
; faxdetect = cng                ; Enables only CNG detection
; faxdetect = t38                ; Enables only T.38 detection
;
;---------------------------------------- OUTBOUND SIP REGISTRATIONS  ------------------------
; Asterisk can register as a SIP user agent to a SIP proxy (provider)
; Format for the register statement is:
;       register => [peer?][transport://]user[@domain][:secret[:authuser]]@host[:port][/extension][~expiry]
;
;
;
; domain is either
;        - domain in DNS
;         - host name in DNS
;        - the name of a peer defined below or in realtime
; The domain is where you register your username, so your SIP uri you are registering to
; is username@domain
;
; If no extension is given, the 's' extension is used. The extension needs to
; be defined in extensions.conf to be able to accept calls from this SIP proxy
; (provider).
;
; A similar effect can be achieved by adding a "callbackextension" option in a peer section.
; this is equivalent to having the following line in the general section:
;
;        register => username:secret@host/callbackextension
;
; and more readable because you don't have to write the parameters in two places
; (note that the "port" is ignored - this is a bug that should be fixed).
;
; Note that a register= line doesn't mean that we will match the incoming call in any
; other way than described above. If you want to control where the call enters your
; dialplan, which context, you want to define a peer with the hostname of the provider's
; server. If the provider has multiple servers to place calls to your system, you need
; a peer for each server.
;
; Beginning with Asterisk version 1.6.2, the "user" portion of the register line may
; contain a port number. Since the logical separator between a host and port number is a
; ':' character, and this character is already used to separate between the optional "secret"
; and "authuser" portions of the line, there is a bit of a hoop to jump through if you wish
; to use a port here. That is, you must explicitly provide a "secret" and "authuser" even if
```

```
; they are blank. See the third example below for an illustration.
;
;
; Examples:
;
;register => 1234:password@mysipprovider.com
;
;     This will pass incoming calls to the 's' extension
;
;
;register => 2345:password@sip_proxy/1234
;
;    Register 2345 at sip provider 'sip_proxy'.  Calls from this provider
;    connect to local extension 1234 in extensions.conf, default context,
;    unless you configure a [sip_proxy] section below, and configure a
;    context.
;    Tip 1: Avoid assigning hostname to a sip.conf section like [provider.com]
;    Tip 2: Use separate inbound and outbound sections for SIP providers
;           (instead of type=friend) if you have calls in both directions
;
;register => 3456@mydomain:5082::@mysipprovider.com
;
;    Note that in this example, the optional authuser and secret portions have
;    been left blank because we have specified a port in the user section
;
;register => tls://username:xxxxxx@sip-tls-proxy.example.org
;
;    The 'transport' part defaults to 'udp' but may also be 'tcp' or 'tls'.
;    Using 'udp://' explicitly is also useful in case the username part
;    contains a '/' ('user/name').

;registertimeout=20             ; retry registration calls every 20 seconds (default)
;registerattempts=10            ; Number of registration attempts before we give up
                                ; 0 = continue forever, hammering the other server
                                ; until it accepts the registration
                                ; Default is 0 tries, continue forever

;----------------------------------------- OUTBOUND MWI SUBSCRIPTIONS ------------------------
; Asterisk can subscribe to receive the MWI from another SIP server and store it locally for retrieval
; by other phones. At this time, you can only subscribe using UDP as the transport.
; Format for the mwi register statement is:
;       mwi => user[:secret[:authuser]]@host[:port]/mailbox
;
; Examples:
;mwi => 1234:password@mysipprovider.com/1234
;mwi => 1234:password@myportprovider.com:6969/1234
;mwi => 1234:password:authuser@myauthprovider.com/1234
;mwi => 1234:password:authuser@myauthportprovider.com:6969/1234
;
; MWI received will be stored in the 1234 mailbox of the SIP_Remote context. It can be used by other phones by
following the below:
; mailbox=1234@SIP_Remote
;----------------------------------------- NAT SUPPORT -----------------------
;
; WARNING: SIP operation behind a NAT is tricky and you really need
; to read and understand well the following section.
;
; When Asterisk is behind a NAT device, the "local" address (and port) that
; a socket is bound to has different values when seen from the inside or
; from the outside of the NATted network. Unfortunately this address must
; be communicated to the outside (e.g. in SIP and SDP messages), and in
; order to determine the correct value Asterisk needs to know:
;
; + whether it is talking to someone "inside" or "outside" of the NATted network.
;   This is configured by assigning the "localnet" parameter with a list
;   of network addresses that are considered "inside" of the NATted network.
;   IF LOCALNET IS NOT SET, THE EXTERNAL ADDRESS WILL NOT BE SET CORRECTLY.
;   Multiple entries are allowed, e.g. a reasonable set is the following:
;
;      localnet=192.168.0.0/255.255.0.0 ; RFC 1918 addresses
;      localnet=10.0.0.0/255.0.0.0      ; Also RFC1918
;      localnet=172.16.0.0/12           ; Another RFC1918 with CIDR notation
```

