

# Setting Up Generic Connectivity

## Purpose

This document details how to setup generic connectivity from Oracle to a Non-Oracle OLEDB or ODBC datasource.

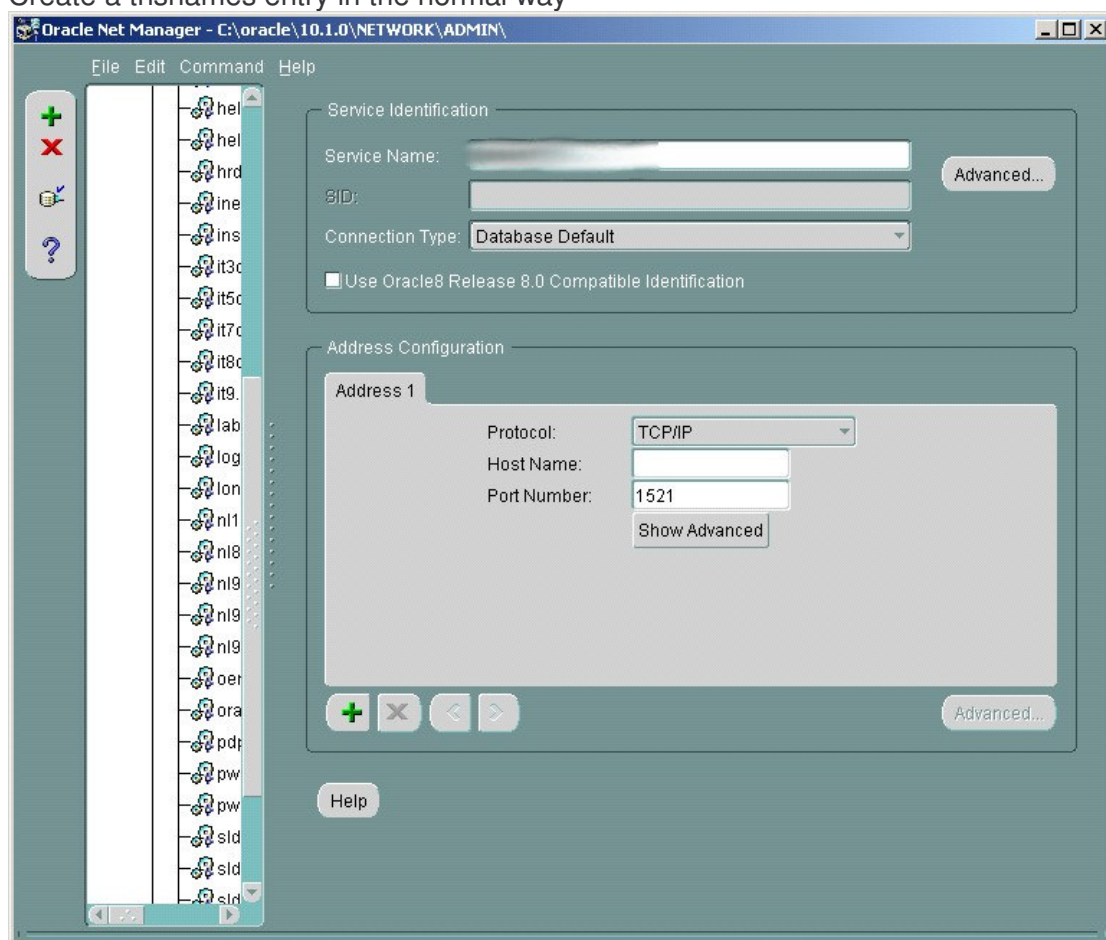
## Requirements

You will need either an ODBC or OLEDB driver for the non-oracle database on the Oracle server.

A basic understanding of Net8 will help.

