

Die Welt ist nicht immer „Oracle“

Alexander Pilfusek
Witt-Gruppe
Weiden i. d. OPf.

Schlüsselworte

Heterogene Umgebung, Gateways, Datenbank, Teradata, Gateway-Server

Einleitung

In unserem Konzern werden unter anderem Daten einer Teradata benötigt, um diese in unserem Oracle-Datwarehouse weiter zu verarbeiten.

Bisher war das Gateway dafür auf einem AIX-Server installiert, auf dem auch ein Knoten des Datawarehouses lief. Da das Warehouse auf eine Exadata migriert wurde, musste auch ein neuer Platz für das Gateway gesucht werden. Dieser sollte zentral sein und darüber hinaus die Möglichkeit bieten, die Verbindungsmöglichkeiten zu erweitern (z. B. MySQL oder MSSQL).

Für dieses Vorhaben bietet sich „Oracle Database Gateways“ an, das über die Oracle-Seite heruntergeladen werden kann. In unserer Umgebung kommt ein eigener virtueller Server zum Einsatz. Dieser Vortrag zeigt die Vorbereitung und Installation eines „Gateway-Servers“ auf einer virtuellen Maschine mit Suse Enterprise Linux 11 in Verbindung mit einer Oracle-DB-Version 12.1 und einer Teradata-DB-Version 15.10.

Erste Fragen und Antworten

Wie oben erwähnt, ist die Software für den Gateway-Server auf der Oracle-Seite abrufbar. Eine Übersicht über „Oracle Database Gateways“ findet sich hier: <http://www.oracle.com/technetwork/database/gateways/>

Hier gibt es allerdings nur ein paar allgemeine Infos, Whitpapers, etc. Links zu Installationen oder Anleitungen sucht man hier vergebens. Erste Fragen tauchen auf:

- Welches Betriebssystem ist zertifiziert für dieses Vorhaben?
- Welche Oracle-Versionen können eingesetzt werden?
- Wo finde ich die Downloads?
- Welche Dimensionen muss meine VM haben?

Für die ersten beiden Fragen hilft bereits die Übersichtsseite. Hier findet sich die Zertifizierungsmatrix:
Supported Platforms for 12.2 Gateways

Platform	O/S Version
Linux x86-64	RH 5, 6, 7 OEL 5, 6, 7 SLES 11, 12 NeoKylin Linux Advanced Server v6, v7 Asianux v7
Windows x64	7, 8, 8.1, 10, 2008, 2008 R2, 2012, 2012 R2
Solaris SPARC	Solaris 10, 11
Solaris x86-64	Solaris 10, 11
AIX	v7.2, v7.1, v6.1
HP-UX Itanium	v11.31

Product v12.2	Oracle	Non-Oracle	Notes
Database Gateway for Teradata	12.2 12.1	15.x 14.x 13.x 12 V2R6	You need to install Teradata client libraries on the same machine where the gateway is running Certification with Teradata 15.1 requires Gateway version 12.2.0.2 or patch #25599964

Abb. 1: Zertifizierungsmatrix 12.2 OS und DB

In unserem Unternehmen setzen wir für diesen Zweck eine virtuelle Maschine mit SLES 11 ein. Als Datenbankversion kommt Oracle 12.1 zum Einsatz. Die Teradata hat Version 15:

```
select * from dbc.dbcinfo;

VERSION 15.10.03.02b
RELEASE 15.10.03.02
LANGUAGE SUPPORT MODE
Standard
```

Abb. 2: Abfrage und Ausgabe der Teradata-DB-Version

Diese Konfiguration ist also unterstützt.

Die weiteren Fragen lassen sich am besten über „docs.oracle.com“ klären. Eine Suche nach „oracle gateway server teradata“ bringt gute Ergebnisse:

<https://docs.oracle.com/apps/search/search.jsp?category=all&q=oracle+gateway+server+teradata>

Im ersten Teil (Overview) findet man, was man alles downloaden muss. Nun gut, so direkt steht es nicht geschrieben, aber man wird auf die Standard-Download-Seite gelotst:

<http://www.oracle.com/technetwork/indexes/downloads/>

Hier ergibt eine Suche nach „gateways“, dass man in der Standard-DB-Seite weiterzusuchen hat:

<ul style="list-style-type: none"> Database 12c Enterprise/Standard Editions Including: Client, Grid Infrastructure, Examples, Gateways more 	(12.1.0.2.0) - Enterprise Edition	
	Microsoft Windows x64 (64-bit)	File 1, File 2 (2.5 GB) See All
	Linux x86-64	File 1, File 2 (2.5 GB) See All
	Oracle Solaris (SPARC systems, 64-bit)	File 1, File 2 (2.6 GB) See All
	Oracle Solaris (x86 systems, 64-bit)	File 1, File 2 (2.3 GB) See All
	HP-UX Itanium	File 1, File 2 (3.1 GB) See All
	AIX (PPC64)	File 1, File 2 (2.7 GB) See All
zLinux64	File 1, File 2 (2.3 GB) See All	

Oracle Database Gateways 12c Release 1 (12.1.0.2.0) for Linux x86-64

 [linuxamd64_12c_gateways.zip](#) (676,459,016 bytes) (cksum - 903640604)

Contains the Oracle Database Gateways to non-Oracle Databases. Download if you want set up a heterogeneous data integration environment

Abb. 3: Download Gateway-Software über OTN

Auch im MOS wird man auf diese Seite verlinkt: „How to Download Oracle Database Gateway Products (Doc ID 1270959.1)“:

<https://support.oracle.com/epmos/faces/DocumentDisplay?id=1270959.1>

In den weiteren Teilen der Dokumentation findet man dann genauere Angaben für die Installation, unter anderem die Hardware-Voraussetzungen. Für die Teradata ist dies Teil sechs:

Hardware Items	Required for Linux x86 64 bit
Temporary Disk Space	400 MB
Disk Space	750 MB
Physical Memory*	512 MB
Swap Space	1 GB
Processor	x86_64

Abb. 4: Hardware-Voraussetzungen für Teradata-Gateway unter Linux

Herunterladen der benötigten Oracle-Dateien

Die Dateien können wie oben beschrieben über OTN geladen werden. Alternativ ist auch ein Download über MOS möglich:



Abb. 5: Download Gateway-Software über MOS – Latest Patchsets

Die heruntergeladenen Installationsdateien werden lokal abgelegt und entpackt. Als Ziel-Verzeichnis wird das Standard-Verzeichnis „/u01“ verwendet, als User „oracle“. Dies entspricht den Standard und ist erprobt. Das Verzeichnis wird ggf. angelegt und die Rechte an den Oracle-Software-Owner gegeben:

```
mkdir /u01
chown -R oracle:oinstall /u01

cd /INSTALL/gateway
unzip p21419221_121020_Linux-x86-64_8of10.zip
```

Abb. 6: Entpacken der Installationsdateien

Installation

Vor der Installation und bei der Verwendung einer virtuellen Maschine wird empfohlen, einen Snapshot zu machen. So kann man bei auftretenden Fehlern leicht wieder zurück und den Installationsprozess wiederholen.

Libraries

Oracle benötigt diverse Libraries. Dies sind für den Gateway-Server die gleichen wie für viele andere Installationen (z. B. Datenbank). Wenn man den Installer durchklickt (man muss Angaben zum Teradata-Treiber machen, er muss aber noch nicht installiert sein), erhält man ein Bild dieser Art:

Checks	Status	Fixable
Checks		
Swap Size	Warning	No
Packages		
Package: gcc-4.3 (x86_64)	Warning	No
Package: gcc-c++-4.3 (x86_64)	Warning	No
Package: libaio-devel-0.3.104	Warning	No
Package: libstdc++-devel-4.3 (x86_64)	Warning	No
Package: libstdc++43-devel-4.3.3_20081022	Warning	No
Package: glibc-devel-2.9	Warning	No

Abb. 7: Fehlende Libraries auf der virtuellen Maschine

In diesem Fall installiert man diese Packages (bei SLES mittels zypper) als root:

- gcc
- gcc-c++
- gcc43-c++
- gcc-locale
- gcc43-locale
- libaio-devel
- libstdc++-devel

Einzelne Packages wie libstdc werden dabei automatisch mitgezogen.

Teradata ODBC Treiber

Die Teradata-Treiber sollten vor dem Gateway installiert werden. Darauf weist auch die Oracle-Dokumentation hin. Bei der Gateway-Installation muss nämlich der „TD_ICU_DATA Path“ angegeben werden. Die Treiber können von Teradata heruntergeladen werden: <http://tays.teradata.com>
Hierfür wird ein Account benötigt!

Damit das Zusammenspiel zwischen Teradata-Treiber und Oracle-Gateway funktioniert, verwenden wir „Installation Bundle TTU 15.00“. Diese Version setzt Pfade in „etc/profile“ noch analog zur Oracle-Anleitung (TD_ICU_DATA, COPLIB und COPERR). Spätere Versionen setzten diese nicht.