```
;      localnet=169.254.0.0/255.255.0.0 ; Zero conf local network
;
; + the "externally visible" address and port number to be used when talking
;   to a host outside the NAT. This information is derived by one of the
;   following (mutually exclusive) config file parameters:
;
;   a. "externaddr = hostname[:port]" specifies a static address[:port] to
;      be used in SIP and SDP messages.
;      The hostname is looked up only once, when [re]loading sip.conf .
;      If a port number is not present, use the port specified in the "udpbindaddr"
;      (which is not guaranteed to work correctly, because a NAT box might remap the
;      port number as well as the address).
;      This approach can be useful if you have a NAT device where you can
;      configure the mapping statically. Examples:
;
;        externaddr = 12.34.56.78          ; use this address.
;        externaddr = 12.34.56.78:9900     ; use this address and port.
;        externaddr = mynat.my.org:12600   ; Public address of my nat box.
;        externtcpport = 9900   ; The externally mapped tcp port, when Asterisk is behind a static NAT or PAT.
;                               ; externtcpport will default to the externaddr or externhost port if either one
is set.
;        externtlsport = 12600  ; The externally mapped tls port, when Asterisk is behind a static NAT or PAT.
;                               ; externtlsport port will default to the RFC designated port of 5061.
;
;   b. "externhost = hostname[:port]" is similar to "externaddr" except
;      that the hostname is looked up every "externrefresh" seconds
;      (default 10s). This can be useful when your NAT device lets you choose
;      the port mapping, but the IP address is dynamic.
;      Beware, you might suffer from service disruption when the name server
;      resolution fails. Examples:
;
;        externhost=foo.dyndns.net       ; refreshed periodically
;        externrefresh=180               ; change the refresh interval
;
;   Note that at the moment all these mechanism work only for the SIP socket.
;   The IP address discovered with externaddr/externhost is reused for
;   media sessions as well, but the port numbers are not remapped so you
;   may still experience problems.
;
; NOTE 1: in some cases, NAT boxes will use different port numbers in
; the internal<->external mapping. In these cases, the "externaddr" and
; "externhost" might not help you configure addresses properly.
;
; NOTE 2: when using "externaddr" or "externhost", the address part is
; also used as the external address for media sessions. Thus, the port
; information in the SDP may be wrong!
;
; In addition to the above, Asterisk has an additional "nat" parameter to
; address NAT-related issues in incoming SIP or media sessions.
; In particular, depending on the 'nat= ' settings described below, Asterisk
; may override the address/port information specified in the SIP/SDP messages,
; and use the information (sender address) supplied by the network stack instead.
; However, this is only useful if the external traffic can reach us.
; The following settings are allowed (both globally and in individual sections):
;
;        nat = no               ; Use rport if the remote side says to use it.
;        nat = force_rport      ; Force rport to always be on. (default)
;        nat = yes              ; Force rport to always be on and perform comedia RTP handling.
;        nat = comedia          ; Use rport if the remote side says to use it and perform comedia RTP handling.
;
; 'comedia RTP handling' refers to the technique of sending RTP to the port that the
; the other endpoint's RTP arrived from, and means 'connection-oriented media'. This is
; only partially related to RFC 4145 which was referred to as COMEDIA while it was in
; draft form. This method is used to accomodate endpoints that may be located behind
; NAT devices, and as such the port number they tell Asterisk to send RTP packets to
; for their media streams is not actual port number that will be used on the nearer
; side of the NAT.
;
; IT IS IMPORTANT TO NOTE that if the nat setting in the general section differs from
; the nat setting in a peer definition, then the peer username will be discoverable
; by outside parties as Asterisk will respond to different ports for defined and
; undefined peers. For this reason it is recommended to ONLY DEFINE NAT SETTINGS IN THE
```

```
; GENERAL SECTION. Specifically, if nat=force_rport in one section and nat=no in the
; other, then valid peers with settings differing from those in the general section will
; be discoverable.
;
; In addition to these settings, Asterisk *always* uses 'symmetric RTP' mode as defined by
; RFC 4961; Asterisk will always send RTP packets from the same port number it expects
; to receive them on.
;
; The IP address used for media (audio, video, and text) in the SDP can also be overridden by using
; the media_address configuration option. This is only applicable to the general section and
; can not be set per-user or per-peer.
;
; media_address = 172.16.42.1
;
; Through the use of the res_stun_monitor module, Asterisk has the ability to detect when the
; perceived external network address has changed.  When the stun_monitor is installed and
; configured, chan_sip will renew all outbound registrations when the monitor detects any sort
; of network change has occurred. By default this option is enabled, but only takes effect once
; res_stun_monitor is configured.  If res_stun_monitor is enabled and you wish to not
; generate all outbound registrations on a network change, use the option below to disable
; this feature.
;
; subscribe_network_change_event = yes ; on by default

;---------------------------------- MEDIA HANDLING -------------------------------
; By default, Asterisk tries to re-invite media streams to an optimal path. If there's
; no reason for Asterisk to stay in the media path, the media will be redirected.
; This does not really work well in the case where Asterisk is outside and the
; clients are on the inside of a NAT. In that case, you want to set directmedia=nonat.
;
;directmedia=yes                ; Asterisk by default tries to redirect the
                                ; RTP media stream to go directly from
                                ; the caller to the callee.  Some devices do not
                                ; support this (especially if one of them is behind a NAT).
                                ; The default setting is YES. If you have all clients
                                ; behind a NAT, or for some other reason want Asterisk to
                                ; stay in the audio path, you may want to turn this off.

