How to install elastix-1.6 with OpenVox B200P_mISDN

Notes:

Test environments:

Elastix version: (Elastix Distro 1.6 Stable)

Kernel version: 2.6.18-164.el5

Hardware: OpenVox B200P

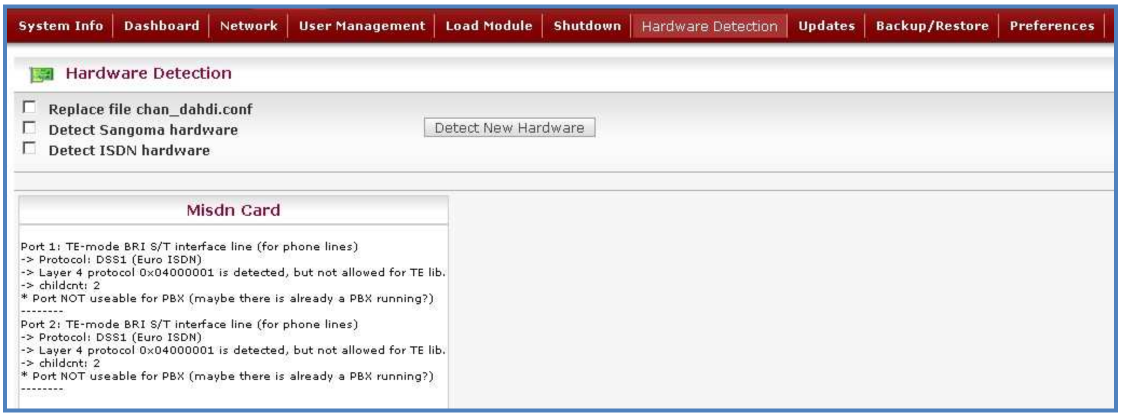
mISDN version: 1_1_9.1

Before you want to refer this manual to install in your case, make sure your system can access the Internet.

Step 1: Access to the elastix GUI via your IP. Almost all of the settings can finish in there.

Step 2: Log in the system with username: admin, secret: palosanto.

Step 3: Follow this order please: System→Hardware Detection, select "Detect ISDN hardware" in the selection lists .And click the button "Detect New Hardware" on it right side. It would be presented more details of the card like this:

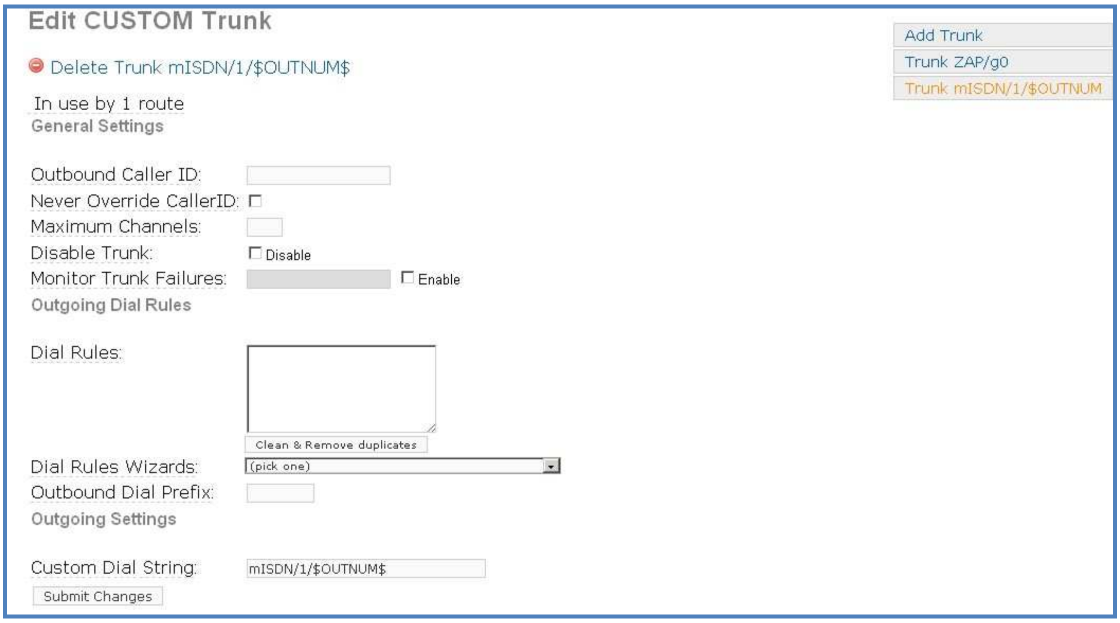


From Misdn Card box we can get some information about the card.

