Telstra Interop involving Ribbon QSBC and PSX Redirector

Table of Contents

Document Overview

This document outlines the configuration best practices for Telstra call flows involving Ribbon QSBC when deployed with Ribbon PSX (Redirector).

Ribbon QSBC is a network element deployed to protect SIP based Voice over Internet Protocol (VoIP) networks. Early deployments of SBCs were focused on the borders between two service provider networks in a peering environment. This role has now expanded to include significant deployments between a service provider's access network and a backbone network to provide service to residential and/or enterprise customers.

Ribbon PSX Ribbon's centralized policy and routing solution (PSX) provides a better way to manage the security, complexity, and cost of routing calls across any operator's network.

This guide contains the following sections:

- Section A: QSBC Configuration
  - Captures general QSBC configurations for Interop with PSX Redirector
- Section B: PSX Redirector Configuration
  - Captures PSX Redirector configuration required for QSBC routing

> **References**
> For additional information on the Ribbon QSBC and Ribbon PSX - Please refer to https://ribboncommunications.com/

Scope

It is not the goal of this guide to provide detailed configurations that will meet the requirements of every customer. Use this guide as a starting point and build the QSBC, PSX product's configurations in consultation with network design and deployment engineers.

Audience

This is a technical document intended for telecommunications engineers with the purpose of configuring Ribbon QSBCs and PSXs. Steps will require navigating the product guide as well as the Operations guide. Understanding the basic concepts of TCP/UDP, IP/Routing, and SIP/RTP is needed to complete the configuration and any necessary troubleshooting.

> **Note**
> This configuration guide is offered as a convenience to Ribbon customers. The specifications and information regarding the product in this guide are subject to change without notice. All statements, information, and recommendations in this guide are believed to be accurate but are presented without warranty of any kind, express or implied, and are provided “AS IS”. Users must take full responsibility for the application of the specifications and information in this guide.

Product and Device Details

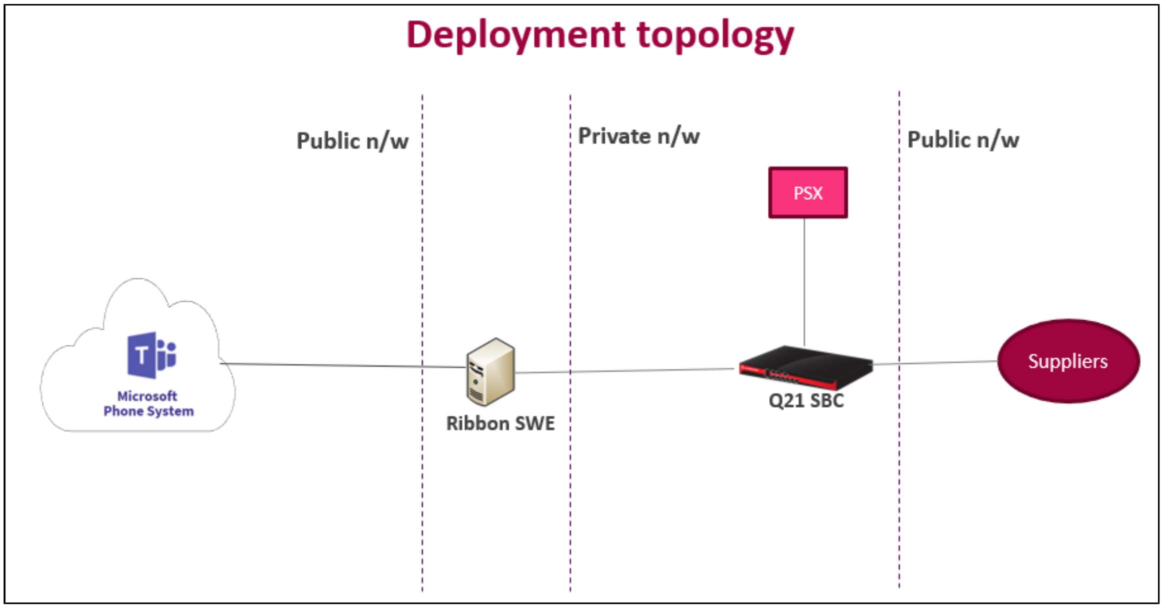
The sample configuration in this document uses the following equipment and software:

Table 1: Requirements

Vendor	Equipment	Software Version
Ribbon Communications	Ribbon QSBC	V9.3.11
	Ribbon PSX	V12.2.2R0

Network Topology Diagram

The following diagram shows the Deployment topology

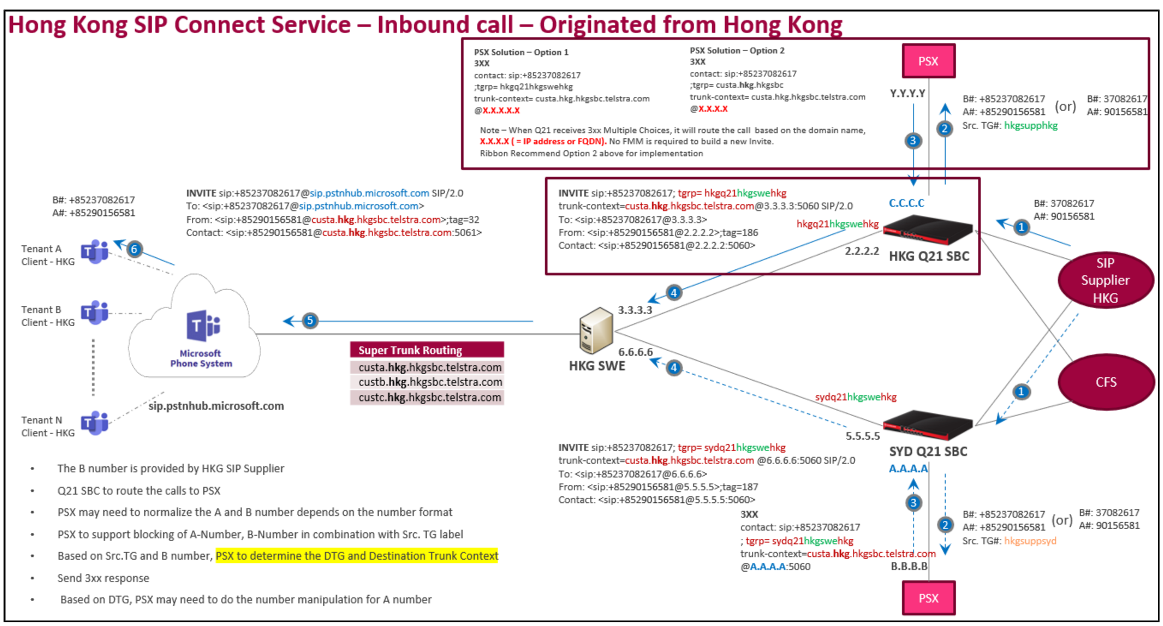


Call Flow Diagram

The following topology diagram shows a Telstra network with a sample inbound call with QSBC & PSX (Redirector).

For the Lab network, we used Simulator instead of the HKG SWE and SIP supplier.