                                ; This setting also affect direct RTP
                                ; at call setup (a new feature in 1.4 - setting up the
                                ; call directly between the endpoints instead of sending
                                ; a re-INVITE).

                                ; Additionally this option does not disable all reINVITE operations.
                                ; It only controls Asterisk generating reINVITEs for the specific
                                ; purpose of setting up a direct media path. If a reINVITE is
                                ; needed to switch a media stream to inactive (when placed on
                                ; hold) or to T.38, it will still be done, regardless of this
                                ; setting. Note that direct T.38 is not supported.

;directmedia=nonat              ; An additional option is to allow media path redirection
                                ; (reinvite) but only when the peer where the media is being
                                ; sent is known to not be behind a NAT (as the RTP core can
                                ; determine it based on the apparent IP address the media
                                ; arrives from).

;directmedia=update             ; Yet a third option... use UPDATE for media path redirection,
                                ; instead of INVITE. This can be combined with 'nonat', as
                                ; 'directmedia=update,nonat'. It implies 'yes'.

;directmedia=outgoing           ; When sending directmedia reinvites, do not send an immediate
                                ; reinvite on an incoming call leg. This option is useful when
                                ; peered with another SIP user agent that is known to send
                                ; immediate direct media reinvites upon call establishment. Setting
                                ; the option in this situation helps to prevent potential glares.
                                ; Setting this option implies 'yes'.

;directrtpsetup=yes             ; Enable the new experimental direct RTP setup. This sets up
                                ; the call directly with media peer-2-peer without re-invites.
                                ; Will not work for video and cases where the callee sends
                                ; RTP payloads and fmtp headers in the 200 OK that does not match the
                                ; callers INVITE. This will also fail if directmedia is enabled when
```

```
                              ; the device is actually behind NAT.

;directmediadeny=0.0.0.0/0      ; Use directmediapermit and directmediadeny to restrict
;directmediapermit=172.16.0.0/16; which peers should be able to pass directmedia to each other
                              ; (There is no default setting, this is just an example)
                              ; Use this if some of your phones are on IP addresses that
                              ; can not reach each other directly. This way you can force
                              ; RTP to always flow through asterisk in such cases.

;ignoresdpversion=yes           ; By default, Asterisk will honor the session version
                              ; number in SDP packets and will only modify the SDP
                              ; session if the version number changes. This option will
                              ; force asterisk to ignore the SDP session version number
                              ; and treat all SDP data as new data.  This is required
                              ; for devices that send us non standard SDP packets
                              ; (observed with Microsoft OCS). By default this option is
                              ; off.

;sdpsession=Asterisk PBX        ; Allows you to change the SDP session name string, (s=)
                              ; Like the useragent parameter, the default user agent string
                              ; also contains the Asterisk version.
;sdpowner=root                 ; Allows you to change the username field in the SDP owner string, (o=)
                              ; This field MUST NOT contain spaces
;encryption=no                 ; Whether to offer SRTP encrypted media (and only SRTP encrypted media)
                              ; on outgoing calls to a peer. Calls will fail with HANGUPCAUSE=58 if
                              ; the peer does not support SRTP. Defaults to no.

;---------------------------------------- REALTIME SUPPORT ------------------------
; For additional information on ARA, the Asterisk Realtime Architecture,
; please read https://wiki.asterisk.org/wiki/display/AST/Realtime+Database+Configuration
;
;rtcachefriends=yes             ; Cache realtime friends by adding them to the internal list
                              ; just like friends added from the config file only on a
                              ; as-needed basis? (yes|no)

;rtsavesysname=yes              ; Save systemname in realtime database at registration
                              ; Default= no

;rtupdate=yes                   ; Send registry updates to database using realtime? (yes|no)
                              ; If set to yes, when a SIP UA registers successfully, the ip address,
                              ; the origination port, the registration period, and the username of
                              ; the UA will be set to database via realtime.
                              ; If not present, defaults to 'yes'. Note: realtime peers will
                              ; probably not function across reloads in the way that you expect, if
                              ; you turn this option off.
;rtautoclear=yes               ; Auto-Expire friends created on the fly on the same schedule
                              ; as if it had just registered? (yes|no|<seconds>)
                              ; If set to yes, when the registration expires, the friend will
                              ; vanish from the configuration until requested again. If set
                              ; to an integer, friends expire within this number of seconds
                              ; instead of the registration interval.