In unserer Umgebung konnten wir keine spätere Version lauffähig bekommen!

Das Zip-Archiv wird nach /opt kopiert und entpackt.

```
cd /opt
cp /INSTALL/teradatatools/ttu_bundle_1500_linux64.zip /opt
unzip ttu_bundle_1500_linux64.zip
tar -xzf TeradataToolsAndUtilitiesBase__linux_indep.15.00.10.00.tar.gz
rm *.gz *.txt *.zip

cd TeradataToolsAndUtilitiesBase
./setup.bat
8 9          # 8 für CLIV2
rm -rf TeradataToolsAndUtilities
```

Abb. 8: Installation Teradata-Libraries

Der Pfad /opt ist hardcoded in setup, deshalb ist die Installation nur unter /opt möglich.

Oracle-Gateway

Die Installation wird als „Software Owner“ durchgeführt, also unter unserem Lieblingsuser namens „oracle“ ☺. Nachdem die Dateien bereits entpackt sind, braucht man zum Starten der Installation nun einen x-Server. Über die Display-Variablen kann man die grafische Ausgabe umleiten. Im beschriebenen Fall passiert die Umleitung auf eine andere virtuelle Maschine (xserver). Ein einfacher Weg ohne zusätzlichen Server ist Xming. Hier fungiert der eigene PC als X-Window-System.

Die Sprache der Installation wird über die LANG-Variablen geregelt. Empfohlen ist hierbei Englisch, da die deutschen Übersetzungen manchmal nicht zu verstehen sind.

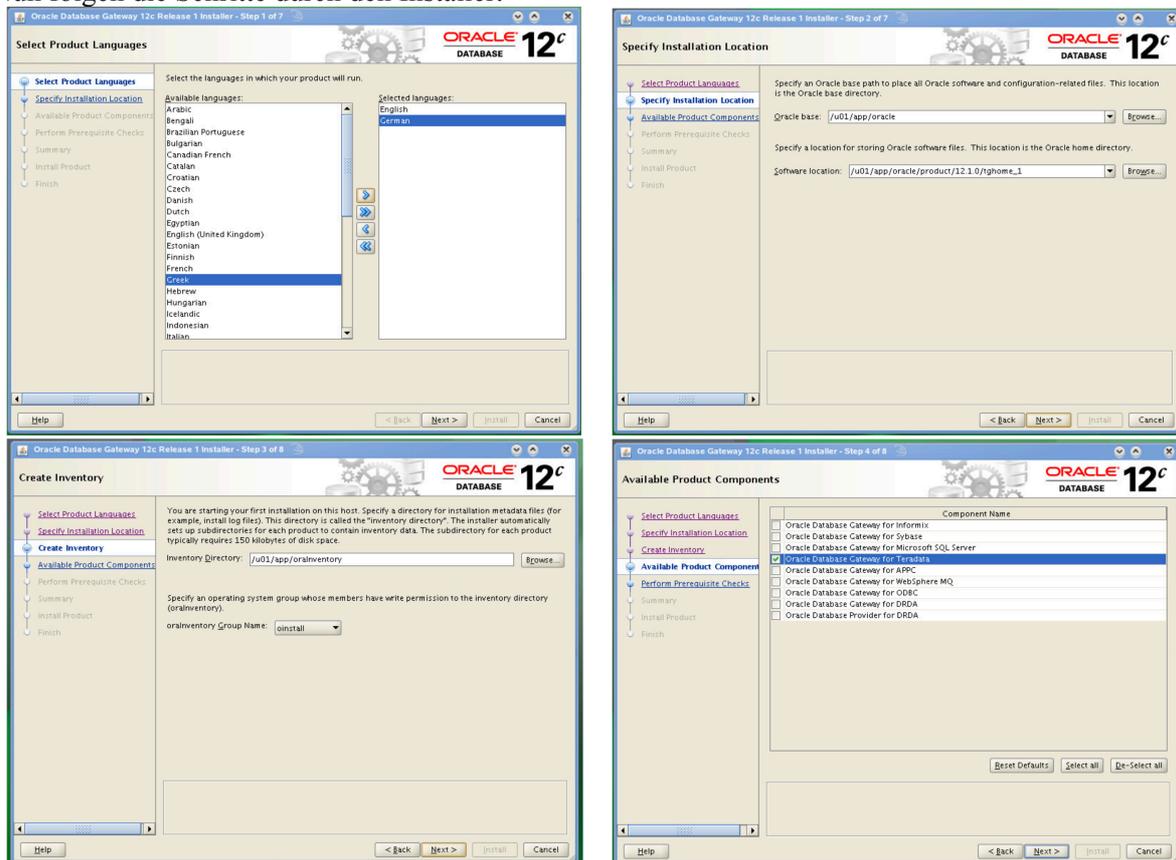
```
export DISPLAY=xserver:0
cd /INSTALL/gateway/gateways
```

```
echo $LANG
export LANG=en_US
./runInstaller
```

