

Ronny Egner's Blog

Blog about Oracle related topics

[Home](#) > [My Profile](#) > [Oracle 11g Release 2 – Install Guide](#) > [Resources](#)

Type text to search here...

[Home](#) > [Oracle in general](#) > Storing Oracle database connection strings in OpenLDAP instead of OID

Storing Oracle database connection strings in OpenLDAP instead of OID



September 25, 2009



ronnyegner



[Go to comments](#)



[Leave a comment](#)



[1 Votes](#)

Dear Readers,

my blog has moved to a different server.

The new address is:

<http://blog.ronnyegner-consulting.de>

Storing Oracle database connection strings in OpenLDAP – or – “Oracle Names for the masses”

Naming Methods

Typically Oracle stores it's database connection strings in a file called TNSNAMES.ORA. Among storing every database connection string in this file there are other different naming schemas such as EZCONNECT, Oracle Names “ONAMES” which is deprecated with >= 10g or LDAP.

I've seen a lot of users with a centralized tnsnames.ora file placed on a cifs share and access by every oracle client. Using such a technique with unix hosts is not straight forward (i am no friend of having cifs shares mounted on database servers) and often due to security restrictions not possible. Sometimes i have also seen the tnsnames distributed by using rsync or just copy-on-demand.



[RSS feed](#)

Recent Posts

- [The blog has moved](#)
- [Backing up your system configuration](#)
- [Oracle on linux – yes of course – but what linux?](#)
- [ORA-01555 \(snapshot too old\) error although undo_retention set to a high value](#)
- [11g Release 2 release rumors](#)
- [Data Pump falsely reports the number of imported rows as 1 \(one\)](#)
- [New public oracle yum server](#)
- [Configuring a small DNS server for SCAN](#)
- [ASM resilvering – or – how to recover your crashed cluster](#)
- [ASM resilvering – or – how to recover your crashed cluster – Test no 4](#)

Categories

- [Oracle 11g Release 2](#)
- [Oracle ASM](#)
- [Oracle in general](#)

From my point of view there is a superior solution for this problem: Oracle Names.

Oracle Names as naming method

Oracle Names was introduced with Oracle 8 (if i remember correctly) and stored database connection strings in a database schema. Clients only needed a SQLNET.ORA pointing to the server running the onames service. If a client tried to resolve a database connection string the oracle client queried the oname service which replied with the current connection string. Afterwards the client connected to the database using the received connection string. Changes to connection strings were nearly instantaneous. If needed connection strings could be overwritten by specifying them in the local tnsnames.ora file.

But starting with Oracle Database 10g Oracle Names was deprecated and replaced by integrating it in the Oracle Internet Directory OID. The naming resolution in the OID was ldap-based. If users wanted to use Oracle Names from now on they had to install and license the complete middle tier stack which included OID.

Due to the high costs for resources and licenses i tried to implement the naming resolution in another LDAP server. I had two choices: openLDAP and Microsoft Active Directory (which was the central ldap server). After reading a bit i discarded storing the objects in active directory due to users complaining about various problems. So openLDAP was left. I found an article somewhere on the net showing how to migrate the needed schema objects from OID to openLDAP. I would have posted the link if i were able to find it again. But maybe you can help me out 😊

Implementation

Implementation on the client

All you need to do is to place the following SQLNET.ORA and LDAP.ORA in the \$TNS_ADMIN directory of your oracle client:

SQLNET.ORA

```
“ NAMES.DIRECTORY_PATH= (LDAP, TNSNAMES)
```

LDAP.ORA

```
“ DIRECTORY_SERVERS=(onames1.regner.de:389,onames2.regner.de:389)  
  DEFAULT_ADMIN_CONTEXT="dc=regner,dc=de"
```

These two files are completely static. As you may note NAMES.DIRECTORY_PATH specifies where to look for database connection strings. As defined above the LDAP server is primary

📄 [Uncategorized](#)

Blogroll

- 📄 [Blog at WordPress.com.](#)
- 📄 [Blog at WordPress.com.](#)

Archives

- 📄 [October 2009](#)
- 📄 [September 2009](#)

Meta

- 📄 [Register](#)
- 📄 [Log in](#)

and TNSNAMES.ORA is secondary. If you want to overwrite a connecting string you just need to exchange the two values for the local TNSNAMES.ORA to gain preference.

The LDAP.ORA contains the comma separated names of the servers to be queried. You can list up to three servers. If one server is unreachable oracle net client will fail over to the second server after one second.

My implementation aliases the destination servers on DNS level so that the client configuration files are completely static. If a server is migrated or replaced you just update the dns alias and everything works just like before. You do not need to touch the clients configuration files ever again!

Implementation on the server

Configuring openLDAP

Required Ressources

First of all you need an installed openLDAP server, openLDAP utilities ("ldapadd", "ldapsearch" ships with them) and openLDAP-libraries. Most distributions include openLDAP. Red Hat Enterprise Server 5 and Oracle Enterprise Linux 5 do so.

