Nortel NorStar MICS to Asterisk Connection

by:

David Gomillion

rev 0.2

Table of Contents

1. Purpose

2. Questions

3. General Description of Plan

4. Implementation of Plan
   1. Selection of DIDs
   2. Programming of DIDs
   3. Making of Crossover
   4. Configuration of TE410P
   5. Configuring the Dialplan

5. Testing

1. Purpose:

I have designed this document to explain the sequence of tasks we performed to tie our Nortel PBX to an Asterisk server. This is being furnished to the Asterisk community in hopes that it will assist all of the users that need to connect an Asterisk system with a Nortel system.

This document is being written during most of the configuration steps. Thus, some errors may exist that will be debugged later. Thus, revision numbers are being placed on the cover page. Updates may be found at http://www.eyecarenow.com/asterisk.

I am sharing this in the spirit of Open Source. This document may be changed, updated, etc. I do not make any money off of it, and do not want to. The caveat is that you cannot take this document, change it a little, then charge people for it. No, I am not putting it under a license of some sort. You are on your honor. Let's keep the lawyer out of it.

THIS IS NOT A COOKBOOK. Please do not complain that it didn't tell you exactly how to install Linux. It wasn't supposed to. There are plenty of resources out there for that purpose. This is designed to give a little insight into how I solved the problem of making the two phone systems play nice.

2. Questions:

I am more than willing to answer some questions about this procedure. The best way to reach me is via email at pbxintegration@eyecarenow.com. I usually answer emails in a few hours, but may take a couple of days to answer if the question requires thought.

Answering questions is free, but if you need more than that, I am available to offer consulting services. It is expected that hiring me will save you many hours of time and frustration, as I have already experienced many of the pitfalls associated with this tie. I assure you, my rates are reasonable and will be repaid many times over by saved costs and irritation.

But this document will probably give you more than enough information to get started. It is expected that anyone with a little Linux experience will be able to successfully complete this minor procedure. A little phone experience can also help a bit, although I can attest to the fact that it is not required.

3. General Description of Plan

Conceptually, what I am doing is so simple, it seems like there should not be a document written about this at all. I agree. But, for those who don't want to reinvent the wheel, or who don't know where to start, start here!

I walked into this company a few months ago. They have a Norstar MICS (bigger than a CICS, slightly different from a Meridian). The MICS doesn't have functioning queues, has problems with the ACD, AA, and Voicemail. Many consultants have come through before I was hired, and the company has sunk between 10,000 and 30,000 each trip of consultants into this system to make it workable. What they have now is a large phone system, with 400 extensions (about 100 used), 2 PRI trunks (usually only about 15 lines in use), and 40 copper lines coming in that nobody uses, has the numbers for, but have been paying for every month since the phone system was put in, which is only 4 years ago.

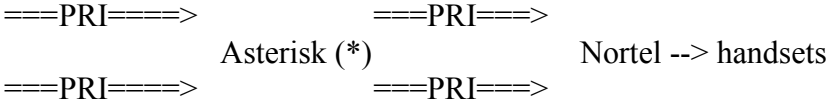
The problems they were experiencing is that phone calls would get dropped, and nobody knew why. The caller would be routing through the ACD correctly, press a valid option, then get hung up on. As we are a service-oriented business, this was far from acceptable. Also, we have been experiencing about 2 hours of downtime per month. This is because the company changed phone company, and when they did so, no maintenance agreement was available with the new telco.

Also, the Norstar has no cost accounting and call auditing system available. That is why we didn't know when calls were getting dropped or why. Process changes for our support personnel have been less successful, as there is no accountability.

Enter Asterisk. We saw a great way to add cost accounting, call monitoring/recording, and replace the call distribution module, with voicemail and queues to boot. The cost is very attractive, and since the company now has an in-house IT person with Linux administration and programming experience, the fit seemed to be a good one.

However, the 100+ handsets that are currently deployed work well. The phones are only a few years old, and the Board of Directors was hesitant to scrap the old system and perform a “forklift upgrade.” The requirement was to find a way to leverage our old handsets while giving an upgrade to the call distribution logic.