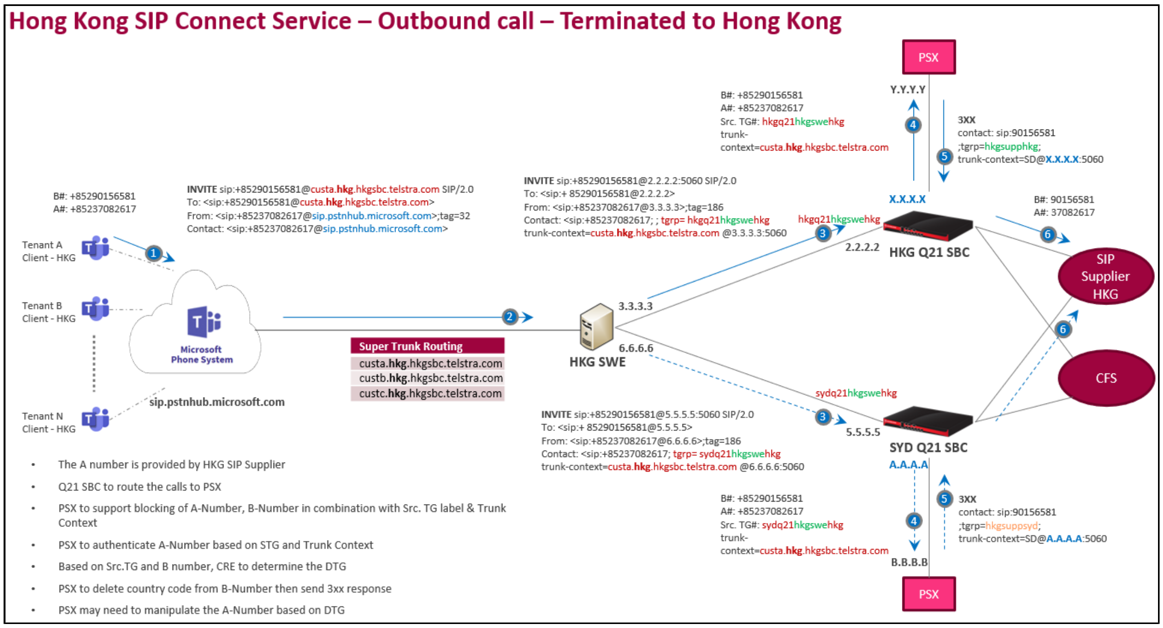
> Below diagram shows signaling flow only. Media flows are bi-directional covering all equipments in the below signaling flow except PSX (PSX doesn't handle media).
>
> Media flows not shown explicitly for simplicity.



The following topology diagram shows the Telstra network with a sample outbound call with QSBC & PSX (Redirector).

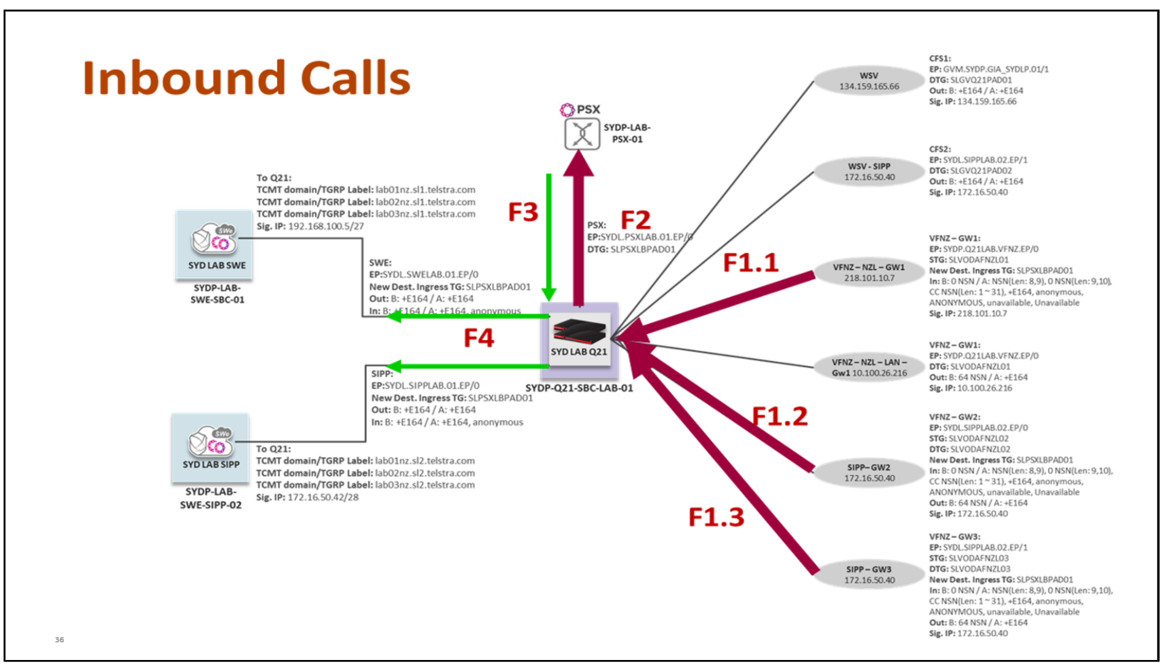
Below diagram shows signaling flow only. Media flows are bi-directional covering all equipments in the below signaling flow except PSX (PSX doesn't handle media).

Media flows not shown explicitly for simplicity.

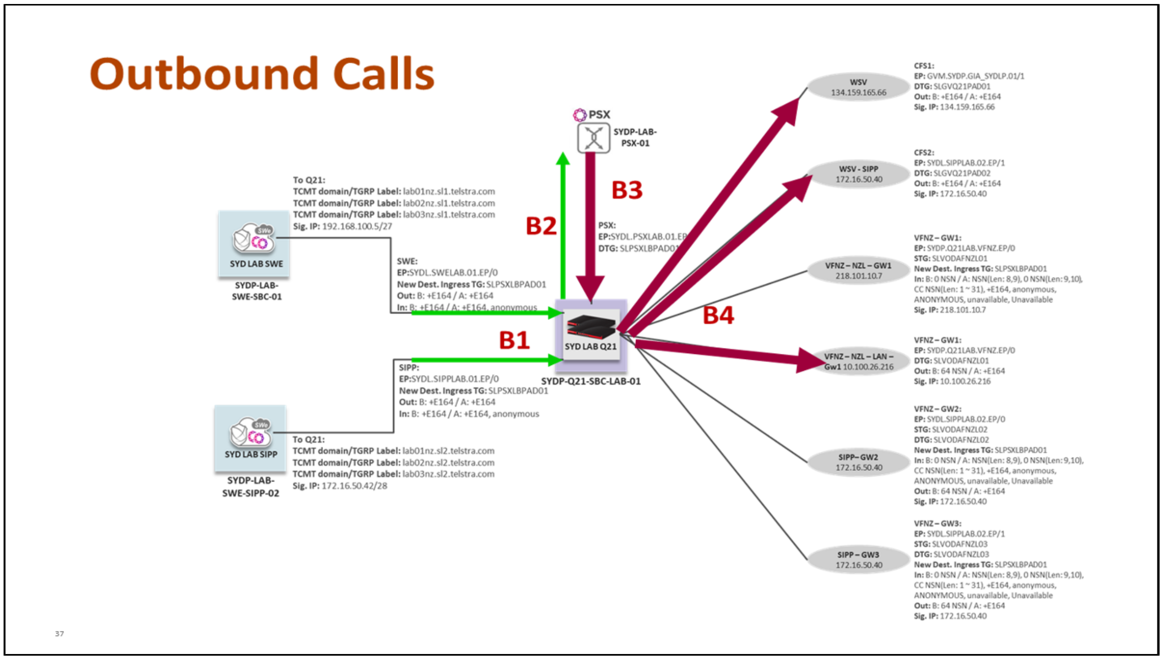


Section A: QSBC Configuration

The following diagram shows the Q21 Routing Design for Inbound calls



The following diagram shows the Q21 Routing Design for Outbound calls



Q21 Configuration

> **Note**
> This section may be modified based on customer specific requirements

