

Installation

Gehören Sie auch zu der Sorte Leute, die den neuen DVD-Spieler zunächst anschließen und sich den ersten Film anschauen, bevor Sie Wochen später die Anleitung lesen und überrascht über die vielen Funktionen sind? Im Prinzip würde es hier analog funktionieren, die Software ist schnell heruntergeladen und mit wenigen Mausklicks installiert. Dennoch gibt es auch hier vieles zu entdecken und ich hoffe, auf den nächsten Seiten einige interessante Informationen bereitstellen zu können.

Mit Oracle Express stellt Oracle eine frei verfügbare Datenbank zur Verfügung, für die keinerlei Lizenzkosten zu entrichten sind. Dafür muss man allerdings einige Einschränkungen in Kauf nehmen.

CPU-Beschränkung

Oracle Express nutzt lediglich eine CPU je Rechner. Man kann die Software zwar auf Mehrprozessor-Systemen installieren, effektiv genutzt wird allerdings immer nur eine davon.

Installation und Ausführung der Datenbank

Je Rechner kann nur eine Instanz von Oracle Express betrieben werden. Oracle Express kann aber sehr wohl parallel zu anderen (lizenzpflichtigen) Oracle-Datenbanken wie der Standard oder Enterprise Edition betrieben werden.

Hauptspeicher

Oracle Express nutzt maximal 1 GB an Hauptspeicher. Dieser setzt sich – wie Sie in den folgenden Kapiteln noch kennen lernen werden, aus der *SGA* (*System Global Area*) und der *PGA* (*Program Global Area*) zusammen.

Plattenspeicher

Oracle Express kann maximal 4 GB an Festplattenspeicher zur reinen Datenspeicherung nutzen. An dieser Stelle darf ich darauf hinweisen, dass Sie unbedingt die Produktiv-Version der Datenbank nutzen sollten. Bevor diese veröffentlicht wurde, gab es schon zu Testzwecken eine Beta-Version. Diese hatte allerdings noch das Problem, dass bei der Kalkulation des Festplattenspeichers auch die TEMP- und UNDO-Tablespaces (auf die im Folgenden noch genauer eingegangen wird) mitgerechnet wurden, sodass effektiv weniger Platz zur Verfügung stand.

Hinweis

Unter Oracle Dokument B32391-02 »Oracle Database Express Edition Upgrade Guide« finden Sie eine Anleitung, wie man die Express Edition mit Hilfe des Oracle Database Upgrade Assistant in eine vollwertige (und lizenzpflichtige) Oracle-Datenbank überführt, um sich damit der oben beschriebenen Restriktionen zu entledigen.

Vor der Kür steht bekanntlich die Pflicht. Die beginnt mit dem Herunterladen der Software. Dabei stehen mehrere Pakete auf der Oracle-Website zur Verfügung.

Die Software steht derzeit für die diversen Windows-Betriebssysteme als auch für die gängigsten Linux-Derivate zum Download bereit. In einigen Diskussionsforen wird auch schon über eine Mac-Variante spekuliert. Die auf den folgenden Seiten beschriebene Installation bezieht sich auf ein Windows-System.

Sowohl für Windows als auch für Linux werden nun drei verschiedene Module zum Download angeboten:

- Oracle Database 10g Express Edition (Western European)
- Oracle Database 10g Express Edition (Universal)
- Oracle Database 10g Express Client

Oracle Database 10g Express Edition (Western European)

Diese Version beinhaltet eine Datenbank, die mit dem Ein-Byte-Zeichensatz LATIN1 (WE8MSWIN1252) angelegt wird. Für jedes Zeichen innerhalb der Datenbank wird also genau ein Byte verbraucht. Natürlich ist es damit möglich, alle gängigen westeuropäischen Zeichensätze zu speichern.

Der größte Vorteil dieser Version dürfte wohl die geringere Größe der Download-Datei sein. Es gibt allerdings einen entscheidenden Nachteil: Nach erfolgreicher Installation der Datenbank werden Sie feststellen, dass die gesamte Administration über ein Web-Interface im Browser erfolgt. Dieses Web-Interface ist bei dieser Version immer in Englisch. Fehlermeldungen der Datenbank werden aber durchaus in der Landessprache ausgegeben.

Oracle Database 10g Express Edition (Universal)

Diese Version verwendet einen sogenannten Unicode-Zeichensatz (AL32UTF8). Dieser ermöglicht das Speichern von Texten, bei denen ein Buchstabe in mehr als einem Byte kodiert ist. Dies ist vorwiegend im asiatischen Raum zu finden; die koreanische und die chinesische Sprache kennen einfach mehr als 255 Zeichen.

Hier schlägt der Download mit rund 250 MB zu Buche, dafür erhält man auch das Web-Interface in der jeweiligen Landessprache. Damit steht dem Administrations-tool auf Chinesisch nichts mehr im Weg.