;ignoreregexpire=yes            ; Enabling this setting has two functions:
                              ;
                              ; For non-realtime peers, when their registration expires, the
                              ; information will _not_ be removed from memory or the Asterisk database
                              ; if you attempt to place a call to the peer, the existing information
                              ; will be used in spite of it having expired
                              ;
                              ; For realtime peers, when the peer is retrieved from realtime storage,
                              ; the registration information will be used regardless of whether
                              ; it has expired or not; if it expires while the realtime peer
                              ; is still in memory (due to caching or other reasons), the
                              ; information will not be removed from realtime storage

;---------------------------------------- SIP DOMAIN SUPPORT ----------------------
; Incoming INVITE and REFER messages can be matched against a list of 'allowed'
; domains, each of which can direct the call to a specific context if desired.
; By default, all domains are accepted and sent to the default context or the
; context associated with the user/peer placing the call.
; REGISTER to non-local domains will be automatically denied if a domain
; list is configured.
```

```
;
; Domains can be specified using:
; domain=<domain>[,<context>]
; Examples:
; domain=myasterisk.dom
; domain=customer.com,customer-context
;
; In addition, all the 'default' domains associated with a server should be
; added if incoming request filtering is desired.
; autodomain=yes
;
; To disallow requests for domains not serviced by this server:
; allowexternaldomains=no

;domain=mydomain.tld,mydomain-incoming
                              ; Add domain and configure incoming context
                              ; for external calls to this domain
;domain=1.2.3.4               ; Add IP address as local domain
                              ; You can have several "domain" settings
;allowexternaldomains=no      ; Disable INVITE and REFER to non-local domains
                              ; Default is yes
;autodomain=yes               ; Turn this on to have Asterisk add local host
                              ; name and local IP to domain list.

; fromdomain=mydomain.tld     ; When making outbound SIP INVITEs to
                              ; non-peers, use your primary domain "identity"
                              ; for From: headers instead of just your IP
                              ; address. This is to be polite and
                              ; it may be a mandatory requirement for some
                              ; destinations which do not have a prior
                              ; account relationship with your server.

;------------------------------ Advice of Charge CONFIGURATION -------------------------
; snom_aoc_enabled = yes;     ; This options turns on and off support for sending AOC-D and
                              ; AOC-E to snom endpoints.  This option can be used both in the
                              ; peer and global scope.  The default for this option is off.


;------------------------------ JITTER BUFFER CONFIGURATION -------------------------
; jbenable = yes              ; Enables the use of a jitterbuffer on the receiving side of a
                              ; SIP channel. Defaults to "no". An enabled jitterbuffer will
                              ; be used only if the sending side can create and the receiving
                              ; side can not accept jitter. The SIP channel can accept jitter,
                              ; thus a jitterbuffer on the receive SIP side will be used only
                              ; if it is forced and enabled.

; jbforce = no                ; Forces the use of a jitterbuffer on the receive side of a SIP
                              ; channel. Defaults to "no".

; jbmaxsize = 200             ; Max length of the jitterbuffer in milliseconds.

; jbresyncthreshold = 1000    ; Jump in the frame timestamps over which the jitterbuffer is
                              ; resynchronized. Useful to improve the quality of the voice, with
                              ; big jumps in/broken timestamps, usually sent from exotic devices
                              ; and programs. Defaults to 1000.

; jbimpl = fixed              ; Jitterbuffer implementation, used on the receiving side of a SIP
                              ; channel. Two implementations are currently available - "fixed"
                              ; (with size always equals to jbmaxsize) and "adaptive" (with
                              ; variable size, actually the new jb of IAX2). Defaults to fixed.

; jbtargetextra = 40          ; This option only affects the jb when 'jbimpl = adaptive' is set.
                              ; The option represents the number of milliseconds by which the new jitter buffer
                              ; will pad its size. the default is 40, so without modification, the new
                              ; jitter buffer will set its size to the jitter value plus 40 milliseconds.
                              ; increasing this value may help if your network normally has low jitter,
                              ; but occasionally has spikes.

; jblog = no                  ; Enables jitterbuffer frame logging. Defaults to "no".

;------------------------------ SIP_CAUSE reporting --------------------------------
; storesipcause = no          ; This option causes chan_sip to set the
```

```
                              ; HASH(SIP_CAUSE,<channel name>) channel variable
                              ; to the value of the last sip response.
                              ; WARNING: enabling this option carries a
                              ; significant performance burden. It should only
                              ; be used in low call volume situations. This
                              ; option defaults to "no".

;------------------------------------------------------------------------------

[authentication]
; Global credentials for outbound calls, i.e. when a proxy challenges your
; Asterisk server for authentication. These credentials override
; any credentials in peer/register definition if realm is matched.
;
; This way, Asterisk can authenticate for outbound calls to other
; realms. We match realm on the proxy challenge and pick an set of
; credentials from this list
; Syntax:
;       auth = <user>:<secret>@<realm>
;       auth = <user>#<md5secret>@<realm>
; Example:
;auth=mark:topsecret@digium.com
;
; You may also add auth= statements to [peer] definitions
; Peer auth= override all other authentication settings if we match on realm