VNET Configuration

A signaling VNET (virtual network) is the combination of a physical interface, a gateway IP address (router), and an optional VLAN (virtual local area network) ID.

```
cli vnet add v1
cli vnet edit v1 gateway 10.34.92.1 ifname eth2
cli vnet add v2
cli vnet edit v2 gateway 10.34.94.1 ifname eth3
```

> **Note**
> Configure the QSBC Media separately.

REALM Configuration

Realm is a logical service entry point for other devices to connect to the SBC. (A realm on the SBC is not synonymous with a SIP realm.)

For calls to connect successfully, you must have at least one realm configured on your system.

These refer to SIP Signaling IPs of QSBC towards various peers including SIP suppliers and SWe for Telstra.

```
cli realm add SYDPQ21LABVFNZREALM
cli realm edit SYDPQ21LABVFNZREALM vnet v1 rsa 10.34.92.166 mask 255.255.254.0 emr alwayson imr alwayson medpool 1

cli realm add GVPSYDPQFLEXLAN394REALM
cli realm edit GVPSYDPQFLEXLAN394REALM vnet v1 rsa 10.34.92.167 mask 255.255.254.0 emr alwayson imr alwayson
medpool 1

cli realm add GVPLABPMPLSGVOIPCUSTREALM
cli realm edit GVPLABPMPLSGVOIPCUSTREALM vnet v1 rsa 10.34.92.168 mask 255.255.254.0 emr alwayson imr alwayson
medpool 1

cli realm add LABGIA210REALM
cli realm edit LABGIA210REALM vnet v1 rsa 10.34.92.169 mask 255.255.254.0 emr alwayson imr alwayson medpool 1

cli realm add TCISYDLSWELABREALM
cli realm edit TCISYDLSWELABREALM vnet v2 rsa 10.34.94.166 mask 255.255.254.0 emr alwayson imr alwayson medpool 2

cli realm add GVPLABPMPLSGVOIPCUSTREALM2
cli realm edit GVPLABPMPLSGVOIPCUSTREALM2 vnet v2 rsa 10.34.94.167 mask 255.255.254.0 emr alwayson imr alwayson
medpool 2

cli realm add PSX
cli realm edit PSX vnet v2 rsa 10.34.94.168 mask 255.255.254.0 emr alwayson imr alwayson medpool 2
```

End Point Configuration

PSX EndPoints (EPs) to send INVITE & receive 300 response

```
cli iedge add SYDLPSXLAB01EP 0
cli iedge edit SYDLPSXLAB01EP 0 realm PSX static 10.54.181.13 sip enable type sipproxy dtg SLPSXLBPAD01 use4904tg
disable setdesttg disable  vendor generic

cli iedge add SYDLPSXLAB01EP 1
cli iedge edit SYDLPSXLAB01EP 1 realm PSX static 10.54.181.13 sip enable type sipproxy use4904tg disable setdesttg
disable vendor generic cp PSX01CP
```

SIP Supplier EPs

F1.1 EP for PSX dedicate Realm (Existing EP) REALM: 10.54.92.166

```
cli iedge add SYDPQ21LABVFNZEP 0
cli iedge edit SYDPQ21LABVFNZEP 0 realm SYDPQ21LABVFNZREALM static 10.54.81.11 sip enable type sipgw cp TEAMS01CP
newsrcdtg SLPSXLBPAD01 use4904tg disable vendor generic
cli iedge edit SYDPQ21LABVFNZEP 0 newsrcitg "SLVODAFNZL01"
```

B4-GW1 (Existing EP) REALM: 10.54.92.167 with 4904 disabled

```
cli iedge add SYDPQ21LABVFNZEP 1
cli iedge edit SYDPQ21LABVFNZEP 1 realm GVPSYDPQFLEXLAN394REALM static 10.54.81.11 contact 10.54.81.11:8012 sip
enable type sipgw dtg SLVODAFNZL01 tg SLVODAFNZL01 use4904tg disable vendor generic
```

F1.2 EP for Selection of Ingress via DNIS REALM: 10.54.92.168

```
cli iedge add SYDLSIPLAB02EP 0
cli iedge edit SYDLSIPLAB02EP 0 realm GVPLABPMPLSGVOIPCUSTREALM static 10.54.81.12 contact 10.54.81.12:8012 sip
enable type sipgw cp TEAMS02CP dtg SLVODAFNZL02 tg SLVODAFNZL02 use4904tg disable setdesttg enable vendor generic
```

F1.3 EP for Selection of Ingress via sourceTG=SLVODAFNZL03 (REALM: 10.54.92.168) with 4904 enabled

```
cli iedge add SYDLSIPLAB02EP 1
cli iedge edit SYDLSIPLAB02EP 1 realm GVPLABPMPLSGVOIPCUSTREALM static 10.54.81.12 contact 10.54.81.12:8014 sip enable type sipgw newsrcdtg SLPSXLBPAD01 dtg SLVODAFNZL03 tg SLVODAFNZL03 use4904tg enable vendor generic
```

B4-CSF2

```
cli iedge add SYDLSIPLAB02EP 2
cli iedge edit SYDLSIPLAB02EP 2 realm GVPLABPMPLSGVOIPCUSTREALM static 10.54.81.12 sip enable type sipgw dtg SLGVQ21PAD02 use4904tg disable vendor generic
```

B4-CSF1 (Existing EP)

```
cli iedge add GVMSYDPGIASYDLP01 1
cli iedge edit GVMSYDPGIASYDLP01 1 realm LABGIA210REALM static 10.54.81.13 sip enable type sipgw dtg SLGVQ21PAD01 use4904tg disable vendor generic
```

F4, B1 (EP for 3XX to LAB SWE1)

```
cli iedge add SYDLSWELAB01EP 0
cli iedge edit SYDLSWELAB01EP 0 realm PSX static 10.54.81.66 sip enable type sipgw newsrcdtg SLPSXLBPAD01 use4904tg enable vendor generic setdesttg enable URI 10.54.81.66
```

F4-B1 (EP for 3XX to LAB SWE2)

```
cli iedge add SYDLSWELAB02EP 0
cli iedge edit SYDLSWELAB02EP 0 realm TCISYDLSWELABREALM static 10.54.81.67 sip enable type sipgw newsrcdtg SLPSXLBPAD01 use4904tg enable vendor generic setdesttg enable URI 10.54.81.67
```