Abb. 9: Starten der Gateway-Installation

Die LANG-Variablen kann überschrieben werden. Wenn z. B. LC_ALL gesetzt ist, zieht dieser Wert und man müsste hier die Anpassung vornehmen.

Nun folgen die Schritte durch den Installer:



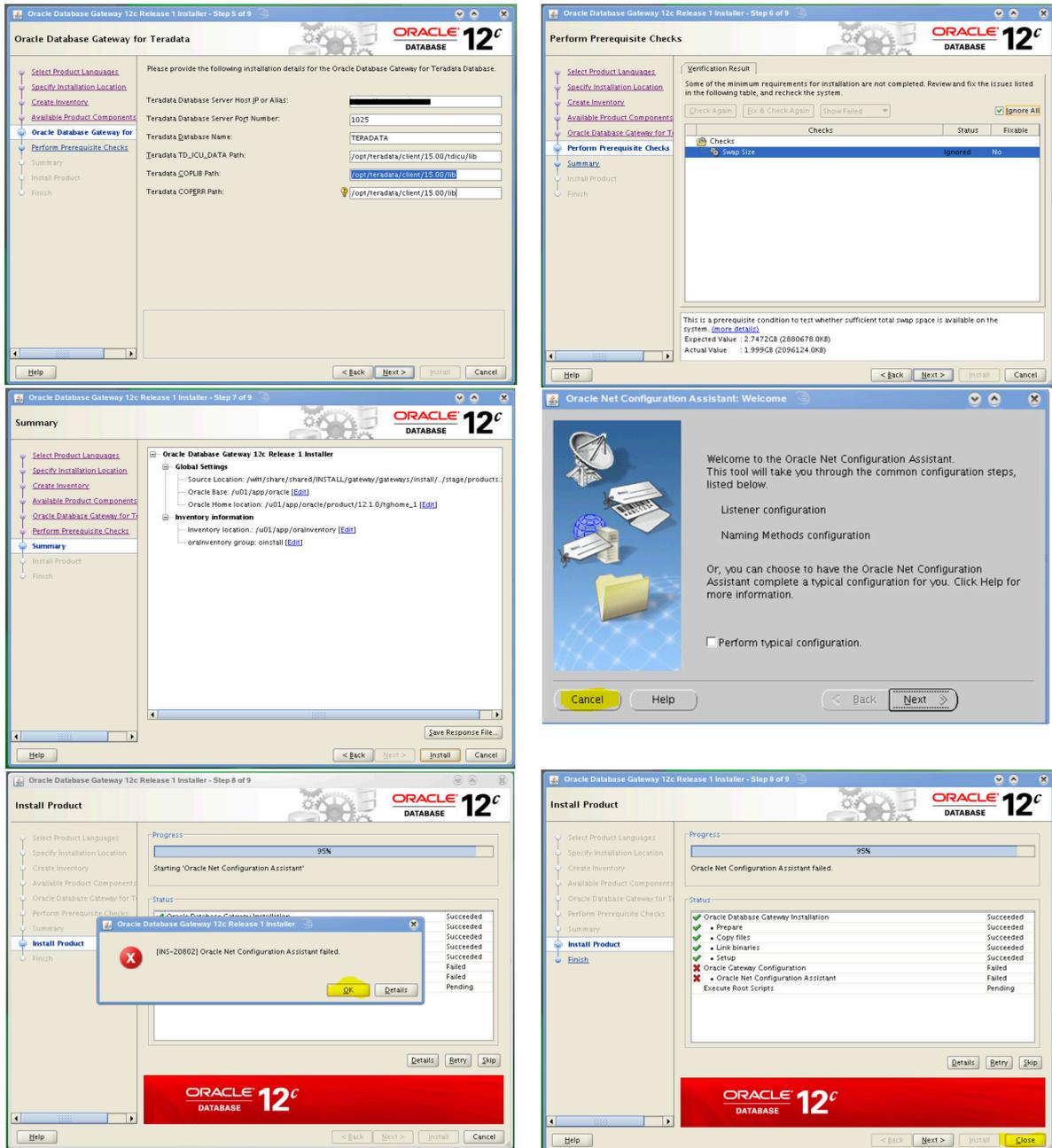


Abb. 10: Schritte durch den Installer

Konfiguration Allgemein

Es folgt nun die Konfiguration des Gateways, Die Oracle-Dokumentation dazu findet sich hier:

<http://docs.oracle.com/database/121/OTGIS/configteradata.htm#OTGIS700>

Hilfreich ist hierfür auch die MOS-Anleitung der 11g-Version, „How to Setup DG4Tera (Oracle Database Gateway for TeraData) on 64bit Unix (Doc ID 823534.1)“:

<https://support.oracle.com/epmos/faces/DocumentDisplay?id=823534.1>

In unserem Fall wollen wir auf zwei Teradata-Datenbanken (Entwicklung & Test) zugreifen können. Da für jede Teradata-Datenbank ein eigener „Gateway System Identifier (SID)“ benötigt wird, reicht der Standard-SID dg4tera nicht aus. Für unsere Umgebung wählen wir daher neue, andere SIDs:

- dg4teraOVPROD
- dg4teraOVDEV

Der Einfachheit halber werden zwei ORACLE-Variablen gesetzt. Diese kann man gleich ins Profil vom User oracle schreiben. Den Pfad und die Variable LD_LIBRARY_PATH können hier auch mitgesetzt werden. Das wird später bei Oracle Net benötigt.

```
export ORACLE_BASE=/u01/app/oracle
export ORACLE_HOME=$ORACLE_BASE/product/12.1.0/tghome_1
export PATH=$ORACLE_HOME/bin:$PATH
export LD_LIBRARY_PATH=$ORACLE_HOME/lib:$LD_LIBRARY_PATH
```

Abb. 11: Umgebungsvariablen

Konfigurationsdateien

Die erzeugte Standard-Parameter-Datei enthält bereits die Einträge, die in der Installationsmaske eingegeben wurden. Die Datenbank und die eckigen Klammern um den Host-Alias (wir nur für IPV6 benötigt) müssen noch manuell entfernt werden.

```
cd $ORACLE_HOME/dg4tera/admin
cat initdg4tera.ora

# This is a customized agent init file that contains the HS parameters
# that are needed for the Database Gateway for Teradata
#
# HS init parameters
#
HS_FDS_CONNECT_INFO=[teradataPROD.subdomain.domain.de]:1025/TERADATA
HS_FDS_TRACE_LEVEL=OFF
HS_FDS_RECOVERY_ACCOUNT=RECOVER
HS_FDS_RECOVERY_PWD=RECOVER
#
# Environment variables required for the non-Oracle system
#
set TD_ICU_DATA=/opt/teradata/client/15.00/tdicu/lib
set COPLIB=/opt/teradata/client/15.00/lib
set COPERR=/opt/teradata/client/15.00/lib
```

Abb. 12: Konfigurationsdatei initdg4tera nach der Installation

In unserem Beispiel haben wir bei der Installation die Produktivdatenbank (Teradata) angegeben. Daher kann die Datei gleich verwendet werden. Wir kopieren die Datei, damit sie den richtigen Namensaufbau hat. Dieser ist: init[SID].ora. Dies kann auch gleich für die Entwicklungs-Teradata gemacht werden. Dann ändert man die gewünschten Werte ab:

```
cd $ORACLE_HOME/dg4tera/admin
cat initdg4tera.ora

cp initdg4tera.ora initdg4teraOVDEV.ora
cp initdg4tera.ora initdg4teraOVPROD.ora

vi initdg4teraOVDEV.ora
vi initdg4teraOVPROD.ora

diff initdg4tera.ora initdg4teraOVPROD.ora
< HS_FDS_CONNECT_INFO=[teradataPROD.subdomain.domain.de]:1025/TERADATA
---
> HS_FDS_CONNECT_INFO=teradataPROD.subdomain.domain.de

diff initdg4tera.ora initdg4teraOVDEV.ora
< HS_FDS_CONNECT_INFO=[teradataPROD.subdomain.domain.de]:1025/TERADATA
---
> HS_FDS_CONNECT_INFO=teradataDEV.subdomain.domain.de
```

Abb. 13: Änderungen in den initdg4tera-Konfigurationsdateien

Oracle Net Listener

Nun benötigt man einen Listener für das Gateway. Eine Beispiel-Konfiguration ist unter \$ORACLE_HOME/dg4tera/admin/listener.ora.sample bzw. \$ORACLE_HOME/dg4tera/admin/tnsnames.ora.sample. Diesen kann man als Vorlage verwenden und anpassen.