Step 4: Add an extension in the system, you should follow this sequence: PBX→PBX Configuration→Extensions. Here, there is an optional list. You can select the Device type in there which you want to. Like as "Generic SIP Device", "Generic ZAP Device", "Generic IAX2 Device", "other (custom) Device" and so on . In my case, I just select "Generic SIP Device", and click "Submit". Then, add the values of the text box, such as "User Extension", "Display Name", "CID number Alias", "SIP Alias" and " secret". Click "Submit" after that. At this time, the system will be presented an option to click like this:

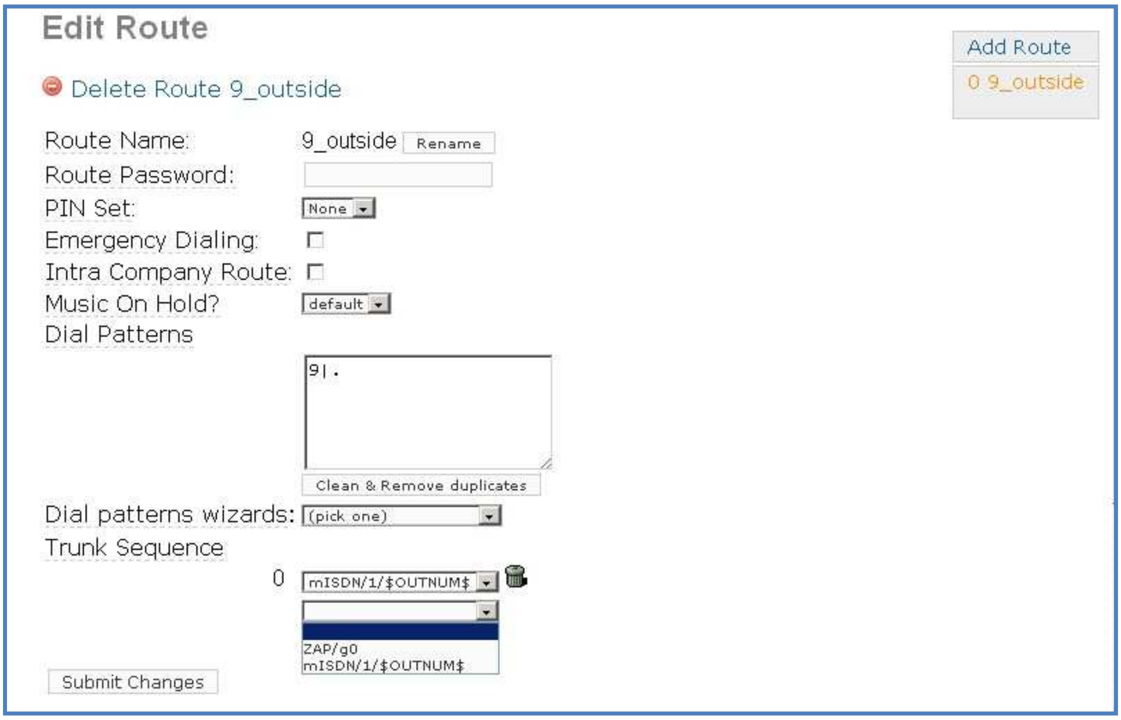
Click it remember! Now, you have finished the task of adding extension. You can add other extensions with this way.