;------------------------------------------------------------------------------
; DEVICE CONFIGURATION
;
; SIP entities have a 'type' which determines their roles within Asterisk.
; * For entities with 'type=peer':
;   Peers handle both inbound and outbound calls and are matched by ip/port, so for
;   The case of incoming calls from the peer, the IP address must match in order for
;   The invitation to work. This means calls made from either direction won't work if
;   The peer is unregistered while host=dynamic or if the host is otherise not set to
;   the correct IP of the sender.
; * For entities with 'type=user':
;   Asterisk users handle inbound calls only (meaning they call Asterisk, Asterisk can't
;   call them) and are matched by their authorization information (authname and secret).
;   Asterisk doesn't rely on their IP and will accept calls regardless of the host setting
;   as long as the incoming SIP invite authorizes successfully.
; * For entities with 'type=friend':
;   Asterisk will create the entity as both a friend and a peer. Asterisk will accept
;   calls from friends like it would for users, requiring only that the authorization
;   matches rather than the IP address. Since it is also a peer, a friend entity can
;   be called as long as its IP is known to Asterisk. In the case of host=dynamic,
;   this means it is necessary for the entity to register before Asterisk can call it.
;
; Use remotesecret for outbound authentication, and secret for authenticating
; inbound requests. For historical reasons, if no remotesecret is supplied for an
; outbound registration or call, the secret will be used.
;
; For device names, we recommend using only a-z, numerics (0-9) and underscore
;
; For local phones, type=friend works most of the time
;
; If you have one-way audio, you probably have NAT problems.
; If Asterisk is on a public IP, and the phone is inside of a NAT device
; you will need to configure nat option for those phones.
; Also, turn on qualify=yes to keep the nat session open
;
; Configuration options available
; --------------------
; context
; callingpres
; permit
; deny
; secret
; md5secret
; remotesecret
; transport
; dtmfmode
```

```
; directmedia
; nat
; callgroup
; pickupgroup
; language
; allow
; disallow
; insecure
; trustrpid
; progressinband
; promiscredir
; useclientcode
; accountcode
; setvar
; callerid
; amaflags
; callcounter
; busylevel
; allowoverlap
; allowsubscribe
; allowtransfer
; ignoresdpversion
; subscribecontext
; template
; videosupport
; maxcallbitrate
; rfc2833compensate
; mailbox
; session-timers
; session-expires
; session-minse
; session-refresher
; t38pt_usertpsource
; regexten
; fromdomain
; fromuser
; host
; port
; qualify
; defaultip
; defaultuser
; rtptimeout
; rtpholdtimeout
; sendrpid
; outboundproxy
; rfc2833compensate
; callbackextension
; timert1
; timerb
; qualifyfreq
; t38pt_usertpsource
; contactpermit         ; Limit what a host may register as (a neat trick
; contactdeny           ; is to register at the same IP as a SIP provider,
;                       ; then call oneself, and get redirected to that
;                       ; same location).
; directmediapermit
; directmediadeny
; unsolicited_mailbox
; use_q850_reason
; maxforwards
; encryption

;[sip_proxy]
; For incoming calls only. Example: FWD (Free World Dialup)
; We match on IP address of the proxy for incoming calls
; since we can not match on username (caller id)
;type=peer
;context=from-fwd
;host=fwd.pulver.com

;[sip_proxy-out]
;type=peer                       ; we only want to call out, not be called
```

```
;remotesecret=guessit             ; Our password to their service
;defaultuser=yourusername         ; Authentication user for outbound proxies
;fromuser=yourusername            ; Many SIP providers require this!
;fromdomain=provider.sip.domain
;host=box.provider.com
;transport=udp,tcp                ; This sets the default transport type to udp for outgoing, and will
;                                 ; accept both tcp and udp. The default transport type is only used for
;                                 ; outbound messages until a Registration takes place.  During the
;                                 ; peer Registration the transport type may change to another supported
;                                 ; type if the peer requests so.

;usereqphone=yes                  ; This provider requires ";user=phone" on URI
;callcounter=yes                  ; Enable call counter
;busylevel=2                      ; Signal busy at 2 or more calls
;outboundproxy=proxy.provider.domain  ; send outbound signaling to this proxy, not directly to the peer
;port=80                          ; The port number we want to connect to on the remote side
                                  ; Also used as "defaultport" in combination with "defaultip" settings

;--- sample definition for a provider
;[provider1]
;type=peer
;host=sip.provider1.com
;fromuser=4015552299              ; how your provider knows you
;remotesecret=youwillneverguessit ; The password we use to authenticate to them
;secret=gissadetdu                ; The password they use to contact us
;callbackextension=123            ; Register with this server and require calls coming back to this extension
;transport=udp,tcp                ; This sets the transport type to udp for outgoing, and will
;                                 ;   accept both tcp and udp. Default is udp. The first transport
;                                 ;   listed will always be used for outgoing connections.
;unsolicited_mailbox=4015552299   ; If the remote SIP server sends an unsolicited MWI NOTIFY message the new/old
;                                 ;   message count will be stored in the configured virtual mailbox. It can be
used
;                                 ;   by any device supporting MWI by specifying <configured value>@SIP_Remote
as the
;                                 ;   mailbox.

