DUNDi Configuration

John Mullinix
Cohutta.Com, Inc.
1-706-632-3343
http://www.cohutta.com

This document is a guideline for setting up DUNDi. It is provided without warranty and may not work for you, but it works for me. I took a lot of what is in this document from Anthony Messina at http://messinet.com/ and want to give him full credit for his work. I took his tutorial and configured Dundi on my machine, and then wrote this how-to with my adjustments.

DUNDi is a peer to peer (Asterisk to Asterisk) phone number sharing system. If I have 17066323343 on my machine and you are a DUNDi peer, your machine will know about my number and use the Internet to connect to me rather than the Public Switched Telephone Network (PSTN). It will also allow me to terminate calls for you on my machine, if they are local. The numbers that terminate to your machine and the exchanges you are willing to terminate are established in your extensions.conf file (or extensions_custom.conf on a machine with FreePBX on it). The beauty of this is that if you add a new number or exchange, you only add it once and within the hour it will be published to all of your peers.

This could be used within an enterprise to automatically interconnect several Asterisk based servers and keep up with dial plans on each.

```
; DUNDi configuration file
; For more information about DUNDi, see http://www.dundi.com
;
[general]
; The "general" section contains general parameters relating
; to the operation of the dundi client and server.
; The first part should be your complete contact information
; should someone else in your peer group need to contact you.
;
;department=Your Department
organization=Cohutta.Com, Inc.
locality=Epworth
stateprov=GA
country=US
email=your e-mail address
phone=+17066323343
```

Above is the first part of the dundi.conf file. Uncomment the appropriate lines and complete them with your information. This is the information that is delivered to a remote machine when a “dundi query” command is issued on the CLI.

In the next section, you can leave the bindaddr and the port commented out unless you have a need to change them. The entityid is how you want your box seen by the cloud. You can leave this commented out and the machine will use the mac address of eth0 on your machine as the entity ID.

```
; Specify bind address and port number.  Default is
; 4520
;
;bindaddr=0.0.0.0
;port=4520
;
; Our entity identifier (Should generally be the MAC address of the
; machine it's running on.  Defaults to the first eth address, but you
; can override it here, as long as you set it to the MAC of *something*
; you own!)
;
;entityid=00:07:E9:3B:76:60
```

In this next section, you don't need to make any changes from the defaults. Just make sure the ttl=32 and autokill=yes are not commented out.

```
; Peers shall cache our query responses for the specified time,
; given in seconds. Default is 3600.
;
;cachetime=3600
;
; This defines the max depth in which to search the DUNDi system.
; Note that the maximum time that we will wait for a response is
; (2000 + 200 * ttl) ms.
;
ttl=32
;
; If we don't get ACK to our DPDISCOVER within 2000ms, and autokill is set
; to yes, then we cancel the whole thing (that's enough time for one
; retransmission only).  This is used to keep things from stalling for a long
; time for a host that is not available, but would be ill advised for bad
; connections.  In addition to 'yes' or 'no' you can also specify a number
; of milliseconds.  See 'qualify' for individual peers to turn on for just
; a specific peer.
;
autokill=yes
```

This section requires no changes.

```
; pbx_dundi creates a rotating key called "secret", under the family
; 'secretpath'.  The default family is dundi (resulting in
; the key being held at dundi/secret).
;
;secretpath=dundi
;
; The 'storehistory' option (also changeable at runtime with
; 'dundi store history' and 'dundi no store history') will
; cause the DUNDi engine to keep track of the last several
; queries and the amount of time each query took to execute
; for the purpose of tracking slow nodes.  This option is
; off by default due to performance impacts.
;
;storehistory=yes
```

This next section is important and requires some changes. The mapping lines tell the remote server that queries you, the route to your resources. The first is the route to the phone numbers terminated on your machine. The second is the route to the extensions on your machine and the last is the route to exchanges that you are willing to terminate. You will need to change one of the variables listed in the default to either your outside IP address or your Fully Qualified Domain Name (FQDN). The original line looks like this:

e164 => dundi-e164-canonical,0,IAX2,dundi:${SECRET}@${IPADDR}/${NUMBER}

You should change this to look similar to the following:

e164 => dundi-e164-canonical,0,IAX2,dundi:${SECRET}@qth.cohutta.org/${NUMBER}

Apparently the variable ${IPADDR} is replaced with the local address of the machine. Here is the mappings portion of my dundi.conf file.

```
[mappings]

e164 => dundi-e164-canonical,0,IAX2,dundi:${SECRET}@qth.cohutta.org/${NUMBER}
e164 => dundi-e164-customers,100,IAX2,dundi:${SECRET}@qth.cohutta.org/${NUMBER}
e164 => dundi-e164-via-pstn,400,IAX2,dundi:${SECRET}@qth.cohutta.org/${NUMBER}
;
```