Sample Calling Plans

INGRESS FROM SUPPLIER CALLING PLAN

```
cli cr add TEAMS09980315INGCR
cli cr edit TEAMS09980315INGCR calltype origin dest 09980315 prefix 64*09980315
cli cp add TEAMS02CP TEAMS09980315INGCR
cli iedge edit SYDLSIPLAB02EP 0 cp TEAMS02CP
```

EGRESS TO PSX CALLING PLAN

(For 64* prefix numbers routed to PSX redirector)

```
cli cr add TEAMS09980315EGRCR
cli cr edit TEAMS09980315EGRCR calltype dest dest 64*09980315 prefix 0998031512345
cli cp add PSX01CP TEAMS09980315EGRCR
cli iedge edit SYDLPSXLAB01EP 1 cp PSX01CP
```

Link Calling Plan with MULTIPLE Calling Route

```
cli cr add TEAMS09980316INGCR
cli cr edit TEAMS09980316INGCR calltype origin dest 09980316 prefix 09980316
cli cp add TEAMS01CP TEAMS09980316INGCR

cli cr add TEAMS0998032INGCR
cli cr edit TEAMS0998032INGCR calltype origin dest 0998032 prefix 0998032
cli cp add TEAMS01CP TEAMS0998032INGCR

cli cr add TEAMS099803301INGCR
cli cr edit TEAMS099803301INGCR calltype origin dest 099803301 prefix 099803301
cli cp add TEAMS01CP TEAMS099803301INGCR

cli cr add TEAMS099803302INGCR
cli cr edit TEAMS099803302INGCR calltype origin dest 099803302 prefix 099803302
cli cp add TEAMS01CP TEAMS099803302INGCR

cli cr add TEAMS0998034INGCR
cli cr edit TEAMS0998034INGCR calltype origin dest 0998034 prefix 0998034
cli cp add TEAMS01CP TEAMS0998034INGCR
cli iedge edit SYDPQ21LABVFNZEP 0 cp TEAMS01CP
```

FMM Rules Manipulation

Use the Flexible Message Manipulation (FMM) feature to manipulate SIP messages and SIP message contents in QSBC.

FMM rules are required for Telstra to modify the digits in the "From" Header & "P-Asserted-Id" Header and to change the domain name in the From header to RSA (Realm Signaling IP Address)

See below for the FMM rules implemented for Telstra.

1. FMM rule for modifying From Header

Trigger to check FROM with 'user' part

```
cli fmm trigger add from-available-t sip-header
cli fmm trigger edit from-available-t method is("INVITE")
cli fmm trigger edit from-available-t msg.type is("request")
cli fmm trigger edit from-available-t header.name is("From")
cli fmm trigger edit from-available-t uri.userinfo.user is-phone-number() and not starts-with("+")
```

Trigger to check FROM header starting with 0 and having a size of 9-10 digit

```
cli fmm trigger add from-remove-zero-t sip-header
cli fmm trigger edit from-remove-zero-t method is("INVITE")
cli fmm trigger edit from-remove-zero-t msg.type is("request")
cli fmm trigger edit from-remove-zero-t header.name is("From")
cli fmm trigger edit from-remove-zero-t uri.userinfo.user starts-with("0") and is-match("^[0-9]{9,10}$")
```

Trigger to check FROM header starting with anonymous or unavailable

```
cli fmm trigger add from-anon-t sip-header
cli fmm trigger edit from-anon-t method is("INVITE")
cli fmm trigger edit from-anon-t msg.type is("request")
cli fmm trigger edit from-anon-t header.name is("From")
cli fmm trigger edit from-anon-t uri.userinfo.user (contains(ci("anonymous")) or contains(ci("unavailable")))
```

Trigger to check FROM header starting without 0 and having digit size 8-9

```
cli fmm trigger add from-not-start-zero-t sip-header
cli fmm trigger edit from-not-start-zero-t method is("INVITE")
cli fmm trigger edit from-not-start-zero-t msg.type is("request")
cli fmm trigger edit from-not-start-zero-t header.name is("From")
cli fmm trigger edit from-not-start-zero-t uri.userinfo.user not starts-with("0") and is-match("^[0-9]{8,9}$")
```

Action to replace 0 with +64

```
cli fmm action add from-remove-zero-a modify
cli fmm action edit from-remove-zero-a from-remove-zero-t.uri.userinfo.user substitute(from-remove-zero-t.uri.
userinfo.user,"^[0]","+64")
```

Action to change From to anonymous

```
cli fmm action add from-anon-a modify
cli fmm action edit from-anon-a from-anon-t.uri.userinfo.user "anonymous"
```

Action to add +64

```
cli fmm action add from-not-start-zero-a modify
cli fmm action edit from-not-start-zero-a from-not-start-zero-t.uri.userinfo.user "+64"+from-not-start-zero-t.uri.
userinfo.user
```

Action to add +

```
cli fmm action add from-available-a modify
cli fmm action edit from-available-a from-available-t.uri.userinfo.user "+"+from-available-t.uri.userinfo.user
```

Rule to replace 0 with +64

```
cli fmm rule add from-remove-zero-r
cli fmm rule edit from-remove-zero-r condition from-remove-zero-t actions from-remove-zero-a
```

Rule to change From to anonymous

```
cli fmm rule add from-anon-r
cli fmm rule edit from-anon-r condition from-anon-t actions from-anon-a
```

Rule to add +64

```
cli fmm rule add from-not-start-zero-r
cli fmm rule edit from-not-start-zero-r condition from-not-start-zero-t actions from-not-start-zero-a
```

Rule to add +

```
cli fmm rule add from-available-r
cli fmm rule edit from-available-r condition (from-available-t and (not from-remove-zero-t) and (not from-not-
start-zero-t) and (not from-anon-t))
cli fmm rule edit from-available-r actions from-available-a
```

Profile to normalize FROM header

```
cli fmm profile add from-remove-zero-p
cli fmm profile edit from-remove-zero-p rules from-remove-zero-r,from-anon-r,from-not-start-zero-r,from-available-r
```

2. FMM for modifying PAID header

Trigger to check PAID with user part

```
cli fmm trigger add paid-available-t sip-header
cli fmm trigger edit paid-available-t method is("INVITE")
cli fmm trigger edit paid-available-t msg.type is("request")
cli fmm trigger edit paid-available-t header.name is("P-Asserted-Identity")
cli fmm trigger edit paid-available-t uri.userinfo.user is-phone-number() and not starts-with("+")
```

Trigger to check PAID header starting with 0 and have size of 9-10 digit

```
cli fmm trigger add paid-remove-zero-t sip-header
cli fmm trigger edit paid-remove-zero-t method is("INVITE")
cli fmm trigger edit paid-remove-zero-t msg.type is("request")
cli fmm trigger edit paid-remove-zero-t header.name is("P-Asserted-Identity")
cli fmm trigger edit paid-remove-zero-t uri.userinfo.user starts-with("0") and is-match("^[0-9]{9,10}$")
```