Step 5: If want to add a trunk in this system. Take this way: PBX→PBX Configuration→Trunks. In there, it lists some selections to select. "Add Zap Trunk (DAHDI Compatibility Mode)", "Add IAX2 Trunk", "Add SIP Trunk", "Add ENUM Trunk", "Add DUNDI Trunk", "Add Custom Trunk". We should select the "Add Custom Trunk" one. Then, we can see this page:



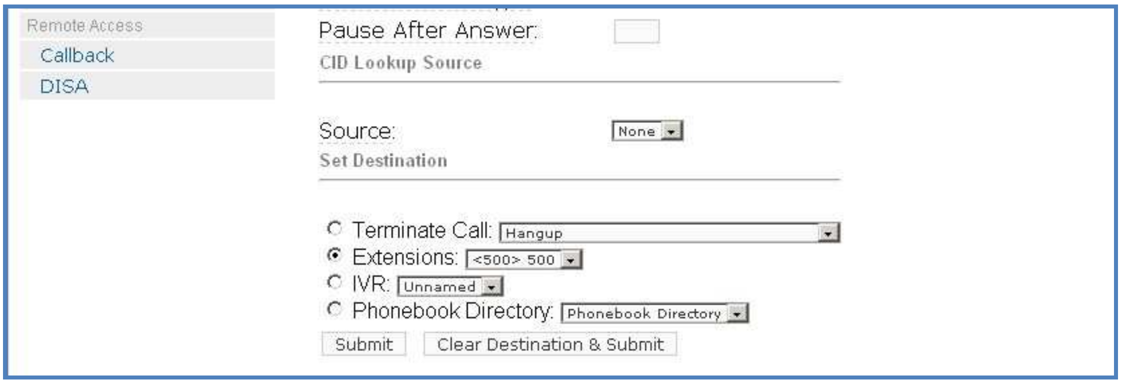
Here, there is a very important value to add. Look at the picture, we can see the Custom Dial String text box. In this box, the default value is:"mISDN /g:isdn/$OUTNUM$".But you should according to your realistic case to write. Maybe there are some differences in difference cases. In my case, I just keep it default. Click the button "Submit Changes", and do as the same way which I mentioned before.

Step 6: Add an outbound Routes in the system is very simple. Go along like this please. PBX→PBX Configuration→Outbound Routes. It will be presented this page:



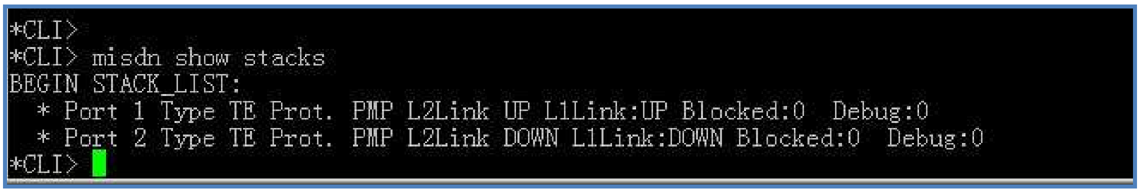
In this picture, we can see a default value of the Route. That is "0 9_outside". You can try to use in your system, but you must select the value in the "Trunk Sequence" selection list. In my system, I keep the default value in the Dial Patterns, and select the Trunk Sequence is "mISDN/1/$OUTNUM$" which I defined before. Actually, this value is decided by yourself. So, in here, you have to pay more attention to your case.

Step 7: Add an inbound Route. PBX→PBX Configuration→Inbound Routes.