OpenLDAP for storing oracle connection entries has a extremely small footprint. The ldap database itself is less than 10 MB in size. The ldap server itself requires less than 50 MB memory. Tuning openLDAP should not be required unless you have a few hundred/thousand entries.

Required files

The following files implement the ldap objects needed for storing the database connection entries:

- oidrdbms.schema
- alias.schema
- oidbase.schema
- oidnet.schema

They are taken entirely from the OID schema objects and a little bit converted for openLDAP. You need to place them in /etc/named/schema-oid. Due to wordpress' file type restriction you can download the four files [here in open office format](#) and [here in pdf format](#).

Configuring openLDAP

Add the following lines to the file slapd.conf (alter the paths if needed):

```
“ # extension for oracle names
```

```
include /etc/openldap/schema-oid/oidbase.schema
include /etc/openldap/schema-oid/oidnet.schema
include /etc/openldap/schema-oid/oidrdbms.schema
include /etc/openldap/schema-oid/alias.schema
```

Configure the ldap base in slapd.conf and make sure you change the password:

```
“ defaultsearchbase "dc=regner,dc=de"
  database bdb
  suffix "dc=regner,dc=de"
  rootdn "cn=Manager,dc=regner,dc=de"
  rootpw changeme
```

You should also set up replication right now. The following is a simple master-slave-replication. OpenLDAP supports different types of replication. See the [documentation](#) for more information.

```
“ # Replicas of this database
  replica host=onames2.regner.de:389
    binddn="cn=Manager,dc=regner,dc=de"
    bindmethod=simple credentials=changeme
  replogfile /var/lib/ldap/replication.log
```

Creating the ldap base

Put the following lines in a file named "add_base.ldif":

```
“ dn: dc=regner,dc=de
  objectClass: top
  objectClass: dcObject
  objectClass: organization
  o: regner.de
  dc: regner

  dn: cn=OracleContext,dc=regner,dc=de
  objectclass: orclContext
  cn: OracleContext
```

Add them with (you will be prompted for the password):

```
“ ldapadd -c -x -D "cn=Manager,dc=regner,dc=de" -W -f add_Base.ldif
```

This will create the ldap base. If everything succeeded you are ready to insert your first connection string.

Managing entries

Add an entry

The following sample adds a database connection string for the database named “foo”:

```
“ dn: cn=foo,cn=OracleContext,dc=regner,dc=de  
  objectclass: top  
  objectclass: orclNetService  
  cn: foo  
  orclNetDescString: (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=rac1.regner.de)(PORT=1521)))
```

It works best when putting the lines above in a file and adding it with:

```
“ ldapadd -c -x -D "cn=Manager,dc=regner,dc=de" -W -f add_entry.ldif
```

We tested the just added entry:

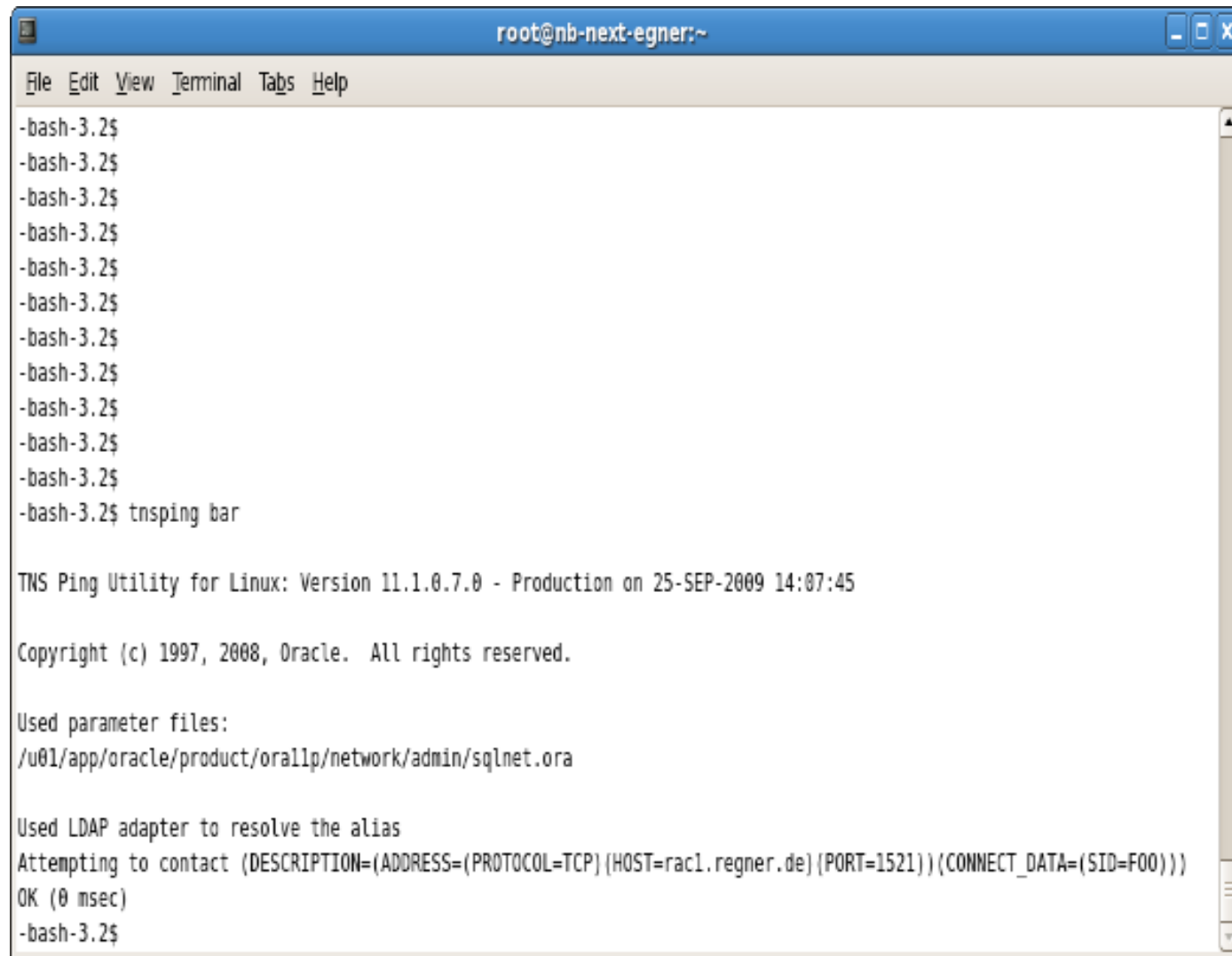
```
root@nb-next-egner:~  
File Edit View Terminal Tabs Help  
-bash-3.2$  
-bash-3.2$ pwd  
/u01/app/oracle/product/orallp/network/admin  
-bash-3.2$ ls -la  
total 36  
drwxr-xr-x 3 orallp dba 4096 Sep 25 14:05 .  
drwxr-xr-x 12 orallp dba 4096 Sep 25 14:05 ..  
-rw-r--r-- 1 orallp dba 74 Sep 25 14:04 ldap.ora  
-rw-rw-r-- 1 orallp dba 329 Aug 6 15:57 listener.ora  
drwxr-xr-x 2 orallp dba 4096 Aug 4 07:39 samples  
-rw-r--r-- 1 orallp dba 39 Sep 25 14:04 sqlnet.ora  
-bash-3.2$ tnsping foo  
  
TNS Ping Utility for Linux: Version 11.1.0.7.0 - Production on 25-SEP-2009 14:07:35  
  
Copyright (c) 1997, 2008, Oracle. All rights reserved.  
  
Used parameter files:  
/u01/app/oracle/product/orallp/network/admin/sqlnet.ora  
  
Used LDAP adapter to resolve the alias  
Attempting to contact (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=rac1.regner.de)(PORT=1521))(CONNECT_DATA=(SID=FOO)))  
OK (0 msec)  
-bash-3.2$
```

Add an alias

Adding an alias requires the destination to be already inserted. In the following example we will alias "bar" pointing to "foo":

```
“ dn: cn=bar,cn=OracleContext,dc=regner,dc=de  
objectclass: orclNetServiceAlias  
objectclass: top  
objectclass: alias  
cn: bar  
aliasedobjectname: cn=foo,cn=OracleContext,dc=regner,dc=de
```

Testing the alias – the expected result is the same connection string as “foo”:



```
root@nb-next-egner:~
File Edit View Terminal Tabs Help
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$
-bash-3.2$ tnsping bar

TNS Ping Utility for Linux: Version 11.1.0.7.0 - Production on 25-SEP-2009 14:07:45

Copyright (c) 1997, 2008, Oracle. All rights reserved.

Used parameter files:
/u01/app/oracle/product/orallp/network/admin/sqlnet.ora

Used LDAP adapter to resolve the alias
Attempting to contact (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=rac1.regner.de)(PORT=1521))(CONNECT_DATA=(SID=FOO)))
OK (0 msec)
-bash-3.2$
```

Delete an entry

Deletion is quite simple. The following command will delete the database connection string “foo” added above:

```
“ ldapdelete -D "cn=Manager,dc=regner,dc=de" -w sparc
-x "cn=foo,cn=OracleContext,dc=regner,dc=de"
```

Required Resources

So looking at the ressources we need:

Ads by Google

Esb Connection

Achieve Success Using an ESB in SOA Initiatives. Free White Paper.
web.progress.com

 Oracle in general



Like

Be the first to like this post.



Leave a comment



Trackback

Trackbacks (0)

Comments (0)

No comments yet.

Leave a Reply

Your email address will not be published. Required fields are marked *

Name *

Email *

Website

☐ Notify me of follow-up comments via email.

[Post Comment](#)

☐ Notify me of site updates

◀ [Oracle 11g Release 2 – SSH equivalence gone after reboot](#)
[Partition layout on Linux – Recommendations](#) ▶



Copyright © 2009-2010 Ronny Egner's Blog
Blog at WordPress.com. Theme: INove by NeoEase.

[⬆ Top](#)