Trigger to check PAID header starting with anonymous or unavailable

```
cli fmm trigger add paid-anon-t sip-header
cli fmm trigger edit paid-anon-t method is("INVITE")
cli fmm trigger edit paid-anon-t msg.type is("request")
cli fmm trigger edit paid-anon-t header.name is("P-Asserted-Identity")
cli fmm trigger edit paid-anon-t uri.userinfo.user (contains(ci("anonymous")) or contains(ci("unavailable")))
```

Trigger to check PAID header starting without 0 and having digit size 8-9

```
cli fmm trigger add paid-not-start-zero-t sip-header
cli fmm trigger edit paid-not-start-zero-t method is("INVITE")
cli fmm trigger edit paid-not-start-zero-t msg.type is("request")
cli fmm trigger edit paid-not-start-zero-t header.name is("P-Asserted-Identity")
cli fmm trigger edit paid-not-start-zero-t uri.userinfo.user not starts-with("0") and is-match("^[0-9]{8,9}$")
```

Action to replace 0 with +64

```
cli fmm action add paid-remove-zero-a modify
cli fmm action edit paid-remove-zero-a paid-remove-zero-t.uri.userinfo.user substitute(paid-remove-zero-t.uri.
userinfo.user,"^[0]","+64")
```

Action to change PAID to anonymous

```
cli fmm action add paid-anon-a modify
cli fmm action edit paid-anon-a paid-anon-t.uri.userinfo.user "anonymous"
```

Action to add +64

```
cli fmm action add paid-not-start-zero-a modify
cli fmm action edit paid-not-start-zero-a paid-not-start-zero-t.uri.userinfo.user "+64"+paid-not-start-zero-t.uri.
userinfo.user
```

Action to add +

```
cli fmm action add paid-available-a modify
cli fmm action edit paid-available-a paid-available-t.uri.userinfo.user "+"+paid-available-t.uri.userinfo.user
```

Rule to replace 0 with +64

```
cli fmm rule add paid-remove-zero-r
cli fmm rule edit paid-remove-zero-r condition paid-remove-zero-t actions paid-remove-zero-a
```

Rule to change PAID to anonymous

```
cli fmm rule add paid-anon-r
cli fmm rule edit paid-anon-r condition paid-anon-t actions paid-anon-a
```

Rule to add +64

```
cli fmm rule add paid-not-start-zero-r
cli fmm rule edit paid-not-start-zero-r condition paid-not-start-zero-t actions paid-not-start-zero-a
```

Rule to add +

```
cli fmm rule add paid-available-r
cli fmm rule edit paid-available-r condition (paid-available-t and (not paid-remove-zero-t) and (not paid-not-
start-zero-t) and (not paid-anon-t))
cli fmm rule edit paid-available-r actions paid-available-a
```

Profile to normalize PAID header

```
cli fmm profile add paid-remove-zero-p
cli fmm profile edit paid-remove-zero-p rules paid-remove-zero-r,paid-anon-r,paid-not-start-zero-r,paid-available-r
```

For importing the above fmm rule in Q21:

```
cli fmm import <fmm file name>
```

For applying the above FMM profile to endpoint:

```
cli iedge edit SYDPQ21LABVFNZEP 0 fmm-ingress-profile from-remove-zero-p,paid-remove-zero-p
```

3. FMM rule to change domain to RSA IP address in FROM header

The Q SBC will send FROM with domain uri, but PSX should receive RSA (Realm Signaling IP address).

```
cli fmm trigger add sbc-modify-request-t sip-header
cli fmm trigger edit sbc-modify-request-t method is("INVITE")
cli fmm trigger edit sbc-modify-request-t msg.type is("request")
cli fmm trigger edit sbc-modify-request-t header.name is(ci("From"))

cli fmm action add sbc-modify-request-a1 modify
cli fmm action edit sbc-modify-request-a1 sbc-modify-request-t.uri.hostport var.dst-realm.sbc-rsa

cli fmm rule add sbc-modify-request-r1
cli fmm rule edit sbc-modify-request-r1 condition sbc-modify-request-t
cli fmm rule edit sbc-modify-request-r1 actions sbc-modify-request-a1

cli fmm profile add sbc-modify-request-p1
cli fmm profile edit sbc-modify-request-p1 rules sbc-modify-request-r1
```

Apply above FMM profile to PSX endpoints

```
cli iedge edit SYDLPSXLAB01EP 0 fmm-egress-profile sbc-modify-request-p1
cli iedge edit SYDLPSXLAB01EP 1 fmm-egress-profile sbc-modify-request-p1
```

General Recommendation - Q21 Configuration

- Create as many EPs as required (as shown above) including EPs (type SIPGW) towards SIP Suppliers & Core SBC SWEs, and EPs (type SIPPROXY) towards PSX Redirector
- Create media related config on QSBC and enable generation of QOS parameters on QSBC CDRs with command **"nxconfig.pl -e mqm -v 1"**
- For ingress Trunk group based routing, refer to "42: Trunk group Support" in the release 9.4 QSBC Operations guide
- For sending TGRP, DTG, OTG parameters in the egress INVITE, refer to "42.7: Support for trunk group context and RFC 4904" and "42.4: Configuring trunk group endpoint options" in the release 9.4 QSBC Operations guide
- Refer to the QSBC Operations guide for any other specific or basic configuration