Oracle Database 10g Express Client

Hierin verbirgt sich die Client-Software, die auf den Rechnern benötigt wird, von denen aus auf einen Oracle-Server innerhalb eines Netzwerks zugegriffen werden soll. Die Client-Software ist in der Server Edition mit enthalten, die Installation der Express Edition – sei es Western European oder Universal – erfordert somit keine separate Installation mehr.

Abhängig von dem Betriebssystem (oder zusätzlich installierter Firewalls), auf dem Sie den XE-Server betreiben, kann es notwendig werden, bestimmte Ports für den Zugriff von außen zuzulassen. Standardmäßig verwendet Oracle XE die folgenden Ports:

- 1521: Listener Prozess der Datenbank
- 2030: Oracle-Service für den Microsoft Transaction Service
- 8080: http-Port für das Web-Administrationstool

2.1 Technische Voraussetzungen

Oracle Express für Windows wird installiert auf einem der folgenden Betriebssysteme (eine Version für Windows Vista stand zum Zeitpunkt der Drucklegung des Buches noch nicht zur Verfügung, ich empfehle regelmäßig die Seite www.oracle.com/technology/x zu konsultieren):

- Windows 2000 Service Pack 4 oder höher
- Windows Server 2003
- Windows XP Professional Service Pack 1 oder höher

Der Speicherbedarf schlägt mit rund 1,5 GB Festplattenplatz zu Buche. 256 MB Hauptspeicher sollten auf dem System mindestens vorhanden sein. Hier gilt die Devise, je mehr desto besser, allerdings: Oracle Express nutzt maximal 1 GB Hauptspeicher, wie oben beschrieben.

Wie schon erwähnt, erfolgt die Administration der Datenbank über ein Web-Interface. Voraussetzung dafür ist, dass Sie einen der folgenden Browser in der beschriebenen oder einer höheren Version einsetzen:

- Microsoft Internet Explorer 6.0
- Netscape Navigator 7.2
- Mozilla 1.7
- Firefox 1.0

2.2 Die Installation

Im Folgenden wird die interaktive Installation, d.h. mittels des grafischen Installationsprogramms, auf einem Windows-System beschrieben. Sie als Anwender bzw. Datenbankadministrator müssen während des Installationsvorgangs einige Fragen beantworten bzw. bestimmte Parameter eingeben.

Darüber hinaus gibt es noch die Möglichkeit einer »silent«-Installation. Hierbei werden in einer Konfigurationsdatei im Prinzip alle Parameter hinterlegt, die man ansonsten interaktiv eingeben müsste. Diese Art der Installation ist vor allem dann interessant, wenn Oracle Express auf mehreren bis hin zu vielen Systemen installiert werden soll und die erforderlichen Parameter immer die gleichen sind.

Hinweis

Ich gehe davon aus, dass Sie Oracle Express auf einem System installieren, auf dem noch keine weitere Software von Oracle in Betrieb ist. Sollte das der Fall sein, stellen Sie bitte sicher, dass die Umgebungsvariable `ORACLE_HOME` zum Zeitpunkt der Installation nicht gesetzt ist.

Wichtig

Sie müssen für die Installation mit einem Benutzer am (Windows-) System angemeldet sein der über Administratorrechte verfügt.

Die Installation des Servers ist denkbar einfach. Auf der beiliegenden Buch-CD finden Sie alle Dateien, die zur Installation notwendig sind. Neben der Universal Edition finden Sie dort auch die kleinere Western European im Order *Oracle-Software*.

Starten Sie einfach die entsprechende ausführbare Datei direkt von der CD, beispielsweise `OracleXEUniv.exe` für die Universal Edition. Die Datei wird zunächst entpackt, bevor die eigentliche Installationsroutine gestartet wird (vgl. Abbildung 2.1).

Im nächsten Fenster legen Sie fest, wo die Datenbank installiert werden soll. Wenn wir hier von »der Datenbank« sprechen, so ist damit sowohl die Software gemeint, die zum Betrieb von Oracle Express notwendig ist, als auch die Datendateien, in denen später die eigentlichen Nutzdaten in Form von Tabellen abgelegt sind. Die eigentliche Datenbanksoftware wird in einem Unterorder `\app` angelegt, die Datendateien im Unterorder `\oradata`. Standardmäßig schlägt der Installations-Assistent hier den Ordner `c:\oracleexe` vor, der jedoch ohne Weiteres überschreibbar ist. Im vorliegenden Fall habe ich es bei dem Vorschlag belassen, alle folgenden Ausführungen beziehen sich also auf das Standardverzeichnis.