As Asterisk cannot yet support the Norstar handsets, we had very little choice. We needed a good bridge between our legacy PBX and Asterisk. Since we have 2 PRIs coming in, and the appropriate hardware on our legacy PBX, this solution seemed best. So our general layout is:



The direction of the arrows is, of course, arbitrary. Calls will flow both ways. In our business we are more concerned about incoming calls, so this model best accentuated the handling of those important calls.

What this gives us is the opportunity to use the legacy PBX to connect to the handsets and the * server to handle all of the call routing and distribution. This also allows us to phase out the legacy PBX. We plan to move to Voice over IP (VoIP) in the near future. The interconnection is only phase 1 of our migration.

4. Implementation of Plan

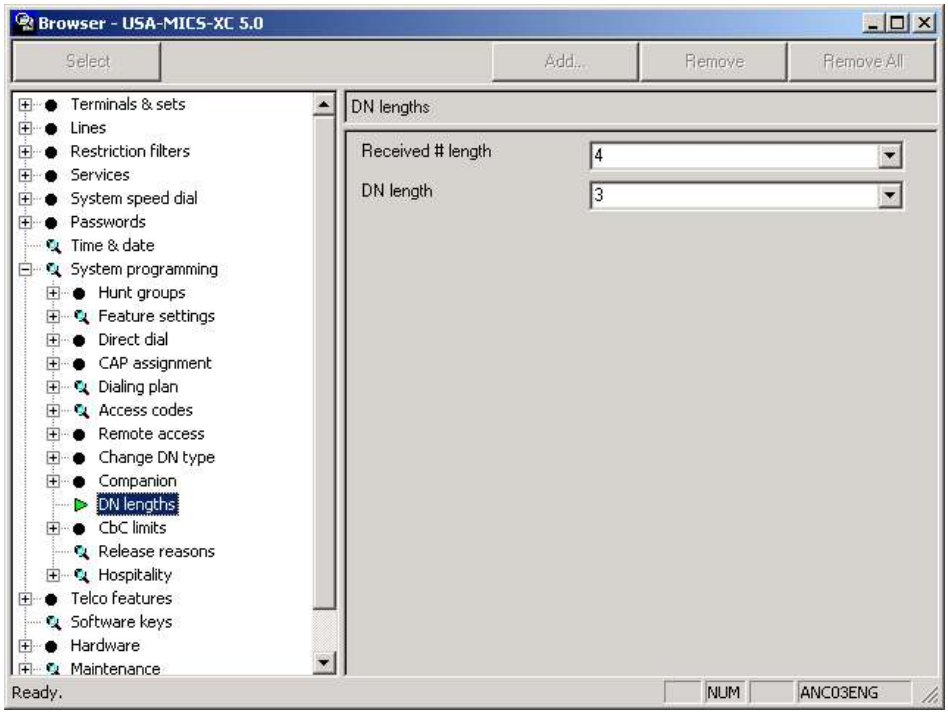
Here I will go through many of the steps needed to implement the plan to connect the MICS with the new * server. I will be using screen shots done in a Windows environment. I know that all of the steps can be completed on a handset using F**CONFIG. In fact, that's how it was completed; however, it is difficult to show a picture of that :).

HINT: F9** will allow you to set the password of a RAD or FASTRAD.

4.1. Selecting DIDs

It was determined that the twain would be linked via a pair of PRIs. This is great, but how do you tell the legacy PBX which extension to send a particular phone call to? We accommodated this by defining a set of bogus DIDs.

First, we looked at our phone system to determine how many digits it was receiving from the telco. We did this by using NRU and this screen:



As you can see, we have 4 digits coming in from the telco. This means that the last four digits of the phone number are the only ones that our legacy PBX has to work with. Our extensions are 3 digits long.

Our list of about 300 DIDs (yes, more DIDs than handsets...) included:

...
1954
1960
1965
2015
2673
3005
3110
...

