**How to Connect Without TNSNAMES**

* by [Ed Chen](https://logic.edchen.org/author/admin/)
* May 30, 2019

**Connect No TNSNAMES**

Yes, sometimes we are not allowable to change tnsnames.ora, even a tiny bit. How can we connect to new or moved databases?

In this post, I provided some ways to connect to databases without touching the original tnsnames.ora.

1. [Use Full Connect Descriptor](https://logic.edchen.org/how-to-connect-without-tnsnames/#use-full-connect-descriptor)
2. [Use Your Own tnsnames.ora](https://logic.edchen.org/how-to-connect-without-tnsnames/#use-your-own-tnsnames-ora)
3. [Use Easy Connect Method](https://logic.edchen.org/how-to-connect-without-tnsnames/#use-easy-connect-method)

**1. Use Full Connect Descriptor**

Intrinsically, tnsnames.ora is an alias management of service names in a file. Every entry in the file contains a pair of **connect identifier** and **connect descriptor**. For example:

...  
ORA11G =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = ora11g-1.example.com)(PORT = 1521))  
    (CONNECT\_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE\_NAME = ORCL)  
    )  
  )

In the above entry, ORA11G is the connect identifier, everything behind the equal sign is the connect descriptor. More specifically, the connect identifier is just an alias which can be used in connection string to represent their own lengthy connect descriptor for convenience.

C:\Users\edchen>sqlplus /nolog  
..  
SQL> conn hr/hr@ora11g  
Connected.

In other words, the connect identifier, the alias will be translated into its own connect descriptor at connect-time. That is to say, the real working thing is the connect descriptor. That's why we replace the connect identifier with the connect descriptor to connect a new or moved database.

SQL> conn hr/hr@"(DESCRIPTION = (ADDRESS\_LIST = (ADDRESS = (PROTOCOL = TCP)(HOST = ora11g-2.example.com)(PORT = 1521)))(CONNECT\_DATA = (SERVICE\_NAME = ORCL)))"  
Connected.

For safety, we use double quotes to wrap the string. It connected successfully.

**2. Use Your Own tnsnames.ora**

I know you cannot touch the original tnsnames.ora, but you can use your own tnsnames.ora. The trick is to [set a TNS\_ADMIN environment variable](https://logic.edchen.org/tns_admin-how-and-why/) temporarily for your session and put your own tnsnames.ora in that directory. Then connection tools including sqlplus will know where to find the new tnsnames.ora.

**For Linux Platform**

**Set TNS\_ADMIN Path**

[oracle@test ~]$ export TNS\_ADMIN=/tmp  
[oracle@test ~]$ echo $TNS\_ADMIN  
/tmp

This setting will override the original path to tnsnames.ora.

**Modify tnsnames.ora File**

[oracle@test ~]$ vi /tmp/tnsnames.ora  
...  
ORA11G\_NEW =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = ora11g-2.example.com)(PORT = 1521))  
    (CONNECT\_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE\_NAME = ORCL)  
    )  
  )

**Test Old Connect Identifier**

[oracle@test ~]$ sqlplus /nolog  
...  
SQL> conn hr/hr@ora11g  
ERROR:  
ORA-12154: TNS:could not resolve the connect identifier specified

Of course it got error, because we point the network configuration to the new location TNS\_ADMIN.

**Test New Connect Identifier**

SQL> conn hr/hr@ora11g\_new  
Connected.

**For Windows Platform**

**Set TNS\_ADMIN Path**

C:\Users\edchen>set TNS\_ADMIN=d:\  
C:\Users\edchen>echo %TNS\_ADMIN%  
d:\

This setting will override the original path to tnsnames.ora.

**Modify tnsnames.ora File**

C:\Users\edchen>notepad d:\tnsnames.ora  
...  
ORA11G\_NEW =  
  (DESCRIPTION =  
    (ADDRESS = (PROTOCOL = TCP)(HOST = ora11g-2.example.com)(PORT = 1521))  
    (CONNECT\_DATA =  
      (SERVER = DEDICATED)  
      (SERVICE\_NAME = ORCL)  
    )  
  )

**Test Old Connect Identifier**

C:\Users\edchen>sqlplus /nolog  
...  
SQL> conn hr/hr@ora11g  
ERROR:  
ORA-12154: TNS:could not resolve the connect identifier specified

Of course it got error, because we point the network configuration to the new location TNS\_ADMIN.

**Test New Connect Identifier**

SQL> conn hr/hr@ora11g\_new  
Connected.

The trick is successfully.

Please note that, TNS\_ADMIN set here is just only for your session, once you exit, the environment variable is reset. Of course, you can also [set TNS\_ADMIN permanently](https://logic.edchen.org/tns_admin-how-and-why/#how-to-set-tns_admin) if you have the right to do it.

**3. Use Easy Connect Method**

By default, you can use Easy Connect naming method to connect databases, this is because [the default connect methods of NAMES.DIRECTORY\_PATH in sqlnet.ora includes EZCONNECT](https://docs.oracle.com/en/database/oracle/oracle-database/19/netrf/parameters-for-the-sqlnet.ora.html#GUID-65EF582E-F414-43EE-A70F-EC90E0B7292F). Which is:

NAMES.DIRECTORY\_PATH = (TNSNAMES, LDAP, EZCONNECT)

You have to provide the following information in order to use Easy Connect naming method.

* Where do you want to go?
* What is the listener port?
* What is the service name?

In this case, the destination is ora11g-2.example.com, the listener port is 1521 and the service name is ORCL.

C:\Users\edchen>sqlplus /nolog  
...  
SQL> conn hr/hr@ora11g-2.example.com:1521/orcl  
Connected.

It seems a simple way to connect to database, you can find out [more concepts and examples about Easy Connect naming method in Oracle documentation](https://docs.oracle.com/en/database/oracle/oracle-database/19/netag/configuring-naming-methods.html#GUID-B0437826-43C1-49EC-A94D-B650B6A4A6EE).

On the other way, if DBA explicitly disabled the naming method by removing EZCONNECT from the list in sqlnet.ora, like the following:

NAMES.DIRECTORY\_PATH = (TNSNAMES)

You are no longer to use Easy Connect.

C:\Users\edchen>sqlplus /nolog  
...  
SQL> conn hr/hr@ora11g-2.example.com:1521/orcl  
ERROR:  
ORA-12154: TNS:could not resolve the connect identifier specified