;
; Because you might have a large number of similar sections, it is generally
; convenient to use templates for the common parameters, and add them
; the the various sections. Examples are below, and we can even leave
; the templates uncommented as they will not harm:

[basic-options](!)               ; a template
        dtmfmode=rfc2833
        context=from-office
        type=friend

[natted-phone](!,basic-options)  ; another template inheriting basic-options
        directmedia=no
        host=dynamic

[public-phone](!,basic-options)  ; another template inheriting basic-options
        directmedia=yes

[my-codecs](!)                   ; a template for my preferred codecs
        disallow=all
        allow=ilbc
        allow=g729
        allow=gsm
        allow=g723
        allow=ulaw

[ulaw-phone](!)                  ; and another one for ulaw-only
        disallow=all
        allow=ulaw

; and finally instantiate a few phones
;
; [2133](natted-phone,my-codecs)
;        secret = peekaboo
; [2134](natted-phone,ulaw-phone)
;        secret = not_very_secret
```

```
; [2136](public-phone,ulaw-phone)
;        secret = not_very_secret_either
; ...
;

; Standard configurations not using templates look like this:
;
;[grandstream1]
;type=friend
;context=from-sip                ; Where to start in the dialplan when this phone calls
;callerid=John Doe <1234>        ; Full caller ID, to override the phones config
                                 ; on incoming calls to Asterisk
;host=192.168.0.23               ; we have a static but private IP address
                                 ; No registration allowed
;directmedia=yes                 ; allow RTP voice traffic to bypass Asterisk
;dtmfmode=info                   ; either RFC2833 or INFO for the BudgeTone
;call-limit=1                    ; permit only 1 outgoing call and 1 incoming call at a time
                                 ; from the phone to asterisk (deprecated)
                                 ; 1 for the explicit peer, 1 for the explicit user,
                                 ; remember that a friend equals 1 peer and 1 user in
                                 ; memory
                                 ; There is no combined call counter for a "friend"
                                 ; so there's currently no way in sip.conf to limit
                                 ; to one inbound or outbound call per phone. Use
                                 ; the group counters in the dial plan for that.
                                 ;
;mailbox=1234@default            ; mailbox 1234 in voicemail context "default"
;disallow=all                    ; need to disallow=all before we can use allow=
;allow=ulaw                      ; Note: In user sections the order of codecs
                                 ; listed with allow= does NOT matter!
;allow=alaw
;allow=g723.1                    ; Asterisk only supports g723.1 pass-thru!
;allow=g729                      ; Pass-thru only unless g729 license obtained
;callingpres=allowed_passed_screen ; Set caller ID presentation
                                 ; See README.callingpres for more information

;[xlite1]
; Turn off silence suppression in X-Lite ("Transmit Silence"=YES)!
; Note that Xlite sends NAT keep-alive packets, so qualify=yes is not needed
;type=friend
;regexten=1234                   ; When they register, create extension 1234
;callerid="Jane Smith" <5678>
;host=dynamic                    ; This device needs to register
;directmedia=no                  ; Typically set to NO if behind NAT
;disallow=all
;allow=gsm                       ; GSM consumes far less bandwidth than ulaw
;allow=ulaw
;allow=alaw
;mailbox=1234@default,1233@default ; Subscribe to status of multiple mailboxes
;registertrying=yes              ; Send a 100 Trying when the device registers.

;[snom]
;type=friend                     ; Friends place calls and receive calls
;context=from-sip                ; Context for incoming calls from this user
;secret=blah
;subscribecontext=localextensions ; Only allow SUBSCRIBE for local extensions
;language=de                     ; Use German prompts for this user
;host=dynamic                    ; This peer register with us
;dtmfmode=inband                 ; Choices are inband, rfc2833, or info
;defaultip=192.168.0.59          ; IP used until peer registers
;mailbox=1234@context,2345       ; Mailbox(-es) for message waiting indicator
;subscribemwi=yes                ; Only send notifications if this phone
                                 ; subscribes for mailbox notification
;vmexten=voicemail               ; dialplan extension to reach mailbox
                                 ; sets the Message-Account in the MWI notify message
                                 ; defaults to global vmexten which defaults to "asterisk"
;disallow=all
;allow=ulaw                      ; dtmfmode=inband only works with ulaw or alaw!


;[polycom]
;type=friend                     ; Friends place calls and receive calls
```

```
;context=from-sip               ; Context for incoming calls from this user
;secret=blahpoly
;host=dynamic                   ; This peer register with us
dtmfmode=rfc2833               ; Choices are inband, rfc2833, or info
;defaultuser=polly              ; Username to use in INVITE until peer registers
;defaultip=192.168.40.123
                                ; Normally you do NOT need to set this parameter
;disallow=all
;allow=ulaw                     ; dtmfmode=inband only works with ulaw or alaw!
;progressinband=no              ; Polycom phones don't work properly with "never"


;[pingtel]
;type=friend
;secret=blah
;host=dynamic
;insecure=port                  ; Allow matching of peer by IP address without
                                ; matching port number
;insecure=invite                ; Do not require authentication of incoming INVITEs
;insecure=port,invite           ; (both)
;qualify=1000                   ; Consider it down if it's 1 second to reply
                                ; Helps with NAT session
                                ; qualify=yes uses default value
;qualifyfreq=60                 ; Qualification: How often to check for the
                                ; host to be up in seconds
                                ; Set to low value if you use low timeout for
                                ; NAT of UDP sessions
;
; Call group and Pickup group should be in the range from 0 to 63
;
;callgroup=1,3-4                ; We are in caller groups 1,3,4
;pickupgroup=1,3-5              ; We can do call pick-p for call group 1,3,4,5
;defaultip=192.168.0.60         ; IP address to use if peer has not registered
;deny=0.0.0.0/0.0.0.0           ; ACL: Control access to this account based on IP address
;permit=192.168.0.60/255.255.255.0
;permit=192.168.0.60/24         ; we can also use CIDR notation for subnet masks
;permit=2001:db8::/32           ; IPv6 ACLs can be specified if desired. IPv6 ACLs
                                ; apply only to IPv6 addresses, and IPv4 ACLs apply
                                ; only to IPv4 addresses.