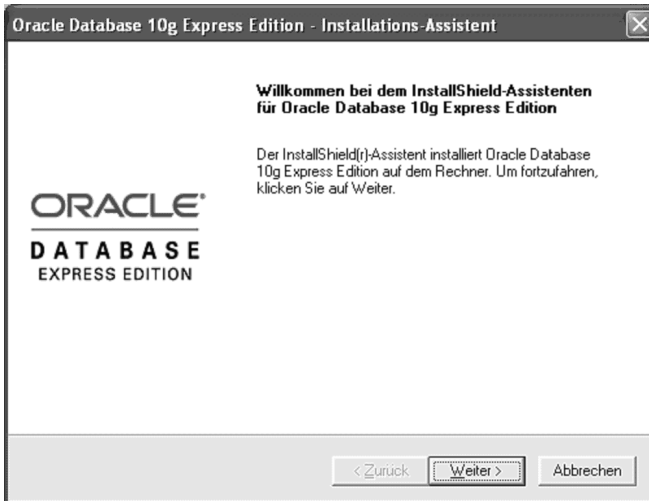


Abb. 2.1: Der Installations-Assistent von Oracle Express

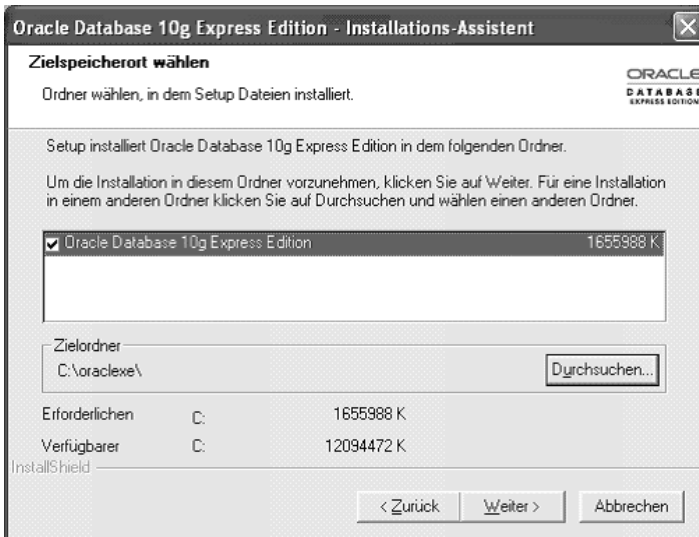


Abb. 2.2: Zielordner der Datenbank festlegen

Über *Weiter* gelangen Sie in einen weiteren Dialog, in dem das Kennwort für zwei Datenbankbenutzer definiert wird. Ein solcher Datenbankbenutzer wird auch als

Account bezeichnet. Innerhalb von Oracle nehmen dabei die Accounts SYS und SYSTEM eine besondere Bedeutung ein, auf die im Folgenden noch näher eingegangen wird. Merken Sie sich dieses Passwort gut, Sie werden es später noch benötigen!

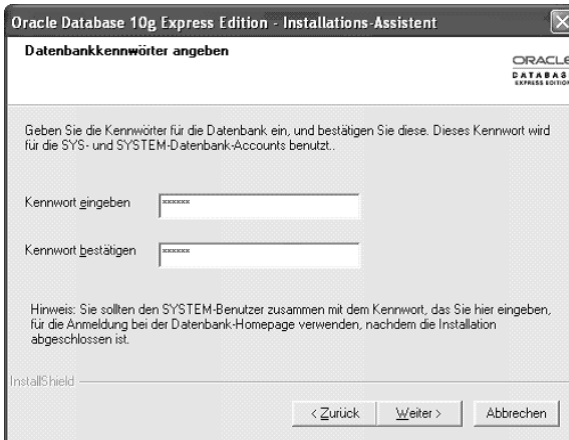


Abb. 2.3: Passwort für den SYS- und SYSTEM-Account

Damit sind alle notwendigen Parameter zur Installation erfasst. Die Abbildung 2.4 zeigt nochmals die Zusammenfassung. Über *Installieren* wird die eigentliche Installation gestartet.

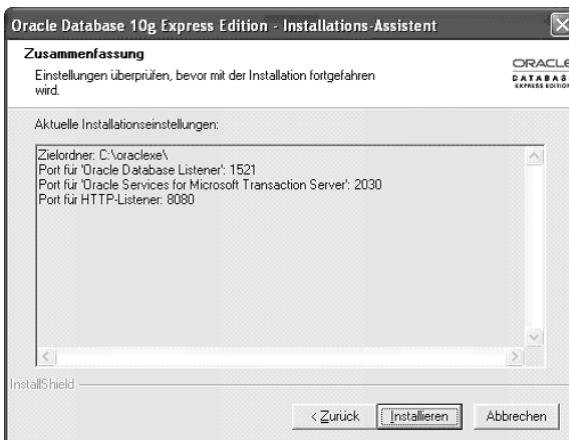


Abb. 2.4: Zusammenfassung der Installationsparameter

Nach Fertigstellung dieser können Sie direkt das Web-Interface starten, denn nach der Installation werden automatisch die entsprechenden Windows-Dienste auf dem System gestartet. Die Datenbank (und damit auch die Datenbank-Homepage) steht somit sofort zur Verfügung.