## How-To

1. Configure a tnsnames entry for the datasource on the server.
  - a. Run NetManager
  - b. Create a tnsnames entry in the normal way



- c. Click advanced next to the service name and tick use for heterogeneous services

Advanced Service Options

Additional Service Settings

Instance Name:

Session Data Unit:

Use for Heterogeneous Services

Oracle Rdb Settings

Rdb Database:

Type of Service:

Global Database Name:

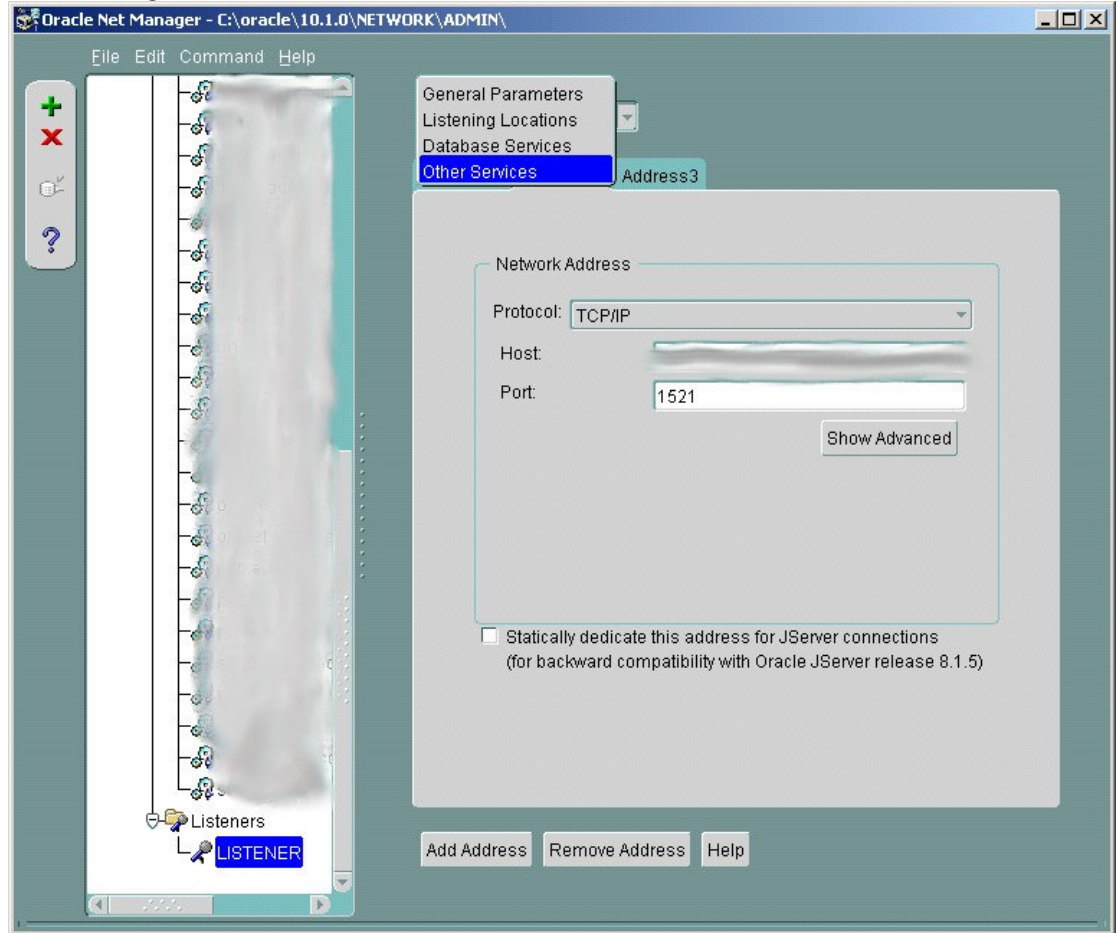
Help OK Cancel

- d. Save the new entry – it will look like this in tnsnames.ora

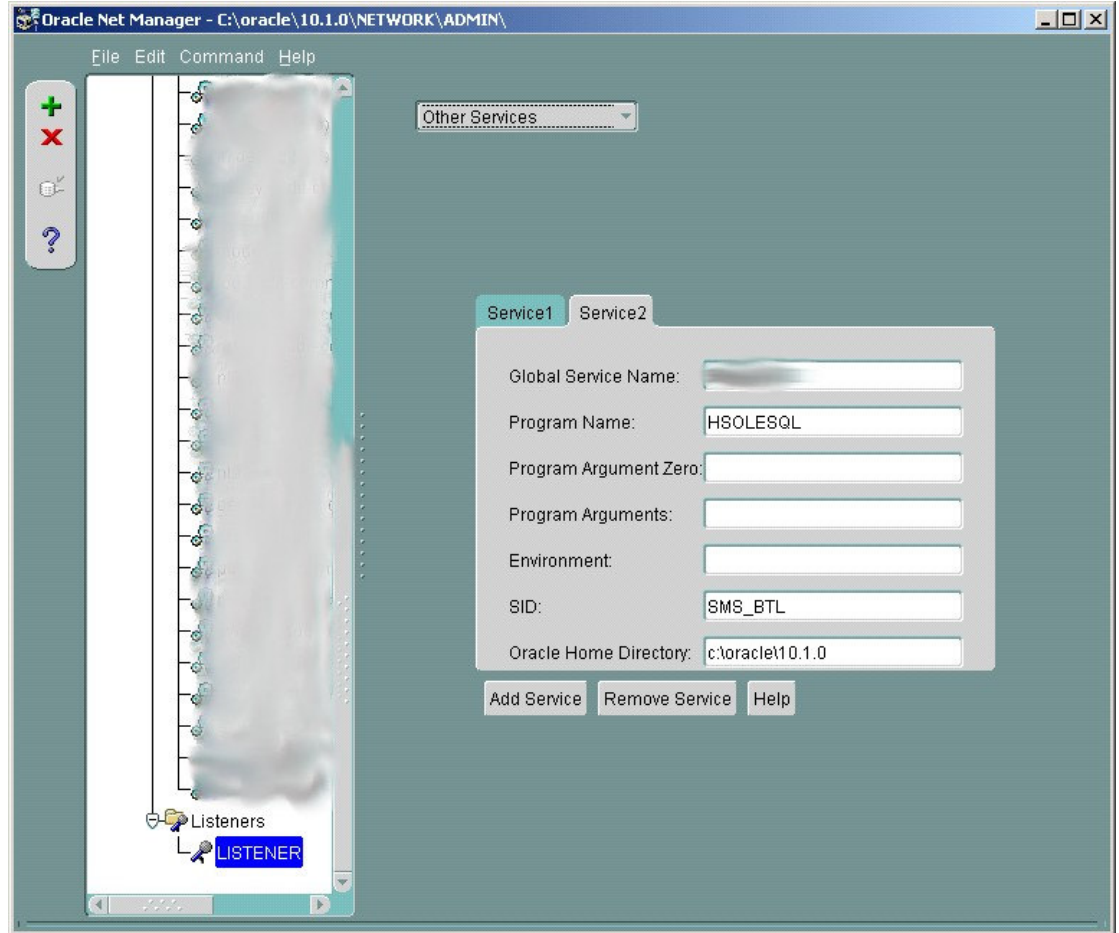
```
<tnsnames entry> =  
(DESCRIPTION =  
  (ADDRESS_LIST =  
    (ADDRESS = (PROTOCOL = TCP)(Server = <server_name>)(PORT = 1521))  
  )  
  (CONNECT_DATA =  
    (SERVICE_NAME = <name of link>)  
  )  
(HS = OK)  
)
```