The rest of this section is reproduced below. It includes only help and nothing there needs to be

changed.

```
; The "mappings" section maps DUNDi contexts
; to contexts on the local asterisk system.  Remember
; that numbers that are made available under the e164
; DUNDi context are regulated by the DUNDi General Peering
; Agreement (GPA) if you are a member of the DUNDi E.164
; Peering System.
;
; dundi_context => local_context,weight,tech,dest[,options]]
;
; 'dundi_context' is the name of the context being requested
; within the DUNDi request
;
; 'local_context' is the name of the context on the local system
; in which numbers can be looked up for which responses shall be given.
;
; 'weight' is the weight to use for the responses provided from this
; mapping.  The number must be >= 0 and < 60000.  Since it is totally
; valid to receive multiple reponses to a query, responses received
; with a lower weight are tried first.  Note that the weight has a
; special meaning in the e164 context - see the GPA for more details.
;
; 'tech' is the technology to use (IAX, SIP, H323)
;
; 'dest' is the destination to supply for reaching that number.  The
; following variables can be used in the destination string and will
; be automatically substituted:
; ${NUMBER}: The number being requested
; ${IPADDR}: The IP address to connect to
; ${SECRET}: The current rotating secret key to be used
;
; Further options may include:
;
; nounsolicited:  No unsolicited calls of any type permitted via this
;                 route
; nocomunsolicit: No commercial unsolicited calls permitted via
;                 this route
; residential:    This number is known to be a residence
; commercial:     This number is known to be a business
; mobile:         This number is known to be a mobile phone
; nocomunsolicit: No commercial unsolicited calls permitted via
;                 this route
; nopartial:      Do not search for partial matches
;
; There *must* exist an entry in mappings for DUNDi to respond
; to any request, although it may be empty.
;
```

This next section requires that you generate a set of keys for you phone server. Log in as root and navigate to the /var/lib/asterisk/keys folder. Then run the command "astgenkey -n <servername>. I used my FQDN for the servername but you can use almost anything you like. On my machine, I ran the command astgenkey -n qth.cohutta.org. On the command line it looked like this

cd /var/lib/asterisk/keys
astgenkey -n qth.cohutta.org

This creates 2 keys <servername>.pub and <servername>.key (or in my case, qth.cohutta.org.pub and qth.cohutta.org.key). Put your public key on you website, or somewhere else that the public can get to it. Leave your private key (the one ending in ".key") in that directory on your server. You can download my public key at: http://www.cohutta.com/qth.cohutta.org.pub

The following section identifies the entities that you are peering with. The first is my peering information. You can copy this information into your dundi.conf file or download it from http://www.cohutta.com/peerinfo.txt. I keep a copy in my dundi.conf file, but you will notice that it is commented out.

The next is one of the servers I peer with at messinet.com. He provided me with this information and I copied it into my dundi.conf file. In these sections, you will need to insert your private key name (without the ".key" on the end.)

```
;[00:08:54:E0:72:1C]
;model = symmetric
;host = qth.cohutta.org
;auth = rsa
;inkey = qth.cohutta.org
;outkey = your.private.key
;include = all
;permit = all
;qualify =yes
;order = primary

[00:12:3F:2A:38:72]
model = symmetric
auth = rsa
host = messinet.com
inkey = messinet.com
outkey = <YOUR_KEY_NAME>
include = e164
permit = e164
qualify = no
order = primary
```

Once you have this information entered do the following at the command line:

cd /var/lib/asterisk/keys
wget http://www.cohutta.com/qth.cohutta.org.pub
and in the case of messinet
wget https://messinet.com/~amessina/projects/dundi/messinet.com.pub
This will download the peer's public key.

Now, you can either restart asterisk or at the asterisk CLI issue:

CLI> reload res_crypto.so
and then
CLI> reload pbx_dundi.so

You should now be able to issue the query command at the CLI.

CLI> dundi query 00:08:54:E0:72:1C

and it should return:

DUNDi Query EID succeeded:
Department:
Organization: Cohutta.Com, Inc.
City/Locality: Epworth
State/Province: GA
Country: US
E-mail: john@cohutta.com
Phone: +17066323343
IP Address: 69.11.191.142

The next step is to create an IAX2 trunk in iax.conf (iax_custom.conf in FreePBX enabled machines.) The trunk information looks like this and does not need to be changed. Your peers will use this trunk to access your machine.

```
[dundi]
type=user
dbsecret=dundi/secret
context=dundi-e164-local
```

The final step is to add the dundi peering to your dial plan. Open extensions_custom.conf as the final steps are completed there (extensions.conf if you are running pure Asterisk.) The first section of this is the numbers you terminate on your machine and the context is called dundi-e164-canonical. I list all of the numbers that I terminate on my machine. I include the country code with the number, which will make this system work with international calls also.