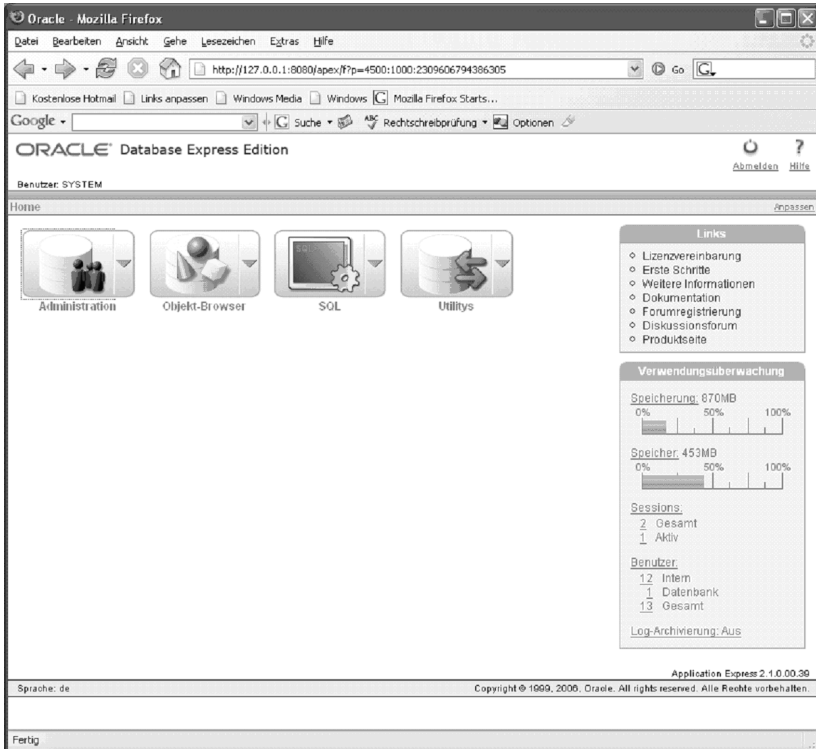


Abb. 2.5: Die Datenbank-Homepage Ihres Servers

Anhand der URL können Sie zwei Dinge erkennen:

Das Webtool läuft auf der Adresse 127.0.0.1, die auch als lokale Loopback-Adresse bezeichnet wird. Der Adresse folgend erkennt man den Port, im vorliegenden Fall die Standardadresse 8080.

Falls Sie das Bild aus Abbildung 2.5 nicht erhalten sollten und stattdessen eine Fehlermeldung der Art *Verbindung konnte nicht hergestellt werden*, so kann das verschiedene Ursachen haben. Sortiert nach der Problemwahrscheinlichkeit finden Sie in den folgenden Abschnitten einige Lösungsanregungen.

2.2.1 Browser-Einstellungen korrekt?

Oftmals liegt der Grund dafür in einem falsch konfigurierten Browser. Hier sollten Sie überprüfen, ob Ihr Browser einen Proxy-Server nutzt. Sollte dies der Fall sein, müssen Sie sicherstellen, dass für die lokale Adresse 127.0.0.1 der Proxy umgangen wird. Wie das genau zu tun ist, hängt vom jeweiligen Browser ab. Bitte haben Sie Verständnis dafür, dass ich an dieser Stelle nicht ein genaues Rezept für jeden einzelnen Browser abgeben kann, wie das zu bewerkstelligen ist. In der Regel müssen die Browser allerdings nach Änderungen an der Konfiguration neu gestartet werden. Die URL zur Datenbank-Homepage ist übrigens im Windows-Startmenü unter *Start | Programme | Oracle Daten 10g Express Edition | Gehe zu Datenbank-Homepage* gespeichert. Mit Aufruf dieses Menüpunktes wird dann der Standardbrowser gestartet.

2.2.2 Netzwerk und Ports korrekt?

Sollte diese Änderung noch nicht den gewünschten Erfolg bringen, überprüfen Sie als nächsten Schritt mittels eines einfachen

```
ping 127.0.0.1
```

ob die Loopback-Adresse Ihres Rechners erreichbar ist. In der Regel ist die lokale Serveradresse in der Datei `<SystemRoot>\system32\drivers\etc\networks` hinterlegt. Falls nicht, holen Sie das entsprechend nach.

Nächster Schritt wäre die Überprüfung des Ports 8080. Die Anweisung

```
netstat -a
```

zeigt Ihnen den Status aller Verbindungen des Servers an. In dieser Liste sollte eine Zeile der Art

```
TCP    obelix:8080          obelix:0              ABHÖREN
```

erscheinen, wobei `obelix` hier der Name meines Computers (in Anlehnung an »stark, allerdings ein wenig schlicht«) ist.