As you can see, the only numbers in the 2XXX range are 2015 and 2673. Since there cannot be an extension 2015, and the extension 673 is not in use, the 2000 range was perfect for our mapping.

I then went through and counted the number of free trunks. There were about 160 free DID trunks that I could use to program in DIDs.

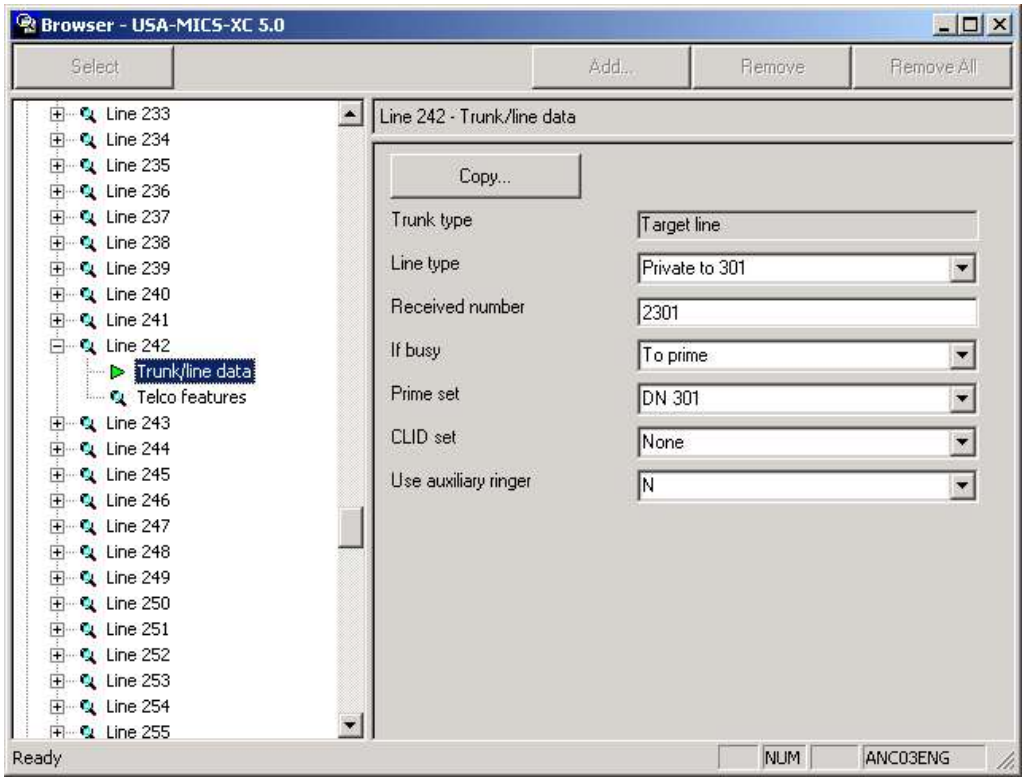
So, the final decision was to append a 2 to the front of the extension number of each handset on the old system. All of the new phone system extensions would have 4 digits. Therefore, any extension that begins with 2 is on the old system, while any extension beginning with a 1 is in the main office, and any extension beginning with an 8 is a service of the new PBX (Voicemail, call parking, conferences, MusicOnHold, etc.).

4.2. Programming of DIDs

The time came to program all of the new DIDs. This was not fun, but it was also not difficult. I simply created a DID for each extension. For example:

Something important to note: I did not change any DID that came in that was valid. This was so that, in the event of a problem with Asterisk, I could simply unplug the T1s from * and put them back into the Nortel. This gave me the peace of mind that we could at least go back to how things used to be...

(Graphic on next page)



4.3. Making of Crossover

Pursuant to much googling, I discovered that I needed a T1 crossover cable. This was made using standard Category 5 UTP. Wire 1 was switched with 4 and wire 2 was switched with 5. Creating patch cables is outside the scope of this document. It is sufficient to say this: 1 to 4 and 2 to 5, 5 to 2 and 4 to 1.