;[cisco1]
;type=friend
;secret=blah
;qualify=200                    ; Qualify peer is no more than 200ms away
;host=dynamic                   ; This device registers with us
;directmedia=no                 ; Asterisk by default tries to redirect the
                                ; RTP media stream (audio) to go directly from
                                ; the caller to the callee.  Some devices do not
                                ; support this (especially if one of them is
                                ; behind a NAT).
;defaultip=192.168.0.4          ; IP address to use until registration
;defaultuser=goran              ; Username to use when calling this device before registration
                                ; Normally you do NOT need to set this parameter
;setvar=CUSTID=5678             ; Channel variable to be set for all calls from or to this device
;setvar=ATTENDED_TRANSFER_COMPLETE_SOUND=beep   ; This channel variable will
                                               ; cause the given audio file to
                                               ; be played upon completion of
                                               ; an attended transfer.

;[pre14-asterisk]
;type=friend
;secret=digium
;host=dynamic
rfc2833compensate=yes          ; Compensate for pre-1.4 DTMF transmission from another Asterisk machine.
                                ; You must have this turned on or DTMF reception will work improperly.
;t38pt_usertpsource=yes         ; Use the source IP address of RTP as the destination IP address for UDPTL
packets
                                ; if the nat option is enabled. If a single RTP packet is received Asterisk
will know the
                                ; external IP address of the remote device. If port forwarding is done at the
client side
```

```
                                ; then UDPTL will flow to the remote device.
```

Sample Config of Users and Peers

For testing, use the following sample **sip.conf**:

> 9199920301 and 9199920302 are Asterisk registered endpoints.
>
> IOTSBX3 is Peer which is Ribbon SBC SWe Lite in this lab setup.

```
[9199920301] type=friend ; Friends place calls and receive calls
context=default ; Context for incoming calls from this user
host=dynamic
dtmfmode=rfc2833
allow=ulaw
allow=alaw
allow=g729
callingpres=allowed_passed_screen ; if caller-id need to be passed
;callingpres=prohib_passed_screen ; if caller-id need to be hidden, currently its commented out
t38pt_udptl = yes
faxdetect = yes ; to make asterisk recognize fax tone

[9199920302]
type=friend
context=default
host=dynamic
allow=ulaw
allow=alaw
allow=g729
callingpres=allowed_passed_screen
;callingpres=prohib_passed_screen
mailbox=9199920302@Asterisk.asterisk.com ; for voice mail configuration
t38pt_udptl = yes
faxdetect = yes

[IOTSBX3]
context=default ; different context is given
type=peer ; to mention this as Peer
host=172.16.102.10 ; IP address of SBC - here, we give ingress IP address of SWe Lite
directmedia=no
canreinvite=yes
dtmfmode=rfc2833
allow=ulaw
allow=alaw
allow=g729
auth=123:123@Realm ; for authentication
username=123
qualify=yes
```

For configuring the dial plan (routing) in Asterisk, use the following **extensions.conf** file.

```
exten => _9199920301,1,NoOp(${EXTEN})
same => n,Goto(AsteriskUser1,${EXTEN},1)

exten => _9199920302,1,NoOp(${EXTEN:1})
same => n,Goto(AsteriskUser2,${EXTEN:1},1)

[AsteriskUser1]
exten => _X.,1,Dial(SIP/9199920301/${EXTEN})

[AsteriskUser2]
exten => _X.,1,Dial(SIP/9199920302/${EXTEN})

exten => _0X.,1,NoOp(${EXTEN:1}) ; for dialing any number which is outside Asterisk PBX, dial with prefix 0 and
then it will be routed separately
same => n,Goto(outgoing,${EXTEN:1},1)

[outgoing]
exten => _X.,1,Dial(SIP/IOTSBX3/${EXTEN})
```

Voicemail

For configuring voice mail in Asterisk, ensure **extensions.conf** and **voicemail.conf** have the necessary configurations.

In the following example, the Asterisk user 9199920302 is configured with the voicemail capability in the extensions.conf.

```
vi /etc/asterisk/extensions.conf

exten => 9199920302,2,NoOp(Call for )
same => n,Dial(SIP/9199920302,10)
same => n,VoiceMail(pbxuser2@default)
same => n,Hangup
```

The following example shows a voicemail.conf sample file.