Zu guter Letzt könnten Sie mithilfe des Programms `lsnctr1` überprüfen, ob der Listener-Prozess korrekt arbeitet. Anfragen an die Datenbank erfolgen auch mittels des TCP-Protokolls, neben der IP-Adresse (die ja weiter oben hoffentlich erfolgreich überprüft worden ist) spielt hier die Portnummer 1521 (sofern Sie bislang noch nicht verändert worden ist) eine wichtige Rolle. Der Listener-Prozess ist der Programmteil, der auf dem definierten Port »horcht«, ob Anfragen an die Datenbank geschickt werden. Nach einer erfolgreichen Installation finden Sie in der

Computerverwaltung unter *Dienste* einen solchen mit dem Namen OracleXETNS-Listener, der gestartet sein sollte. Das Programm `lsnrctl.exe` finden Sie im Verzeichnis `<Installationsverzeichnis von XE>\app\oracle\product\10.2.0\server\BIN`. Es handelt sich hierbei um ein zeichenorientiertes Programm, das von der Kommandozeile aus aufgerufen wird. Über die Option `help` können Sie sich einen Überblick über die verschiedenen Aufrufoptionen verschaffen. Mit dieser Anwendung können Sie den eben angesprochenen Dienst starten oder stoppen oder eben auch Statusabfragen durchführen. Sollte beispielsweise der Dienst nicht korrekt gestartet sein, würde die Ausgabe von `lsnrctl status` etwa so aussehen:

```
LSNRCTL> status
Anmeldung bei (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC_FOR_XE)))
TNS-12541: TNS: Kein Listener
TNS-12560: TNS: Fehler bei Protokolladapter
TNS-00511: Kein Listener
32-bit Windows Error: 2: No such file or directory
Anmeldung bei (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=obe-
lix)(PORT=1521)))
TNS-12541: TNS: Kein Listener
TNS-12560: TNS: Fehler bei Protokolladapter
TNS-00511: Kein Listener
32-bit Windows Error: 61: Unknown error
LSNRCTL>
```

Bitte beachten Sie auch hier wieder, dass es sich bei `obelix` um meinen persönlichen Supercomputer handelt.

Einen Überblick über die gestarteten Dienste erhalten Sie in der Computerverwaltung, die über *Start | Einstellungen | Systemsteuerung | Verwaltung | Computerverwaltung* erreichbar ist. Die Abbildung 2.6 zeigt die fünf OracleXE-Dienste und deren »Startaufstellung«.

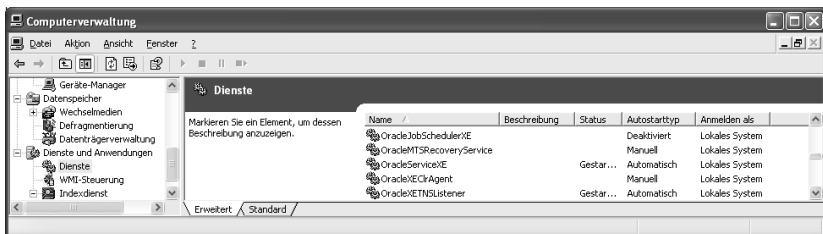


Abb. 2.6: OracleXE-Dienste

2.3 Ein Blick hinter die Kulissen

Nach diesen recht einfachen Schritten ist es jetzt an der Zeit, einen Blick unter die Motorhaube zu werfen. Als angehender Datenbankadministrator sollten Sie sich damit auskennen, welche Teile der Software wo zu finden sind und vor allem wie die verschiedenen Komponenten miteinander im Zusammenhang stehen.

2.3.1 Die Log-Dateien

Informationen über den Installationsvorgang selbst finden Sie im System-Root-Verzeichnis in der Datei `OracleDatabaseXEserverInstall.log`. Das System-Root-Verzeichnis auf einem Windows-System ist in der Regel `c:\windows`. Sie können es aber auch interaktiv ermitteln, indem Sie an einer Eingabeaufforderung die Anweisung

```
set systemroot
```

eingeben.

Die Datei selbst ist sehr groß, ca. 5 MB, was die Analyse derselbigen nicht unbedingt vereinfacht. Ich habe mir in solchen Fällen immer angewöhnt, nach bestimmten Schlüsselwörtern wie beispielsweise »Error« oder »Abort« zu suchen, wenn man mal davon ausgeht, dass etwaige Probleme in englischer Sprache in den Log-Dateien geschrieben werden.

Neben der eigentlichen Log-Datei des Windows-Installers schreibt die Datenbank selbst noch einige Protokolle, die sich ebenfalls noch auf die Installation beziehen. Diese sind zu finden im Verzeichnis `<Installationsverzeichnis von XE>\app\oracle\product\10.2.0\server\config\log`.

2.3.2 Die Gruppe ORA_DBA

Vielleicht ist Ihnen bei der Analyse der Log-Dateien aus dem vorhergehenden Abschnitt aufgefallen, dass die Installation von Oracle XE im Windows-System eine neue Benutzergruppe ORA_DBA (sofern diese noch nicht existiert) angelegt und den aktuellen Benutzer (unter dem die Installation von Oracle Express erfolgt ist) dieser Gruppe zugeordnet hat. Die neue Gruppe finden Sie ebenfalls in der *Computerverwaltung*, wie Abbildung 2.7 zeigt.