Section B: PSX Configuration

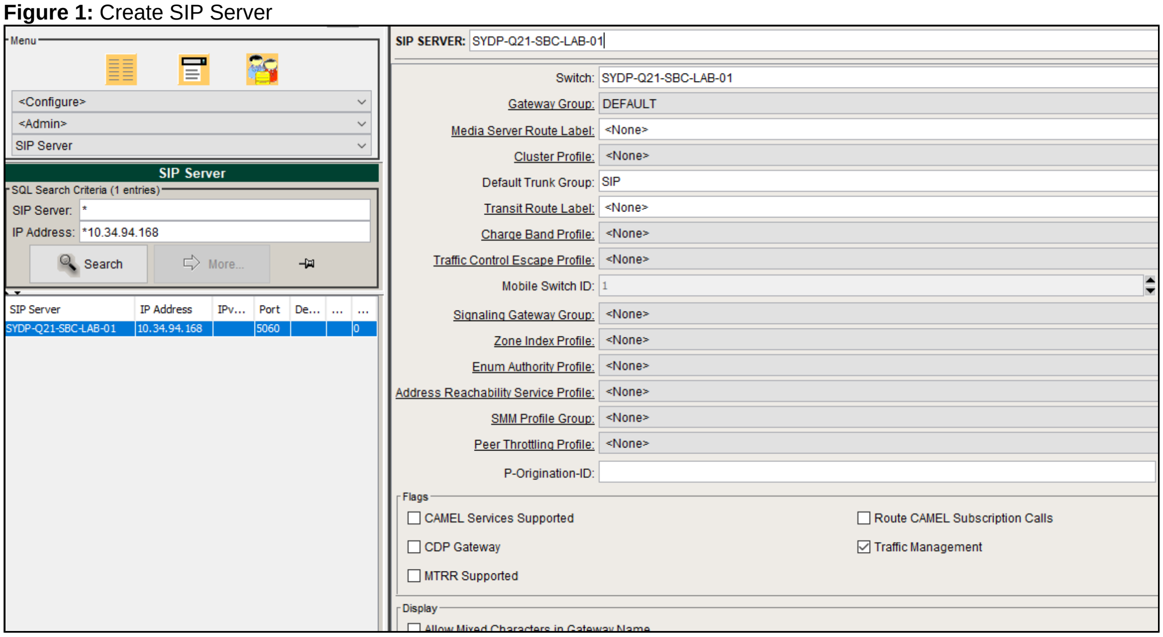
Redirector related configuration

- The PSX requires a SIP server entry created with QSBC SIP IP towards PSX side.
- The PSX requires a Trunk Group created with SIP server created from the above step. If this step is missed, the PSX picks the default SIP server entry "DEFAULTSIPSERVER".
- The PSX Trunk Group requires a Feature Control Profile with "IP Protocol Flags" enabled, "Support Domain Name in 300 Contact" enabled, and PSX processing mode set to "Redirector".
- If required, enable the following Feature control profile options: "Process TGRP", "Process Trunk-context", and "Process Originating Trunk Group And Trunk-context over OTG".
- The PSX Trunk Group requires an IP Signaling profile with the "Destination Trunk Group Options" set to "Include Tgrp with domain name" based on customer requirement.
- If required, the IP Signaling profile can have the "Originating Trunk Group Options" set to either "Include DTG" or "Include Tgrp with domain name" based on customer requirement.
- The PSX can have any kind of routing including standard routing based on trunk group or destination number with a Routing Label containing all the routes. Each route should have a Destination "SIP server" and "Trunk Group". The SIP server can have FQDN configured or IP address as per the requirement. It is recommended to not resolve FQDN by PSX and pass FQDN as it is due to restriction in the number of routes being capped at 10.
- The PSX Ingress Trunk Group requires the "Use IPTG Routing" option be enabled.
- PSX can return up to 10 Routes only in the 300 Response when acting as a redirector.

Creating SIP Server

You must configure the Q21 SIP Signaling IP address & Port in the SIP Server, through which the Q21 sends INVITE requests.

