Tipps & Tricks: April 2003

Bereich:	DBA	Erstellung:	04/2003 MP
Versionsinfo:	8.2 9.2 10.2 11.1	Letzte Überarbeitung:	05/2009 BK

Local Naming

Namen sind Schall und Rauch - wie sich Datenbanken finden...

Eine Datenbank kommt selten allein - manchmal steht sie im Verbund mit anderen Datenbanken und fast immer ist sie von verschiedenen Client-Applikationen umgeben. Diese Clients greifen von diversen Rechnern auf die Datenbank zu. Es gibt verschiedene Verfahren, um die für den Verbindungsaufbau erforderlichen Informationen bereitzustellen. In diesem Artikel wird das häufig eingesetzte Local Naming mit seinen Steuerdateien beschrieben. Alle benötigten Informationen werden in vereinheitlichter Form in speziellen Dateien bereitgestellt. Hier ein paar Gedanken zu Syntax und Semantik dieser Dateien und zur Verbindungsaufbausteuerung.

Überblick

Im Prinzip ist alles ganz einfach. Soll eine Verbindung zur Datenbank aufgebaut werden, wird ein Datenbank-User benötigt, der mindestens über das create session Recht verfügt. Der Name sowie das Passwort dieses Users werden dem Connect-Request mitgegeben. Außerdem ein Alias-Name (net service name) für die Zieldatenbank. Die Client-Anwendung sendet nun einen Verbindungsaufbau-Request an eine bestimmte physikalische Adresse. Dort läuft (hoffentlich) ein entsprechendes Dämon-Programm (Listener), welches auf solche Verbindungsaufbau-Requests wartet. Der Listener stellt nun eine Verbindung zwischen dem rufenden Client und einem der Server-Prozesse der Zieldatenbank her. Je nach Art der Verbindung kann dabei ein Datenbankserver-Prozeß mehrere Clients gleichzeitig (shared Server) oder auch nur einen Client exklusiv bedienen. Der Listener startet die benötigten Datenbank-Server-Prozesse. Nach dem Verbindungsaufbau wird der Listener dann nicht mehr benötigt und steht zur Bearbeitung weiterer Verbindungsaufbau-Requests zur Verfügung.

Das Mapping zwischen Zieldatenbank-Alias (net service name) und physikalischer Adresse des Listeners wird auf jedem Client in einer Datei mit dem Namen tnsnames.ora hinterlegt. Diese Datei befindet sich normalerweise im Verzeichnis $ORACLE_HOME/network/admin. Davon abweichende Verzeichnisnamen können in der Umgebungsvariablen TNS_ADMIN hinterlegt werden. Da zeigt sich auch schon ein Nachteil dieser "Local Naming" Namensauflösungsmethode - Änderungen (z.B. die Aufnahme neuer Datenbanken) müssen auf allen Clients eingepflegt werden.

tnsnames.ora

Die wohl bekannteste unter den Steuerdateien. Sie beinhaltet das Mapping zwischen "net service name" und "listener protocol address". Die "listener protocol address" enthält alle Informationen, die für den Verbindungsaufbau zum Listener benötigt werden. Das sind neben der Internet-Adresse der Zielmaschine auch der Port (auf dem der Listener horcht) und das zu verwendende Protokoll (z.B. TCP).

Die tnsnames.ora enthält für jede Zieldatenbank einen Eintrag der folgenden Form:

```
<net_service_name> =                          -- Alias für die Datenbankverbindung
(DESCRIPTION_LIST=                            -- Liste von Verbindungsdeskriptoren (nicht
mit OracleNetManager)
DESCRIPTION=                                 -- Verbindungsdeskriptor-Container
  (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION)) -- Protokoll-Adresse eines Listeners
  (ADDRESS=(PROTOCOL_ADDRESS_INFORMATION))
```

```
      ...
  (CONNECT_DATA=(SERVICE_NAME=<service_name>))
  )
   (DESCRIPTION=
  )
)
```

Wenn mehrere Listener-Protokolladressen für einen net_service_name angegeben werden, wird je nach eingestellter LoadBalancing- und Failover-Strategie entweder sequentiell eine Adresse nach der anderen oder zufällig die jeweils nächste Listener-Protokolladresse ausgewählt. Es ist sogar möglich, mehrere Adress-Listen mit verschiedenen LoadBalancing- und Failover-Verhalten zu definieren.

Beispiel für eine Protokoll-Adresse:

```
(ADDRESS=
  (PROTOCOL=tcp)
  (HOST=<hostname>)
  (PORT=1521)
)
```

listener.ora

Das "Gegenstück" zur tnsnames.ora stellt diese Datei auf der Datenbank-Server Maschine dar. Diese Parametrisierungsdatei enthält zwei Arten von Information. Zum einen wird in einer sogenannten "protocol address section" definiert, auf welcher physikalischen Adresse ein oder mehrere Listener horchen (Verbindungsaufbau-Requests akzeptieren) sollen. Zum anderen wird in einer sogenannten "static service registration section" festgelegt, welche Datenbanken mit welchen Service-Namen vom Listener bearbeitet werden. Für Datenbanken ab Version 8i aufwärts ist die zweite Sektion (static service registration section) nicht mehr erforderlich, weil sich jede Instanz beim Start automatisch bei allen Listenern auf der Maschine anmeldet. Soll allerdings die Überwachung mit OracleEnterpriseManager erfolgen, ist die "static service registration" wieder erforderlich.