Nun wird der Listener gestartet:

```
lsnrctl stop  
lsnrctl start
```

```
lsnrctl status
```

```
LSNRCTL for Linux: Version 12.1.0.2.0 - Production on 09-NOV-2016  
14:43:54  
Copyright (c) 1991, 2014, Oracle. All rights reserved.  
Connecting to (ADDRESS=(PROTOCOL=tcp) (HOST=db-hogs) (PORT=1521))  
STATUS of the LISTENER  
-----  
Alias                LISTENER  
Version              TNSLSNR for Linux: Version 12.1.0.2.0 - Production  
Start Date           09-NOV-2016 14:43:48  
Uptime               0 days 0 hr. 0 min. 6 sec  
Trace Level          off  
Security             ON: Local OS Authentication  
SNMP                 OFF  
Listener Parameter File  
/u01/app/oracle/product/12.1.0/tghome_1/network/admin/listener.ora  
Listener Log File  
/u01/app/oracle/diag/tnslnr/db-hogs/listener/alert/log.xml  
Listening Endpoints Summary...  
  
(DESCRIPTION=(ADDRESS=(PROTOCOL=tcp) (HOST=db-  
hogs.subdomain.domain.de) (PORT=1521)))  
Service Summary...  
Service "dg4teraOVDEV" has 1 instance(s).  
  Instance "dg4teraOVDEV", status UNKNOWN, has 1 handler(s) for this  
service...  
Service "dg4teraOVPROD" has 1 instance(s).  
  Instance "dg4teraOVPROD", status UNKNOWN, has 1 handler(s) for this  
service...  
The command completed successfully
```

Abb. 14: Listenerstatus

Datenbank

Nun muss das Gateway auf dem Datenbankserver, der die Verbindung mit der Teradata herstellen können soll, bekannt gemacht werden. Zum Abfragen wird später ein DB-Link verwendet. Um diesen bequem anlegen zu können, macht man das Gateway in der tnsnames.ora des DB-Servers bekannt:

```
# Teradata Testumgebung  
DB-HOGS_TERA_TEST =  
  (DESCRIPTION=  
    (ADDRESS=(PROTOCOL=tcp) (HOST=db-hogs) (PORT=1521))  
    (CONNECT_DATA=(SID=dg4teraOVDEV))  
    (HS=OK)  
  )
```

Abb. 15: Beispiel für einen Eintrag in der tnsnames.ora

Über das Kommando „tnsping“ kann man überprüfen, ob die Syntax richtig ist:
tnsping DB-HOGS_TERA_TEST

Nun fehlen noch das Erstellen eines DB-Links und das Testen der Verbindung:

```
create public database link TERATEST connect to MY_TERA_USER identified by
„geheimesPasswort“ using ‚DB-HOGS_TERA_TEST‘ ;

select user from dual@TERATEST;
```

Abb. 16: DB-Link erstellen und testen

Überwachung und Autostart des Listeners

Überwachung

Da auf dem Server keine GI oder ähnliches installiert wird, begnügen wir uns in diesem Fall mit einer einfachen Prüfung über crontab. Dazu wird ein Shell-Skript regelmäßig aufgerufen und informiert (z. B. per Mail), sollte der Listener nicht laufen.

Autostart

Um den Listener bei jedem Systemstart gleich mit hochzufahren, gibt es ein kleines Wrapper-Skript für die Kontrolle des DG-Listeners. Dies versteht folgende Parameter:

- start
- stop
- status
- reload
- force-reload
- restart

Damit kann es per init-Skript beim Systemstart aufgerufen werden. Da sich der Server im Runlevel 3 befindet (die Bash ist schließlich grafisch genug ☺), linkt man das Skript in das entsprechende Runlevel-Verzeichnis.

Kontaktadresse:

Alexander Pilfusek
Witt-Gruppe (a member of the otto group)
Schillerstraße 4-12
D-92637 Weiden i. d. OPf.

Telefon: +49 (0) 961-400 1099
Fax: +49 (0) 961-400 1580
E-Mail: alexander.pilfusek@witt-gruppe.eu
Internet: www.witt-gruppe.eu