```
[general]
format=wav49|gsm|wav
serveremail=asterisk
attach=yes
skipms=3000
maxsilence=10
silencethreshold=128
maxlogins=3

emaildateformat=%A, %B %d, %Y at %r
pagerdateformat=%A, %B %d, %Y at %r
sendvoicemail=yes ; Allow the user to compose and send a voicemail while inside

[zonemessages]
eastern=America/New_York|'vm-received' Q 'digits/at' IMp
central=America/Chicago|'vm-received' Q 'digits/at' IMp
central24=America/Chicago|'vm-received' q 'digits/at' H N 'hours'
military=Zulu|'vm-received' q 'digits/at' H N 'hours' 'phonetic/z_p'
european=Europe/Copenhagen|'vm-received' a d b 'digits/at' HM



[default]
1234 => 1234,fullname,root@Asterisk.asterisk.com,root@Asterisk.asterisk.com
9902875981 => 1234,Payodhi,pdwivedi@sonusnet.com,,Tz=european
9199920302 => 1234,pbxuser2,pbxuser2@Asterisk.asterisk.com,pbxuser2@Asterisk.asterisk.com,Tz=european
pbxuser2  => 1234,pbxuser2,pbxuser2@Asterisk.asterisk.com,pbxuser2@Asterisk.asterisk.com,Tz=european
```

Call Park and Pickup

For Call Park and retrieval in the Asterisk PBX, use the following configuration in the extensions.conf and features.conf files.

```
vi /etc/asterisk/extensions.conf

[default]
include => parkedcalls
exten => 700,1,Goto(parkinglot,${ARG1},1) [parkinglot]
exten => s,1,NoOp(once a parked call times out it will resume here)

include => parkedcalls
exten => i,1,Playback(pbx-invalidpark)
exten => i,2,Hangup
```

```
vi /etc/asterisk/features.conf

[general]
parkext => 700 ; What extension to dial to park. Set per parking lot.
;parkext_exclusive=yes ; Specify that the parkext created for this parking lot ; will only access this parking
lot. (default is no)
parkpos => 701-720 ; What extensions to park calls on. (defafult parking lot) ; These need to be numeric, as
Asterisk starts from the start position ; and increments with one for the next parked call. ; Set per parking
lot.
context => parkedcalls ; Which context parked calls are in (default parking lot) ; Set per parking lot.
```

Network Conference

Configure one extension or number in Asterisk PBX to configure a "Meet me" conference as shown in the following extensions.conf file. All Asterisk users can connect to the conference bridge by dialing the extension or number configured in the Asterisk.

```
vi /etc/asterisk/extensions.conf

[default]
exten => 2115,1,Answer()
exten => 2115,n,Set(CHANNEL(language)=gb-f)
exten => 2115,n,Set(CHANNEL(musicclass)=default)
exten => 2115,n,ConfBridge(1234,Mcs,123)
```

Music On Hold

Enter an appropriate context in the /etc/asterisk/musiconhold.conf file as shown in the following example.

```
vi /etc/asterisk/musiconhold.conf

[default]
mode=files
directory=moh
```

Other Features

For configuring any other features in Asterisk, check the **features.conf** file.

Commit changes in Asterisk using reload command as shown below:

```
asterisk -v
Asterisk*CLI> sip reload
Asterisk*CLI> dialplan reload
Asterisk*CLI> voicemail reload
```

Check the Asterisk user configuration using below command:

```
asterisk -v
Asterisk*CLI> sip show users
Username     Secret     Accountcode       Def.Context   ACL         Forcerport
9199920302   default       No                Yes
9199920301   default       No                Yes
```

Check the Asterisk user registrations using below command:

> If a user is registered, IP Address is visible. If a user is not registered, IP address is not visible.

```
Asterisk*CLI> sip show peers
Name/username             Host             Dyn       Forcerport       ACL     Port    Status
9199920301/9199920301     172.16.108.28     D          N                       5060    Unmonitored
9199920302/9199920302     (Unspecified)     D          N                        0      Unmonitored
```

Supplementary Services & Features Coverage

The following checklist depicts the set of services/features covered through the configuration defined in this Interop Guide.

Sr. No.	Supplementary Services/ Features	Coverage
1	Registration over UDP/TCP/TLS	✓
2	Basic Call Setup & Termination	✓
3	Simultaneous Ringing	✓
4	Music on Hold	✓
5	Cancel Call & Call Rejection	✓
6	Call Forwarding Busy	✓
7	Call Forward No Answer	✓
8	Call Transfer (Attended)	✓
9	Call Transfer (Blind/ Unattended)	✓
10	Voice mail & retrieval	✓
11	Call Park & Pickup	✓
12	Call Conference	✓
13	Calling Party ID Restriction	✓
14	G.711 Fax	✓

Legend

Supported	✓
Not Supported	✗

Caveats

The following items should be noted for this interop:

- Ribbon SBC SWe Lite does not support Dynamic PBX Registration.
- Ribbon SBC SWe Lite does not have support for out-of-dialog SIP INFO.

Support

For any support related queries about this guide, please contact your local Ribbon representative or the following numbers and website:

- Sales and Support: 1-833-742-2661
- Other Queries: 1-877-412-8867
- Website: https://ribboncommunications.com/about-us

References

For detailed information about Ribbon products & solutions, please visit:

https://ribboncommunications.com/products

Conclusion

This Interoperability Guide describes how to successfully configure the Asterisk PBX interop involving Ribbon SBC SWe Lite, Ribbon SBC SWe Core & Ribbon C20-AS products.

The guide provides information about all tested features and capabilities. It records all limitations, notes, and observations to provide you with an accurate understanding of what this guide covers and what it does not.

Configuration guidance in this document enables you to replicate the same base setup; however, you may require additional configuration changes to suit the exact deployment environment.