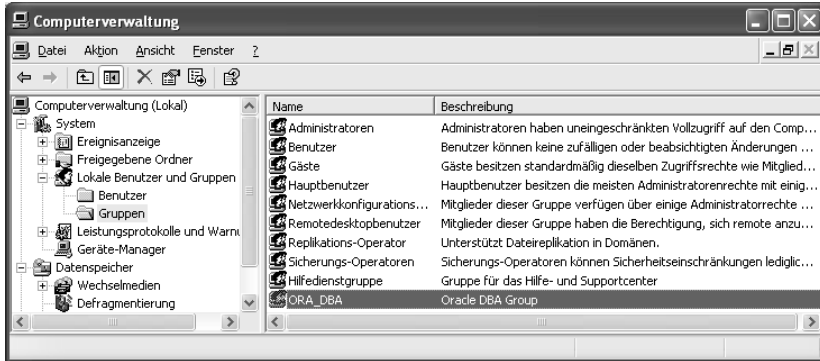


Abb. 2.7: Neue Oracle-DBA-Gruppe

Alle Mitglieder der Gruppe ORA_DBA erhalten alle Rechte des SYSDBA-Users. Dazu gehören unter anderem alle Administrationsrechte, die die Datenbank selbst betreffen. Als SYSDBA-User können Sie folgende Anweisungen durchführen:

- CREATE DATABASE
- ALTER DATABASE (BACKUP | OPEN/MOUNT)
- STARTUP
- SHUTDOWN
- ARCHIVE LOG
- RECOVER
- ...

Hinweis

Neben SYSDBA (als Synonym für *System Database Administrator*) gibt es im Übrigen noch die besondere Rolle SYSOPER (als Synonym für *System Operator*). In dieser Rolle sind alle oben aufgeführten Anweisungen bzw. Rechte enthalten bis auf das Recht, neue Datenbanken anzulegen (SYSDBA besitzt zwar noch einige weitere Rechte, beispielsweise das Ändern des Zeichensatzes; diese können wir allerdings an dieser Stelle getrost vernachlässigen).

Sie können sich beispielsweise an der Datenbank anmelden, indem Sie unter *Start | Programme | Oracle Database 10g Express Edition* den Menüpunkt *SQL Befehlszeile ausführen* aufrufen. Daraufhin wird SQL*Plus, ein Programm, dem wir uns in

Kapitel 3 noch intensiv widmen werden, gestartet. Geben Sie am Prompt die Anweisung

```
connect / as sysdba
```

ein. Hier die entsprechende Ausgabe:

```
SQL*Plus: Release 10.2.0.1.0 - Production on Di Apr 4 18:06:33 2006  
Copyright (c) 1982, 2005, Oracle. All rights reserved.
```

```
SQL> connect / as sysdba  
Connect durchgeführt.  
SQL>
```

Wie Sie sehen, müssen Sie kein weiteres Passwort angeben. Dadurch, dass der Benutzer, mit dem Sie am Windows-Betriebssystem angemeldet sind, Mitglied in der Gruppe ORA_DBA ist, erfolgt direkt eine Anmeldung an der Datenbank. Man könnte der Administration von Benutzern und insbesondere von externen Benutzern und deren Rollen und Rechten ein eigenes Kapitel widmen. An dieser Stelle darf ich in diesem Zusammenhang auf weiterführende Literatur, beispielsweise »Oracle-Datenbank, Administration und Management« von Michael Ault verweisen. Erlauben Sie mir dennoch noch einen Tipp in eigener Sache zu erwähnen. Mir ist es tatsächlich einmal passiert dass ich das Passwort vergessen habe, das ich bei der Installation der Datenbank festgelegt hatte. Hier kam mir die Mitgliedschaft in der ORA_DBA-Gruppe zugute. So konnte ich mich nämlich an der Datenbank anmelden als sysdba und über die folgende Zeile das Passwort ändern:

```
connect / as sysdba;  
alter user sys identified by NeuesPasswort;  
alter user system identified by NeuesPasswort;
```

2.3.3 Die Dienste

Ein weiterer Blick auf die Computerverwaltung, genauer gesagt auf die Dienste, fördert ein weiteres Ergebnis der Installation von Oracle XE zutage. Wie die Abbildung 2.8 zeigt, sind auf dem Windows-System fünf Dienste eingerichtet. Glauben Sie mir, diese Dienste entstammen der Installation und waren nicht schon vorher auf dem System.

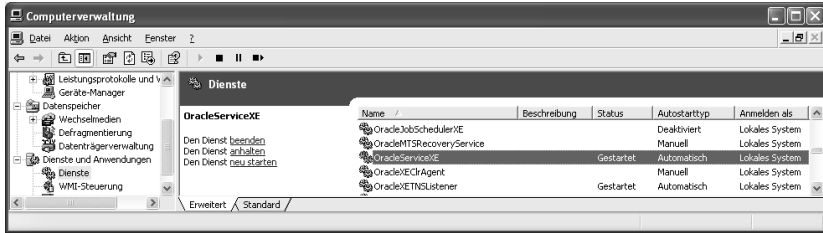


Abb. 2.8: Die Oracle-XE-Dienste

OracleJobSchedulerXE

Mithilfe dieses Jobs können Sie zeitgesteuert Routinen oder Programme – auch solche außerhalb der Datenbank – ausführen. Nach der Standardinstallation ist dieser Dienst zunächst deaktiviert und müsste aktiviert und anschließend gestartet werden. Das allein reicht allerdings noch immer nicht aus. Sie als Datenbankadministrator müssen innerhalb der Datenbank entsprechende Benutzer und Rollen anlegen, denen es erlaubt ist, Programme außerhalb der Datenbank zu starten. Wir werden im Folgenden darauf noch näher eingehen. Hinter diesem Dienst verbirgt sich übrigens das folgende Programm:

```
<Installationverzeichnis>\app\oracle\product\10.2.0\server\Bin\extjob.exe XE
```