Beispiel:

protocol address section:

```
LISTENER=
(
   (DESCRIPTION=
      (ADDRESS_LIST=
         (ADDRESS=(PROTOCOL=tcp)(HOST=<hostname>)(PORT=1521))
         ...
      )
   )
)
```

Static service registration section:

```
SID_LIST_LISTENER=
(
   (DESCRIPTION=
      (SID_LIST=
         (SID_DESC=
            (GLOBAL_DBNAME=<global_db_name>)
```

```
            (ORACLE_HOME=<pfadangabe>)
            (SID_NAME=<sid>)
         )
         ...
      )
   )
)
```

hinter dem global_db_name verbirgt sich eine systemweit eindeutige Datenbankbezeichnung:

GLOBAL DATABASE NAME

Jede Datenbank hat einen Namen. Über diesen sogenannten "global database name" ist sie eindeutig identifizierbar. Dieser Name besteht aus zwei Komponenten und wird bereits bei der Erzeugung der Datenbank aus dem Inhalt der beiden Initialisierungsparameter DB_NAME und DB_DOMAIN zusammengesetzt und im DataDictionary der Datenbank hinterlegt. Diese Initialisierungs-Parameter werden dann bei späteren Datenbankstarts nicht mehr berücksichtigt - nachträgliche Änderungen dieser Parameter in der Initialisierungsdatei bleiben also unberücksichtigt. Eine Änderung des globalen Datenbanknamens ist später nur noch über den SQL-Befehl

```
ALTER DATABASE RENAME GLOBAL_NAME TO <newname>
```

möglich. Der aktuelle globale Datenbankname kann aus dem DataDictionary der Datenbank ausgelesen werden:

```
select * from global_name;
```

sqlnet.ora

Die dritte Parameterdatei im Zusammenhang mit dem Verbindungsaufbau enthält neben Voreinstellungen von Authentifizierungsmethoden und Default-DB-DOMAIN-Erweiterung von net service names vor allem detaillierte Tracing- und Logging-Einstellungen:

`LOG_DIRECTORY_CLIENT`
Zielverzeichnis für Client-Logfile

`LOG_DIRECTORY_SERVER`
Zielverzeichnis für Server-Logfile

`LOG_FILE_CLIENT`
Name für Client-Logfile

`LOG_FILE_SERVER`
Name für Server-Logfile

`NAMES.CONNECT_TIMEOUT (default 3)`
Sekunden Wartezeit für Verbindungen zum ORACLE_NAMES_SERVER

`NAMES.DCE_PREFIX`
Spezifiziert "distributed computing environment"-Cell-Name für name-Lookups

`NAMES.DEFAULT_DOMAIN`
Dieser Suffix wird automatisch an alle unqualifizierten net service names angehängt (qualifizierte namen bleiben unverändert)

`NAMES.DIRECTORY_PATH = (tnsnames, onames, hostname)`
Reihenfolge der zur Namensauflösung verwendeten Naming-Methoden

`SQLNET.EXPIRE_TIME`
Zeitintervall in Sekunden zur Prüfung, ob Client/Server-Verbindungen noch aktiv sind (Server Prozess wird beendet, falls die Verbindung beendet wurde oder nicht mehr aktiv ist)

`SQLNET.AUTHENTICATION_SERVICES = (NTS)`
Authentifizierung bei der Datenbank wird über das Betriebssystem abgewickelt

`TCP.EXCLUDED_NODES`
verhindert, dass die aufgeführten Clients eine Datenbankverbindung erhalten

`TCP.INVITED_NODES`
bewirkt, dass nur die aufgeführten Clients eine Datenbankverbindung erhalten

`TCP.VALIDNODE_CHECKING`
schaltet die Prüfung ein

`TNSPING.TRACE_DIRECTORY`
Zielverzeichnis für tnsping-Trace

`TNSPING_TRACE_LEVEL off, user, admin, support`
Trace-Level für tnsping-Trace

`TRACE_DIRECTORY_CLIENT`
Zielverzeichnis Client

`TRACE_DIRECTORY_SERVER`
Zielverzeichnis Server

`TRACE_FILE_CLIENT`
Dateiname

`TRACE_FILE_SERVER`
Dateiname

`TRACE_FILEN_CLIENT`
Größe in KB

`TRACE_FILEN_SERVER`
Größe in KB

`TRACE_FILENO_CLIENT`
Anzahl Trace-Dateien Client

`TRACE_FILENO_SERVER`
Anzahl Trace-Dateien Server

`TRACE_LEVEL_CLIENT off, user, admin, support`
Trace-Level für Clienttrace

`TRACE_LEVEL_SERVER off, user, admin, support`
Trace-Level für Servertrace

`TRACE_TIMESTAMP_CLIENT on, off`
schaltet Ausgabe von Zeitstempel fuer jedes Trace-Event f. Client

`TRACE_TIMESTAMP_SERVER on, off`
schaltet Ausgabe von Zeitstempel fuer jedes Trace-Event f. Server

`USE_CMAN true, false`
Schaltet Routing fuer die Clients auf die Protokolladresse des OracleConnectionManagers (wird am Port 1630 erkannt)

`USE_DEDICATED_SERVER on, off`
steuert, ob bei jedem Verbindungs-Request SERVER=DEDICATED gesetzt wird

`WALLET_LOCATION`
Lokation von WALLETS: SSL Zertifikate, Keys, Trustpoints