2. Configure the listener to listen for requests for the service

a. In NetManager Choose Other services



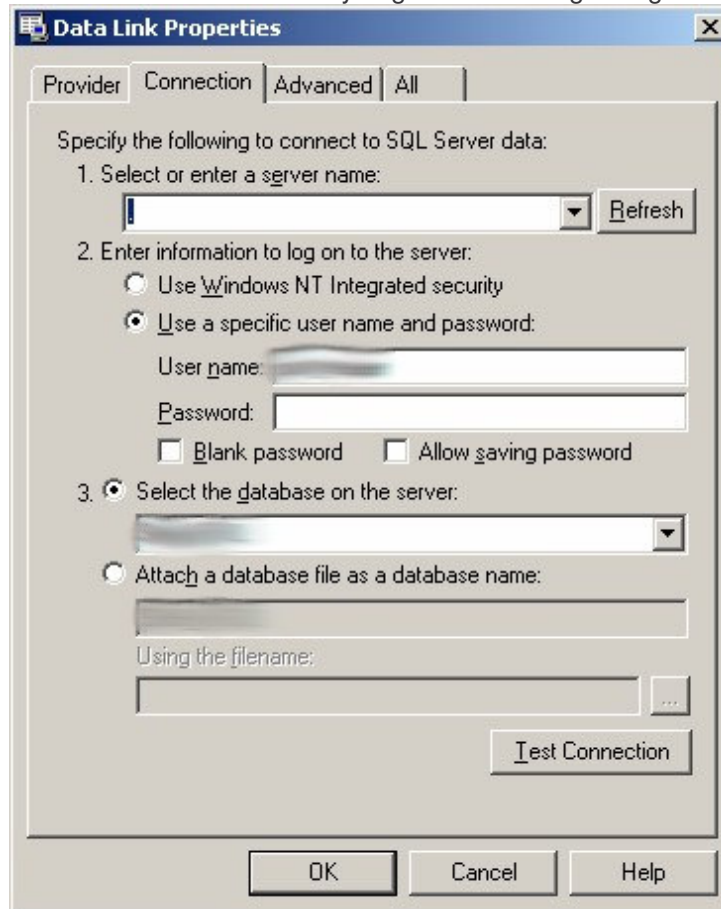
- b. Click Add Service Fill out the screen as follows



Global Service name should be the same as the service name in tnsnames.ora, as should sid  
 Program name should be HSOLESQLE if you are using OLEDB for a Database (i.e not the text driver). For ODBC it should be HSODBC – this corresponds to the executable in the oracle home bin directory.  
 Oracle Home also has to be filled in. Save listener.ora

- c. Load the changes you have made by running lsnrctl reload at the server command prompt
- d. Check that the service is listed if you type lsnrctl status
3. Configure an initialisation file for the generic agent (hsolesql.exe/hsodbc.exe).
  - a. this is called init<service\_name>.ora and sits in oracle\_home\hs\admin it only **requires** One parameter  
 HS\_FDS\_CONNECT\_INFO
  - b. For OLEDB you need to have a datalink file and the parameter reads  
 HS\_FDS\_CONNECT\_INFO="UDLFILE=<FILENAME>" eg  
 HS\_FDS\_CONNECT\_INFO="c:\\temp\\<service\_name>.udl" – note double slashes
  - c. For ODBC you need to have a dsn – a system dsn is easiest and the parameter reads  
 HS\_FDS\_CONNECT\_INFO=<dsnname>
4. Configure the datasource
  - a. For ODBC just create a datasource in the standard way
  - b. For OLEDB create a blank text file called by the filename listed above

- c. Double click the udl file and you get the following dialog



- d. You can choose the provider and connection details from various tabs.
5. Create the database link.
- This is done in the usual way eg  
create database link <link>  
connect to <user> identified by <password>  
using '<tnsnames entry>';
  - You can also authenticate using the logged in Oracle user name, but then you would have to maintain the security at the remote sql server site as well.
  - Test using `select * from table@link;` - may need quotes and owner name for sql server systems eg `select * from "dbo"."sysusers"@<link>;`

## Troubleshooting

If it all goes wrong

Include a parameter `HS_FDS_TRACE_LEVEL = ON` in the `init<sid>.ora` file in `oracle_home\hs\admin` this should create a trace file with useful info in it. In `oracle_home\hs\trace`

If the trace file is not created then check `listener.log` – file not found means you have typed the program name wrong in the listener configuration.

## Tip

Use the same name for all sids and service names so that you always call the remote database the same thing – I'd use the same name for the database link as well.

### ***How It works (briefly)***

The listener calls the external program (hsolesql.exe for example) which makes the connection using the connect info that it finds in the init<sid>.ora file in hs/admin and returns the data to the server.