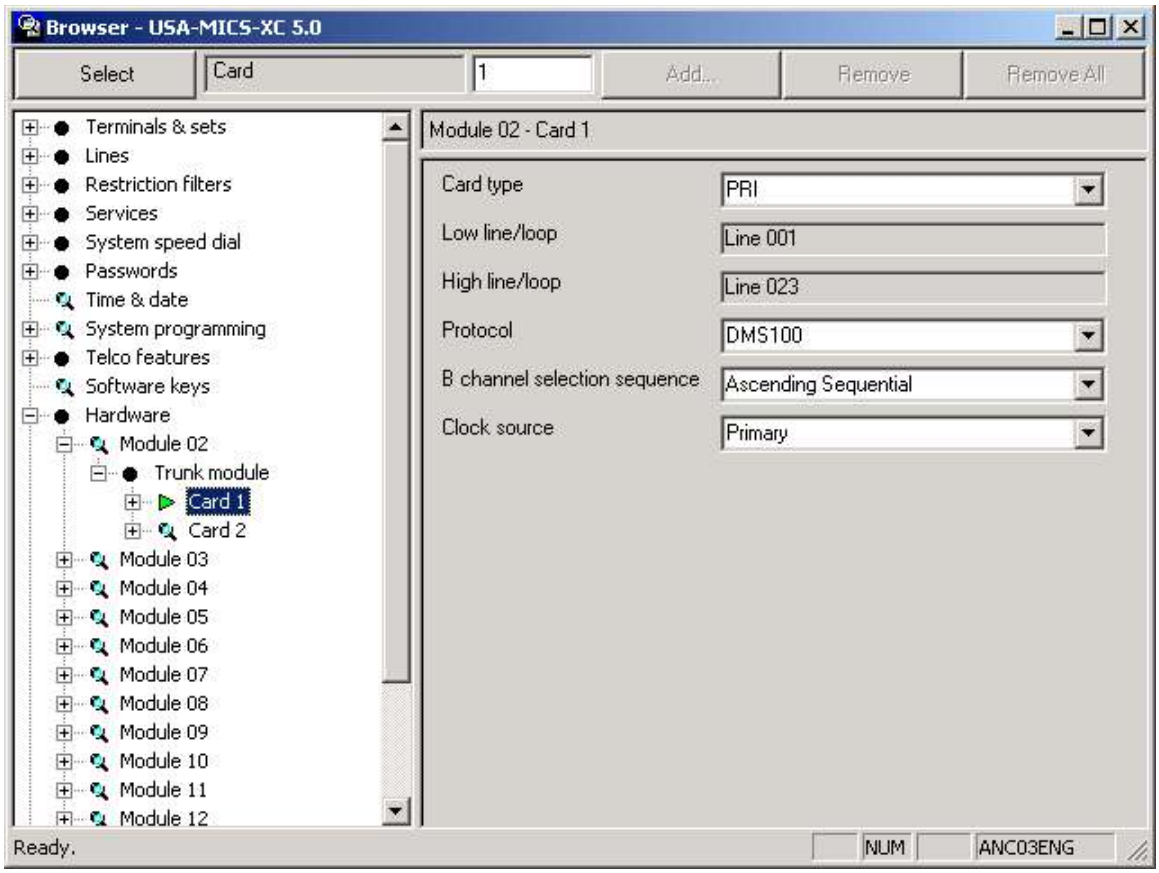
4.4. Configuring of TE410P

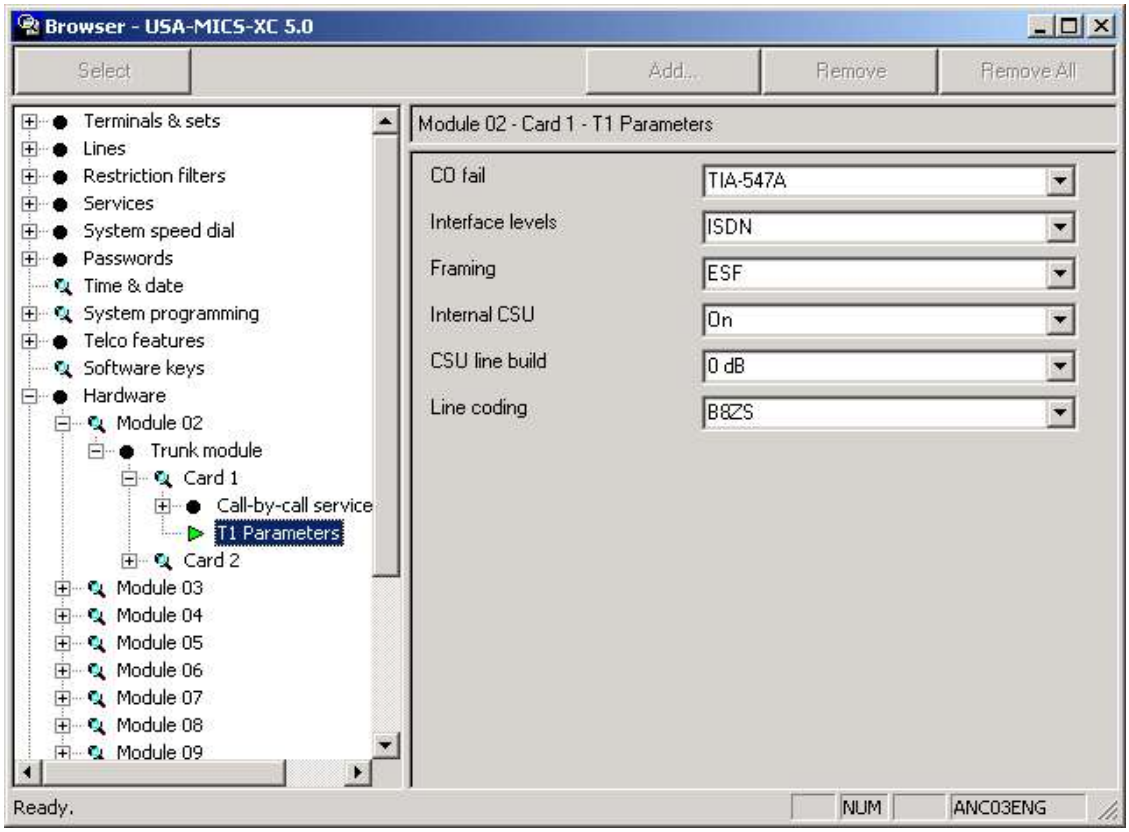
I selected the new TE410P from Digium to provide the four needed PRI interfaces. While I was hoping to hold out for the new TE405P with the 5 V PCI version, we had another serious crash of the phone system; therefore, I moved up the timetable of the entire project.

This configuration works for a TE410P. It is expected that, when the 405 ships, it will work with it as well.

First things first, I logged into the Nortel to see what the settings for our PRIs are. Unfortunately, our telco is not the most helpful, and providing such information was not convenient to them. But, since our phone systems have been working (well, sending and receiving calls, let's say), I figured the settings were safe.

When logged into the Nortel, I found:


Browser - USA-MICS-XC 5.0
Select
Card
1
Add...
Remove
Remove All
Terminals & sets
Lines
Restriction filters
Services
System speed dial
Passwords
Time & date
System programming
Telco features
Software keys
Hardware
Module 02
Trunk module
Card 1
Card 2
Module 03
Module 04
Module 05
Module 06
Module 07
Module 08
Module 09
Module 10
Module 11
Module 12
Module 02 - Card 1
Card type
PRI
Low line/loop
Line 001
High line/loop
Line 023
Protocol
DMS100
B channel selection sequence
Ascending Sequential
Clock source
Primary
Ready.
NUM
ANC03ENG


Browser - USA-MICS-XC 5.0
Select
Add...
Remove
Remove All
Terminals & sets
Lines
Restriction filters
Services
System speed dial
Passwords
Time & date
System programming
Telco features
Software keys
Hardware
Module 02
Trunk module
Card 1
Call-by-call service
T1 Parameters
Card 2
Module 03
Module 04
Module 05
Module 06
Module 07
Module 08
Module 09
Module 02 - Card 1 - T1 Parameters
CO fail
TIA-547A
Interface levels
ISDN
Framing
ESF
Internal CSU
On
CSU line build
0 dB
Line coding
B8ZS
Ready.
NUM
ANC03ENG