OracleMTSRecoveryService

Hierunter verbirgt sich der Oracle-Service für den *Microsoft Transaction Server*. Was ist der Microsoft Transaction Server, werden Sie sich jetzt vermutlich fragen. Auch wenn Sie ein Buch über Oracle in den Händen halten, lassen Sie mich drei Worte zu dem Transaction Server verlieren. Der Microsoft Transaction Server ist ein Applikations- und Transaktionsserver mit dessen Hilfe man verteilte Anwendungen betreiben kann. Unter einer *verteilten Anwendung* versteht man Programme, die aus mehreren Teilen oder Komponenten bestehen. Man spricht in diesem Zusammenhang auch von *Schichten*. In der Regel kommen dabei drei Schichten zum Einsatz:

- die Präsentationsschicht, d.h. das GUI des Anwenders
- die Geschäftsprozessschicht, in der die Business-Logik abgelegt ist
- die Datenbankschicht, in der die Daten gespeichert werden

Ein verteiltes Programm besteht also aus mehr als nur aus einer ausführbaren Datei, sondern aus mehreren Komponenten. Der Microsoft Transaction Server sorgt nun bei verteilten Anwendungen dafür, dass die Transaktionssicherheit über alle Komponenten hinweg gewährleistet wird.

Soll also Oracle Express als Datenbankschicht für verteilte Anwendungen zum Einsatz kommen, wird dieser Dienst benötigt. Hinter diesem Dienst verbirgt sich die folgende Anwendung:

```
<Installationsverzeichnis>\app\oracle\product\10.2.0\server\BIN\omts-reco.exe "OracleMTSRecoveryService"
```

Ich werde in diesem Buch nicht weiter auf dieses Thema eingehen, darf an dieser Stelle aber vielleicht auf die Oracle-Dokumentation zum Thema »Oracle Services for Microsoft Transaction Server Developer's Guide« verweisen. Eine Suche auf der Oracle-Website sollte ein umfangreiches Dokument zutage fördern.

OracleServiceXE

Kommen wir nun zum Herzstück von Oracle Express. Hinter diesem Dienst verbirgt sich das folgende Programm:

```
<Installationsverzeichnis>\app\oracle\product\10.2.0\server\bin\ORACLE.EXE XE
```

Es vereint verschiedenste Aktivitäten in einem Multithread-Prozess in sich. Auf Unix- bzw. Linux-Systemen würden Sie für jede einzelne Aktivität bzw. jeden Thread (dieses Services) einen Hintergrundprozess in der Prozessliste sehen können. Zu diesen Aktivitäten des Windows-Service bzw. Hintergrundprozessen gehören unter anderem:

Der Database Writer (dbw) Dieser (Teil-)Prozess ist dafür zuständig, Daten aus dem Hauptspeicher, genauer gesagt aus der SGA (vgl. Abschnitt 2.1.3), auf die Platten zu schreiben. Der Prozess erkennt, wenn bestimmte Puffer-Bereiche als »dirty« bzw. geändert markiert sind, und sorgt dann für die entsprechende Persistenz, in dem die geänderten Datensätze in die Tabellen eingefügt werden. Es ist der einzige Prozess, der Daten in die Tabellen schreibt.

Der Process Monitor (pmon) Der Process Monitor überwacht die Sessions der Benutzer, die auf die Datenbank zugreifen. Er erkennt beispielsweise, wenn ein Anwender seine Sitzung (gewollt oder ungewollt) abrupt beendet hat. Denkbar wäre dabei ein Absturz der Client-Anwendung, während der Anwender gerade z.B. eine Bestellung erfasst. Denkbar wäre auch, dass der Anwender über **Strg** + **Alt** + **Entf** die Anwendung beendet. In solchen Fällen könnte es sein, dass in der SGA bestimmte Puffer schon als geändert markiert sind und vom Database Writer auf die Platte geschrieben werden müssten, was unter Umständen zu inkonsistenten Daten führen würde. Der Process Monitor erkennt solche abgebrochenen Sitzungen und führt die entsprechenden Aufräumarbeiten durch. Vergleichen Sie diesen Prozess mit dem Garbage Collector von Java.

Der System Monitor (smon) Die Aufgabe des System Monitors ähnelt der des Process Monitors, allerdings konzentriert sich dieser Prozess nicht auf Benutzeraktivitäten sondern auf systeminterne Aktivitäten.