You will notice that I have Noop() statements next to each number. The numbers have to be listed here and you have to include instructions on routing the call. In my case, the lines that include the numbers:

exten => 17066323343,1,noop()

are just placeholders to advertise my numbers to peers. When the peer hits this context, with one of my numbers, it will hit the noop line and then fall to the include at the bottom:

include => from-trunk

and that routes the call into my server as if it is a call from the outside world. You must have a DID inbound route that matches the numbers you are advertising for this to work.

```
; the first three contexts are separated in order to group them by the appropriate dundi priority

; Private DUNDi network

[dundi-e164-canonical]
; List canonical entries here
exten => 17066323343,1,noop()
exten => 17066323824,1,noop()
exten => 16782794442,1,noop()
exten => 14109884220,1,noop()
exten => 17062583885,1,noop()
include => from-trunk
```

Next we deal with the local extensions on your machine. If you do not want to advertise them, do what I have done here; comment it out.

```
[dundi-e164-customers]
; If you are an ITSP or Reseller, list your customers here.
;exten => _12564286000,1,Dial(SIP/customer1)
;exten => _12564286001,1,Dial(IAX2/customer2)
```

Finally I advertise the exchanges I am willing to terminate. Be sure to include the preceding "_" since you are pattern matching. If it matches, the call is sent to the outbound route I created in FreePBX for my local calls. This context is what advertises your numbers to the peers connected to you. I have changed this portion of my dial plan since the first edition of this document. I discovered a flaw in the original that allowed your machine to match any number. If you have the original version of this paper, you will notice that the last line of this context is exten => x.,n,..... That "_X." matched everything hitting your machine, thus creating a bit of a problem. Here, each line calls a macro if it matches and the macro sends the call to be dialed.

The macro makes the call, so make sure you have a good route for it. Mine prepends a *67 to the number to be dialed and send the number to a context called from-dundi that calls the outbound route for my local dial tone.

```
[dundi-e164-via-pstn]
; If you are freely delivering calls to the PSTN, list them here
exten => _1706251XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706258XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706455XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706572XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706632XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706633XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706851XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706946XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1423219XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1423241XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1423496XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1423548XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1423761XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706374XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706492XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706838XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706900XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1706964XXXX,1,macro(call_dundi,${EXTEN:1})
exten => _1828494XXXX,1,macro(call_dundi,${EXTEN:1})

[macro-call_dundi]
exten => s,1,goto(from-Dundi,*67${ARG1},1)
```

The next addition to extension_custom.conf is some routing information for the inbound IAX trunk you created called “dundi” in your iax_custom.conf file. This context forms a wrapper of the 3 contexts you created above and lets inbound calls check all 3 contexts.

```
[dundi-e164-local]
; this context is used to group the 3 above and reference in iax.conf for the incoming
calls referred by dundi
include => dundi-e164-canonical
include => dundi-e164-customers
include => dundi-e164-via-pstn
```

Next create the switch. I am not really sure what this does, but probably calls Dundi stuff in Asterisk.

```
[dundi-e164-switch]
switch => DUNDi/e164
```

This section of extensions_custom sets up how to handle lookups.

```
[dundi-e164-lookup]
; Locally to lookup, try looking for a local E.164 solution
; then try DUNDi if we don't have one.
include => dundi-e164-local
include => dundi-e164-switch
```

This last section includes a macro that is run when you hit the trydundi context and is for outbound dialing.

```
[macro-dundi-priv]
exten => s,1,Goto(${ARG1},1)
include => dundi-e164-lookup

[trydundi]
exten => _.,1,Macro(dundi-priv,${EXTEN})
exten => _.,2,Congestion
```

Finally, go to FreePBX and add a custom trunk. In the dial rules section I put:

1+NXXNXXXXXX

to prepend a 1 to 10 digit dials
and in the Custom Dial String I put:

local/$OUTNUM$@trydundi

Include this trunk as the first choice of all outbound routes where you want to try Dundi.

This should get you up and running. When you are ready to try it, send me your peering information and we will give it a whirl. My public key and peering information can be downloaded from:

http://www.cohutta.com/qth.cohutta.org.pub
http://www.cohutta.com/peerinfo.txt

I would like to find peers in Lake Wales, Florida, Baltimore, Maryland and Atlanta, Georgia, but I will take them from anywhere.

If you have questions or find anything that needs correction, feel free to contact me by e-mail: john at cohutta dot com or 1-706-632-3343.

John Mullinix
Cohutta.Com, Inc.
http://cohutta.com
1-706-632-3343 ENUM enabled
sip://17066323343@qth.cohutta.org

john at cohutta dot com