SIP Redirect Routing

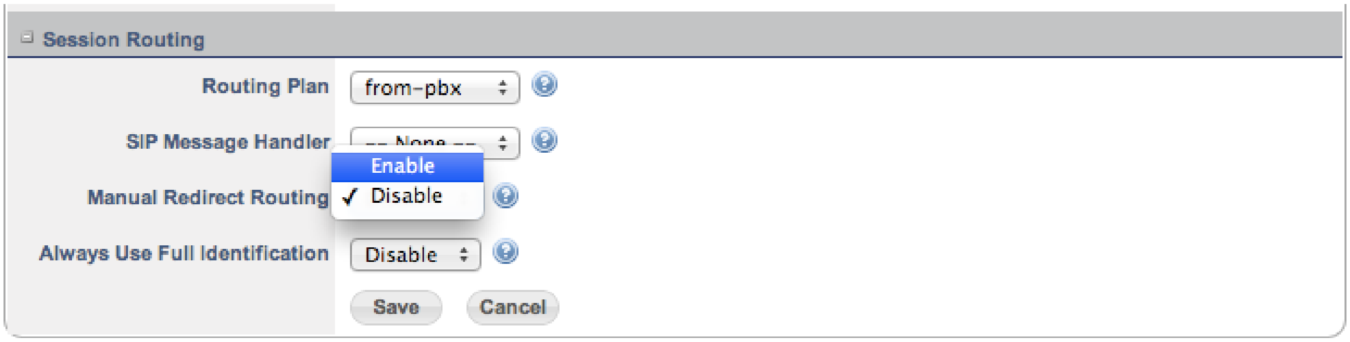
his section covers custom control of SIP redirect handling.

Redirect handling can be set to "Auto" or "Manual" mode.

- **Auto mode** - SIP 3XX responses are handled automatically based on information provided in the SIP 3XX response.
- **Manual mode** - SIP 3XX responses trigger a call routing lookup, executing custom routing logic under your control (the same way you are able to route INVITE requests).

To change the redirect handling mode of a SIP Profile:

- Go to **"Configuration -> Signaling -> SIP Profiles"**, modify and edit a SIP Profile.
- Under **"Session Routing"**, toggle **"Manual Redirect Routing"** to **"Disable"** for "Auto" mode and **"Enable"** for "Manual" mode.



Session Routing

Routing Plan: from-pbx

SIP Message Handler: -- None --

Manual Redirect Routing: Enable / ✓ Disable

Always Use Full Identification: Disable

In Manual mode. We must specify a "redirected" dialplan extension as follow:

Dialplan configuration

```
<extension name="outbound-route">
  <condition field="${destination number} expression="(.*)">
    <action application="export" data="sip_redirect_context=user-dialpan_name"/>
    <action application="bridge" data="sip/trunk/trunk1/$1"/>
  </condition>
</extension>

<extension name="redirected">
  <condition>
     <action application="bridge" data="sofia/trunk/trunk2/${sip_redirect_contact_0}"/>
  </condition>
</extension>
```

In the above example**, "sip_redirect_context"** Channel Variable must be set before the bridge application. This instructs the system which dialplan to look for the redirect instruction. The "**user-**" is a prefix which must be there literally, followed by the name of the dialplan. In the above, this dialplan must be named "dialplan_name". Once this is specified, when a SIP 3XX Redirect is received when the call is bridging, an extension named **"redirected"** in dialplan "dialplan_name" will be used for the redirection. Here, the bridge application uses a different trunk to establish the new leg.

Customization

In the above example, we used input Channel Variable **"sip_redirect_context"** to specify the dialplan name. Using the output Channel Variable **"sip_redirect_contact_0"** to retrieve the first redirect address. The following is a list of output Channel Variables that can be used within the **"redirected"** extension:

- **sip_redirect_contact_<index>** - After receiving a 3xx SIP message, this variable will be set with the contact header of each Contact header on the received message. <index> ranging from 0 to the number of contacts in the message.
- **sip_redirect_contact_host_<index>** - After receiving a 3xx SIP message, this variable will be set with the contact host of each Contact header on the received message. <index> ranging from 0 to the number of contacts in the message.
- **sip_redirect_contact_params_<index>** - After receiving a 3xx SIP message, this variable will be set with the contact params of each Contact header on the received message. <index> ranging from 0 to the number of contacts in the message.
- **sip_redirect_dialstring_<index>** - After receiving a 3xx SIP message, this variable will be set with the dialstring formed from **each** Contact header contained on the message. <index> ranging from 0 to the number of contacts in the message.
- **sip_redirect_dialstring** - After receiving a 3xx SIP message, this variable will be set with the dialstring formed from **all** Contact header contained on the message.
- **sip_redirected_to** - This variable is populated with the contact to which the call was bridged to.

The following further modify the "redirected" extension to log the value of each of the above output Channel Variables.

```
<extension name="redirected">
  <condition>
     <action application="log" data="The first redirect contact is ${sip_redirect_contact_0}"/>
     <action application="log" data="The first redirect contact host is ${sip_redirect_contact_host_0}"/>
     <action application="log" data="SIP redirect parameters in the first redirect contact are ${sip_redirect_contact_params_0}"/>
     <action application="log" data="The first redirect dialstring is ${sip_redirect_dialstring_0}"/>
     <action application="log" data="All dialstrings ${sip_redirect_dialstring}"/>
     <action application="log" data="To Header value ${sip_redirected_to}"/>
     <action application="bridge" data="sofia/trunk/trunk2/${sip_redirect_contact_0}"/>
  </condition>
</extension>
```

If we receive the below SIP 302, the above dialplan extension will have the following output:

```
SIP/2.0 302 Temporarily Moved
Via: SIP/2.0/UDP 192.168.10.10;rport;branch=z9hG4bK2mNt7QyUDc5Br
From: "sipp" <sip:sipp@192.168.10.10>;tag=a8eym43p180FK
To: <sip:1029@192.168.10.102:5090>;tag=32210SIPpTag011
Call-ID: 383a9dac-6e47-1231-9ca8-0800272926ff
CSeq: 46958840 INVITE
Contact: <sip:222@192.168.10.102:6600;transport=UDP>
Content-Length: 0
```

Log output:

```
The first redirect contact is <sip:222@192.168.10.102:6600;transport=UDP>

The first redirect contact host is 192.168.10.102

SIP redirect parameters in the first redirect contact are transport=UDP

The first redirect dialstring is sofia/trunk/trunk1/sip:222@192.168.10.102:6600;transport=UDP

All dialstrings sofia/trunk/trunk1/sip:222@192.168.10.102:6600;transport=UDP

To Header value <sip:222@192.168.10.102:6600;transport=UDP>
```