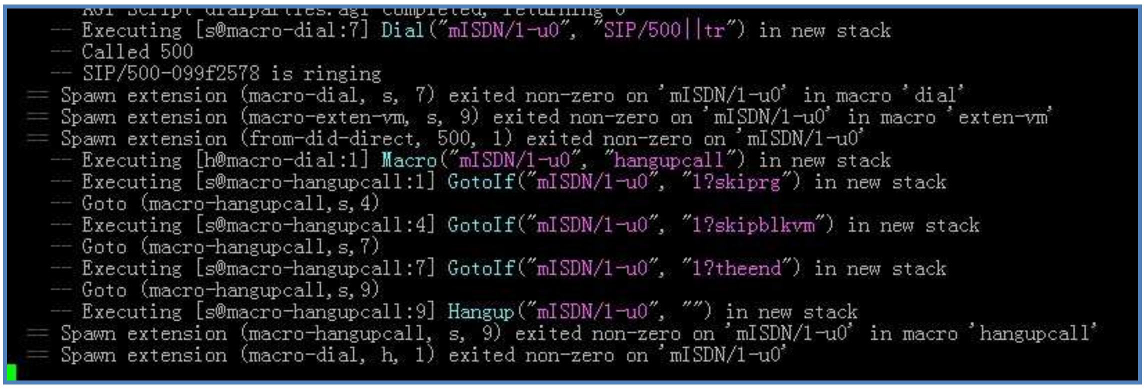
Here, you must select the extension which you added before. In my case, I add an extension 500(SIP Device). Click the button "Submit" remember.

Step 8: If you finish all of the steps I mention above, then login into your system with text mode, and start asterisk. In the asterisk console, run the command misdn show stacks, we can get this:



Here, the system will show the ports info of the cards.

Step 9: Dial the extension number which you add before, in my case, I dial the sip number 500 with my mobile phone, and it shows as:



If it shows like this, which means it runs correctly. In the picture, we can read about the info and status of the phone when I dialing. Generally, we can get lots of information from here, and it would give us some advices, tips and so on.

Step 9: I change another way, I dial the number of the telephone company via my extension (SIP), and it shows:

```
*CLI>
*CLI>
*CLI>     -- Executing [10000@from-internal:1] Dial("SIP/500-099f24a8", "mISDN/1/10000") in new stack
    -- Called 1/10000
    -- mISDN/2-u4 is proceeding passing it to SIP/500-099f24a8
    -- mISDN/2-u5 is ringing
    -- mISDN/2-u5 answered SIP/500-099f24a8
    -- Executing [h@from-internal:1] Macro("SIP/500-099f24a8", "hangupcall") in new stack
    -- Executing [s@macro-hangupcall:1] GotoIf("SIP/500-099f24a8", "1?skiprg") in new stack
    -- Goto (macro-hangupcall,s,4)
    -- Executing [s@macro-hangupcall:4] GotoIf("SIP/500-099f24a8", "1?skipblkvm") in new stack
    -- Goto (macro-hangupcall,s,7)
    -- Executing [s@macro-hangupcall:7] GotoIf("SIP/500-099f24a8", "1?theend") in new stack
    -- Goto (macro-hangupcall,s,9)
    -- Executing [s@macro-hangupcall:9] Hangup("SIP/500-099f24a8", "") in new stack
  == Spawn extension (macro-hangupcall, s, 9) exited non-zero on 'SIP/500-099f24a8' in macro 'hangupcall'
  == Spawn h extension (from-internal, h, 1) exited non-zero on 'SIP/500-099f24a8'
  == Spawn extension (from-internal, 10000, 1) exited non-zero on 'SIP/500-099f24a8'
    -- Executing [h@from-internal:1] Macro("SIP/500-099f24a8", "hangupcall") in new stack
    -- Executing [s@macro-hangupcall:1] GotoIf("SIP/500-099f24a8", "1?skiprg") in new stack
    -- Goto (macro-hangupcall,s,4)
    -- Executing [s@macro-hangupcall:4] GotoIf("SIP/500-099f24a8", "1?skipblkvm") in new stack
    -- Goto (macro-hangupcall,s,7)
    -- Executing [s@macro-hangupcall:7] GotoIf("SIP/500-099f24a8", "1?theend") in new stack
    -- Goto (macro-hangupcall,s,9)
    -- Executing [s@macro-hangupcall:9] Hangup("SIP/500-099f24a8", "") in new stack
  == Spawn extension (macro-hangupcall, s, 9) exited non-zero on 'SIP/500-099f24a8' in macro 'hangupcall'
  == Spawn extension (from-internal, h, 1) exited non-zero on 'SIP/500-099f24a8'
```

If you get the info like this, which means the setting is right.