Figure 1: Create SIP Server



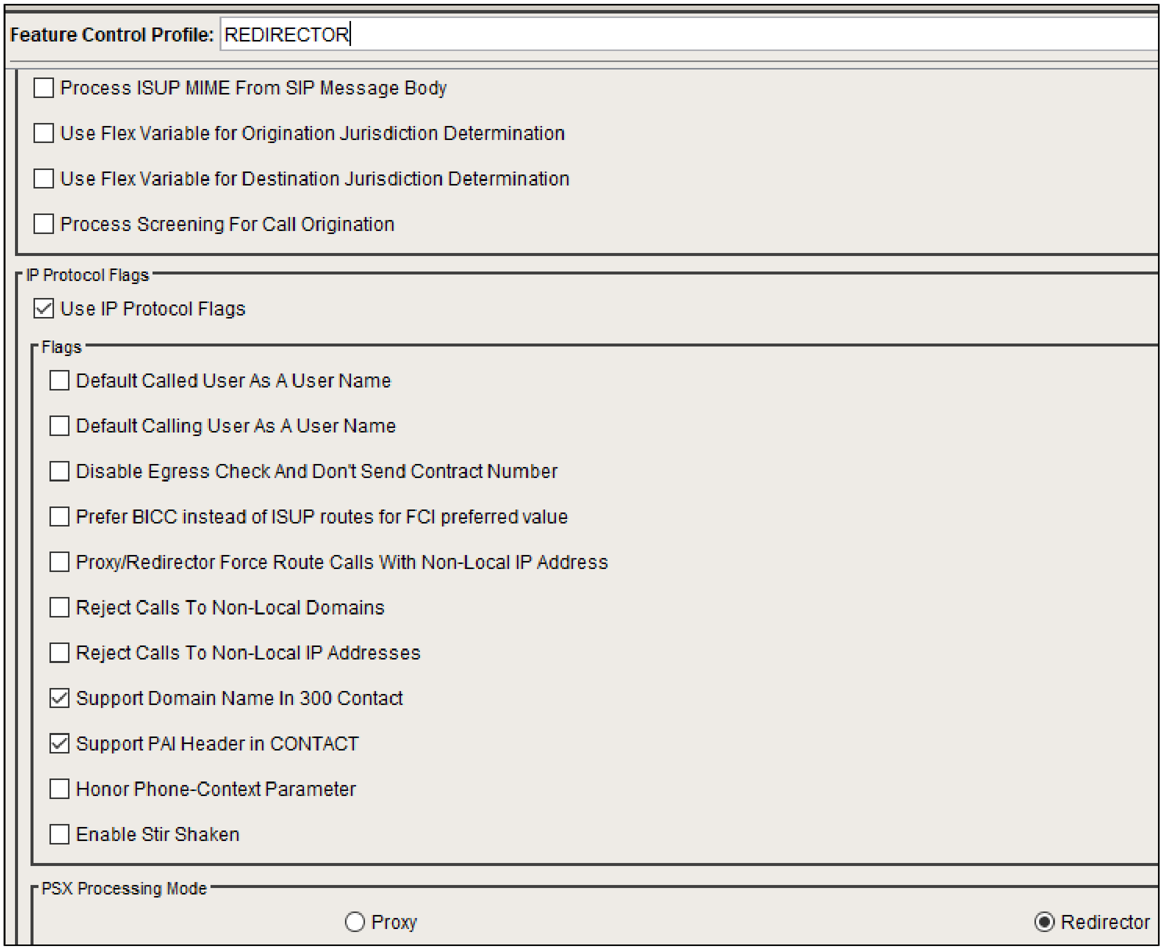
Creating Feature Control Profile

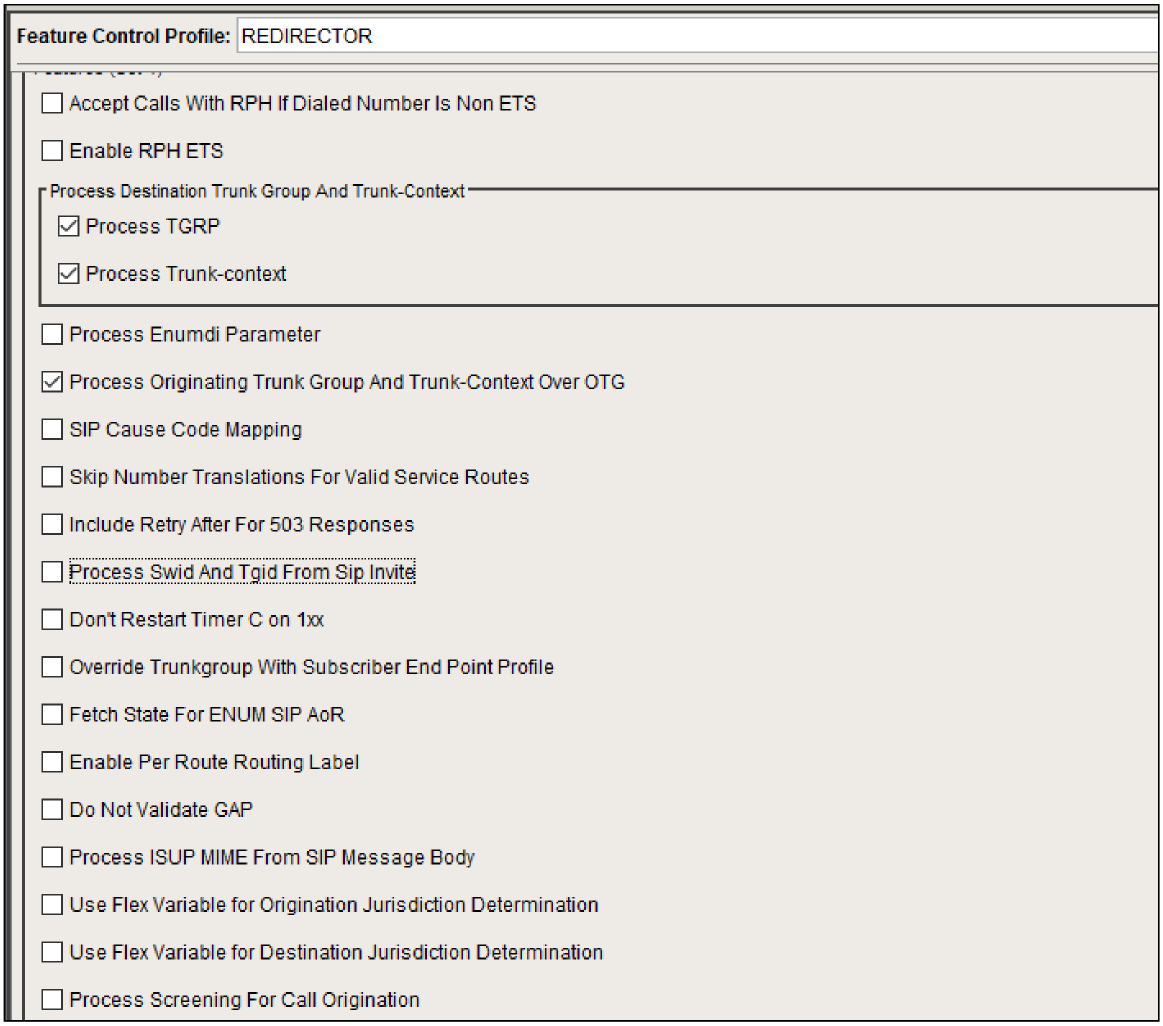
The Feature Control Profile determines whether the PSX acts as Redirector or Proxy for a particular call.

- In the below snapshot, the "Use IP Protocol Flags" is enabled and PSX processing mode is chosen as Redirector, so PSX acts as Redirector
- "Support Domain name in 300" will send the FQDN as it is instead of doing a DNS query for the FQDN. This is recommended so PSX can send up to 10 FQDNs.
- "Support PAI Header in Contact" enables sending embedded PAI in the contact header of a 300 response

Please refer to the Policy Server (PSX) documentation for details on each of the required flags.

Figure 2: Create Feature Control Profile

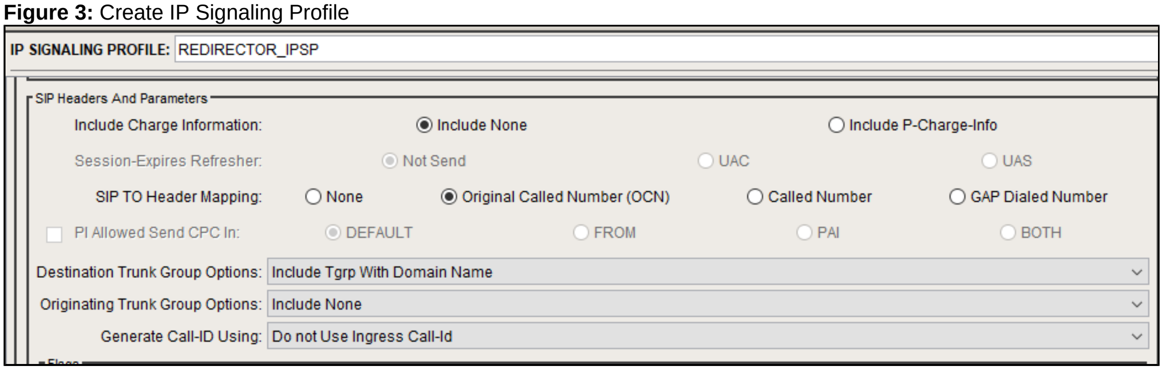




Creating IP signaling profile

Setting Destination Trunk Group option to "Include Tgrp with domain name" will send Trunk Group and Trunk-context values in the contact header of the 300 response.

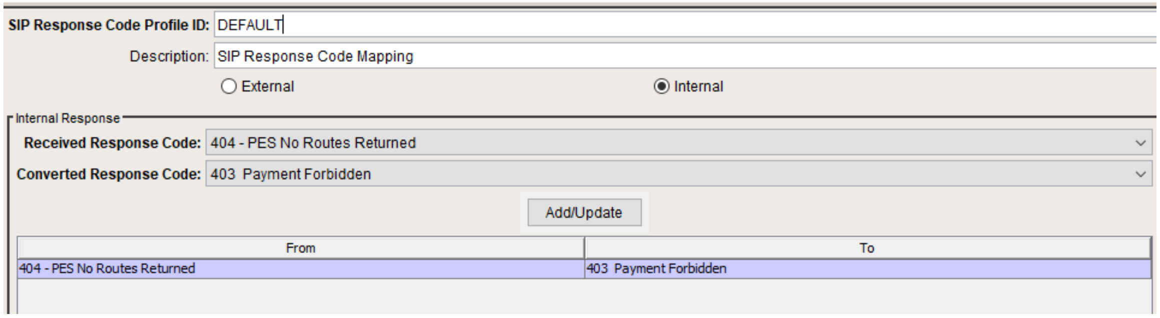
Figure 3: Create IP Signaling Profile



Creating SIP Response Code Profile

This profile can be used to change the SIP response code to the desired response code if the default PSX response code is not accepted.

Figure 4: Create SIP Response Code Profile

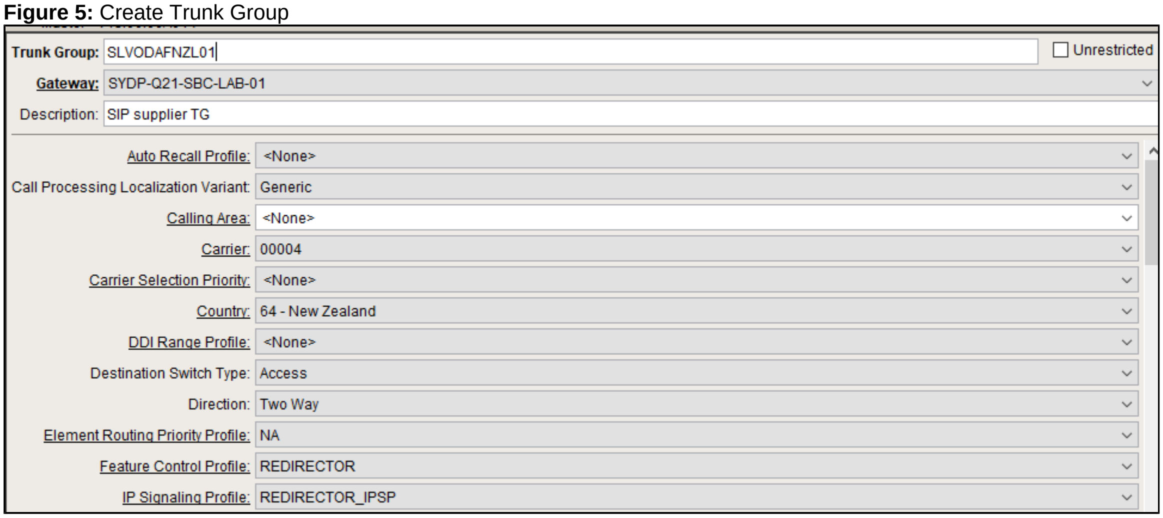


Creating Trunk Group

Create a Trunk Group and attach a Feature Control Profile, and IP Signaling Profile to it. Optionally, you can also attach the SIP Response Code Profile to the Trunk Group.