From this information, I created this zaptel.conf file, in /etc:

```
# PRI to telco

span=1,1,0,esf,b8zs
bchan=1-23
dchan=24

# PRI to Nortel

span=2,0,0,esf,b8zs
bchan=25-47
dchan=48

# Second PRI to telco

span=3,2,0,esf,b8zs
bchan=49-71
dchan=72

# Second PRI to Nortel

span=4,0,0,esf,b8zs
bchan=73-95
dchan=96
```

Next, I had to create the /etc/asterisk/zapata.conf file. Here it goes:

```
; incoming PRI
callerid=asreceived
context = incoming
switchtype = dms100
signalling = pri_cpe
group = 2
channel => 1-23

; pri to Nortel
context = fromnortel
switchtype = dms100
signalling = pri_net
group = 1
channel => 25-47

; second incoming PRI
callerid=asreceived
context = incoming
switchtype = dms100
signalling = pri_cpe
group = 2
channel => 49-71

; other PRI to Nortel
callerid=asrecieved
context = fromnortel
switchtype = dms100
signalling = pri_net
group = 1
channel => 73-95
```

With this created, I ran /etc/init.d/zaptel stop, then /etc/init.d/zaptel start, and the alarm lights on the Zaptel card went to green in about 6 seconds, and about 10 seconds later the alarm lights on the Nortel went off. Success!!!

4.5. Configuring the Dialplan

Then I proceeded to create the stuff in the extensions.conf. The non-default relevant sections are here:

```
[incoming]
include => did
include => baddid

[internal]
include => panamacity
include => services
include => parkedcalls

[baddid]
exten => _X.,1,Goto(panamacity,1000,1)
exten => s,1,Goto(panamacity,1000,1)
exten => i,1,Goto(panamacity,1000,1)

[did]
;
;
exten => 3956,1,Goto(panamacity,1000,1)
exten => 1689,1,Goto(panamacity,1000,1)
....

[fromnortel]
exten => _XXXXXXX,1,Goto(longdistance,9${EXTEN},1)
exten => _1XXXXXXXXXX,1,Goto(longdistance,9${EXTEN},1)
exten => _XXXXXXXXXX,1,Goto(longdistance,91${EXTEN},1)
exten => _XXXX,1,Goto(internal,${EXTEN},1)
exten => 911,1,Dial(Zap/g2/911)

[panamacity]
....
;
; 2xxx - Temporary Nortel Bridge
exten => _2XXX,1,Dial(Zap/g1/${EXTEN})
```

Yes, the fromnortel context is more general than it needs to be. But that's OK. It works, and I wanted to make sure I didn't break anything that was working.

As you can see, the actual connection between the two is quite simple. To dial extension 306 on the Nortel, just pick up a phone on the * server and dial 2306. To dial extension 1901 on * from a Nortel phone, just pick it up and dial 91901. Probably not the most elegant solution, but it works well.

5. Testing

As anyone will tell you, the best way to know if something works is to test it. Well, test it I did. To test, I generated a ton of phone calls and put them in /var/spool/asterisk/outgoing. You can find the format of call files in the * documentation. Suffice it to say, there were to main classes of testing that needed to be done:

1. PSTN calls (made them to my cell phone, didn't pick up)
2. Nortel calls (made to my desk extension)

Of course, this is an over-simplification. I had to test each DID to make sure it rang the right phone, I had to test all of the call queues I set up, I had to test long-distance from the Nortel switch, and various other tests.

The thing to remember is to always leave yourself an out. I only added configuration to the Nortel switch. If I had replaced things, then when the card I got from Digium had problems, we would have had to restore from a backup.

Speaking of backups, please be sure you have a current backup of the Nortel phone switch. Even though you should remove any configuration, it is a great CYA feature. If you don't know what CYA is, then just trust me.