If there is no SIP Response Code profile attached to a specific Trunk Group, the DEFAULT profile will be used.

Figure 5: Create Trunk Group

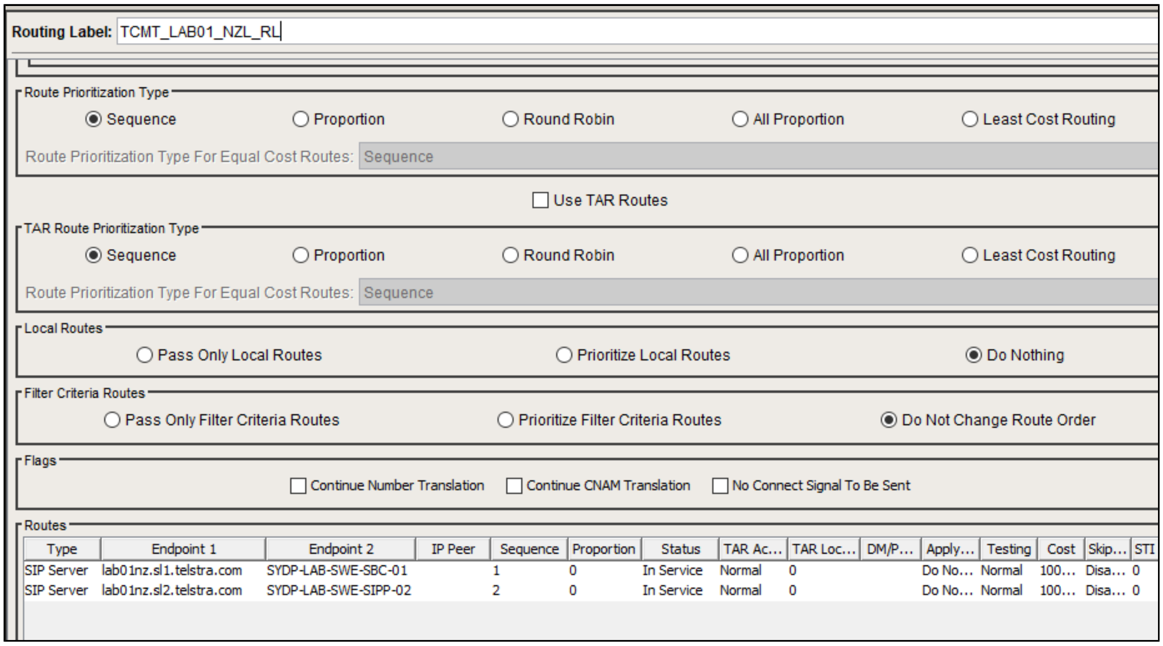


Creating Routing Profile

Create the Routing Profile and attach as many routes as required. Each Route will have a SIP Server and Trunk Group.

> ⓘ Up to 10 Routes can be passed in a 300 response in Redirector mode

Figure 6: Create Routing Label

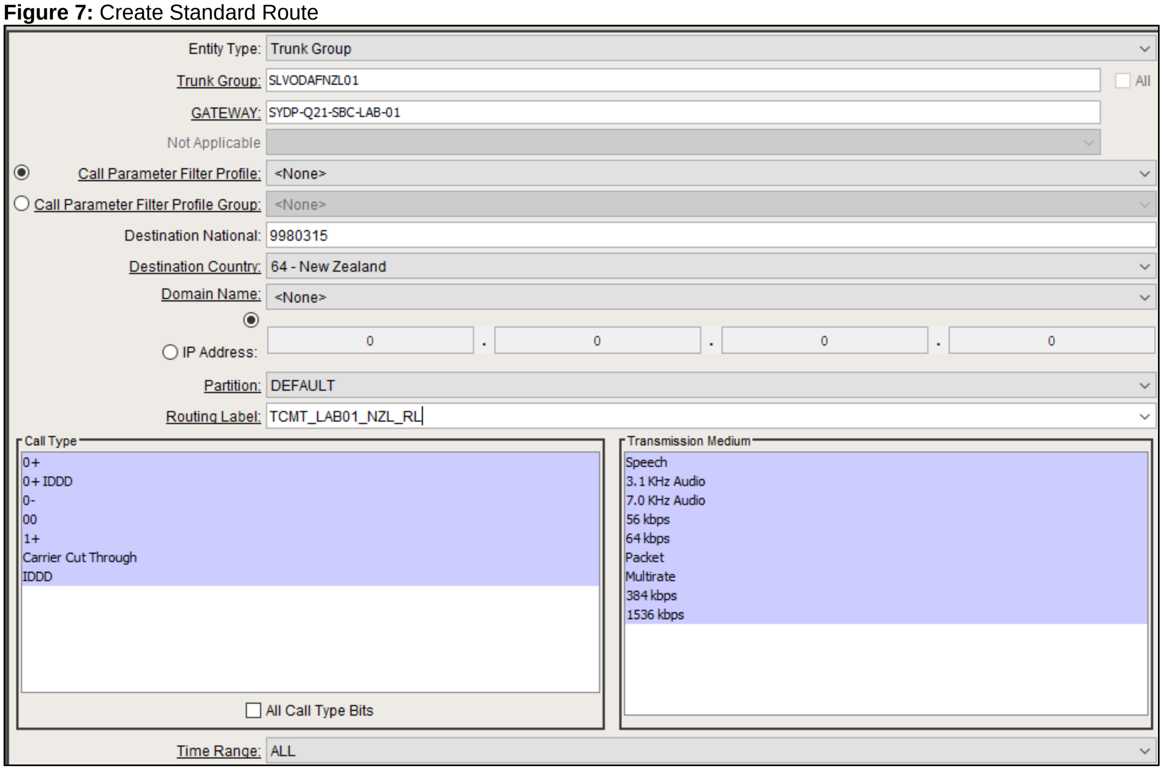


Creating Standard Route

Create the Standard Route and attach a Routing Label to it.

> The sample below shows Trunk Group based Routing on PSX.
>
> There are various ways of routing in PSX such as username based routing, number based routing, etc. depending on customer requirement.
> For more information, refer to the Policy Server (PSX) Documentation.

Figure 7: Create Standard Route



Conclusion

These Application Notes describe the configuration steps required for QSBC interop with PSX Redirector. All features and serviceability test cases were completed.