Der Log Writer (lgwr) Hier geht es um das Thema Transaktionssicherheit. Stellen Sie sich folgenden Geschäftsprozess vor: Sie möchten 1.000 Euro vom Girokonto auf das Sparbuch überweisen. EDV-technisch verbergen sich dahinter zwei Aktionen: den Bestand des Girokontos um 1.000 Euro reduzieren, den Bestand des Sparbuches um 1.000 Euro erhöhen. Beide Teilprozesse müssen als Ganzes, in einer Transaktion, durchgeführt werden können. Sollte es in einem Teilprozess zu einem Problem kommen, muss die gesamte Transaktion rückgängig, man spricht auch von »zurückgerollt«, werden können. Dazu muss das System allerdings noch wissen, welche Teilschritte zu dieser Transaktion gehörten. Genau dies ist die Aufgabe des Log Writers: Dieser Prozess »merkt« sich sozusagen, was zu einer bestimmten Transaktion zu tun ist. Diese Informationen werden auch persistent in den sogenannten Redo-Log-Puffer gesichert.

Der Checkpoint (ckpt) Der Checkpoint-Prozess schreibt in regelmäßigen Abständen bestimmte Informationen in Control-Dateien. Diese bestimmten Informationen definieren die Stellen in dem Redo-Log-Puffer, bei denen z.B. ein Recovery beginnen muss. Dies dient der Sicherstellung der Datenkonsistenz, falls der Datenbankserver unkontrolliert seine Arbeit beendet. Nach einem Neustart kann er dann in den entsprechenden Log-Dateien nachschauen, wann der letzte Checkpoint erreicht war (und damit ein konsistenter Datenstand innerhalb der Datenbank vorlag) und was sich seitdem geändert hat. Letzteres findet man in den Redo-Log-Puffern, die vom Log-Writer-Prozess geschrieben worden sind.

Checkpoints werden auch herangezogen, wenn die Datenbank bzw. Instanz kontrolliert heruntergefahren wird. Hier erkennt dann der Database-Writer-Prozess, wann der letzte Checkpoint war und was sich seitdem geändert hat. Diese letzten Änderungen werden dann in den Datenbanktabellen verewigt.

Der Recoverer (reco) Der Recover-Prozess ist dafür zuständig, gescheiterte, d.h. abgebrochene Transaktionen wieder in einen konsistenten Zustand zu überführen.

Der Archiver (arc) Der Archiver-Prozess kopiert und speichert die Redo-Log-Puffer-Dateien an einem vom DBA definierten Ort. Neben den eigentlichen Datenbankdateien kommt den Redo-Log-Puffern eine besondere Bedeutung zu, denn hierin sind die aktuell laufenden Änderungen festgehalten. Diese nochmals an einen anderen Ort zu kopieren, macht nicht nur aus Sicherheitsgesichtspunkten Sinn. Es ist gängige Praxis, über diesen Weg eine Sicherheitskopie oder auch Offline-Datenbank zu erstellen. Man kann z.B. einmal in der Nacht die eigentlichen Datenbankdateien auf einen anderen Server und tagsüber die Redo-Log-Puffer-Dateien kopieren. Über diesen Weg steht Ihnen komfortabel und zeitnah immer eine aktuelle Version Ihrer Produktiv-Datenbank zur Verfügung.

OracleXEClrAgent

Zu diesem Service gibt es kein Pendant auf Linux-Seite. Es ist unter Windows möglich, Anwendungen mit dem .NET Framework von Microsoft zu entwickeln. Diese Anwendungen werden in der Regel nicht in eine ausführbare Datei, sondern in eine sogenannte Assembly-Datei überführt, die dann von einem Interpreter zur Laufzeit analysiert und ausgeführt wird. Dieser Interpreter ist vergleichbar mit einer Java Virtual Machine, unter Windows heißt das Kind eben nur *Microsoft Common Language Runtime* oder kurz *CLR*. Diese Laufzeitumgebung wird mit dem oben genannten Dienst zur Verfügung gestellt. Ist er gestartet, können Sie innerhalb der Datenbank .NET Stored Procedures ausführen.

Voraussetzung für die Verwendung dieser Technologie ist neben dem Microsoft Visual Studio (2003 ff.) auch die Installation der Oracle Developer Tools für Visual Studio, deren Installation hier nicht beschrieben ist. Dieser Dienst verbirgt sich hinter der Datei

```
<Installationsverzeichnis>\app\oracle\product\10.2.0\server\bin\OraCl-  
rAgt.exe agent_sid=CLRExtProc max_dispatchers=2 tcp_dispatchers=0  
max_task_threads=6 max_sessions=25
```

Er ist standardmäßig nach der Installation nicht gestartet, weil er eben nicht zwingend für den Betrieb der Datenbank notwendig ist.

OracleXETNSListener

Wenn der OracleServiceXE-Dienst das Herzstück der Datenbank ist, ist der OracleXETNSListener die Hauptschlagader. Beide Komponenten müssen gestartet sein und fehlerfrei arbeiten, ansonsten wird der Anwender oder Sie als DBA und Entwickler nicht viel Spaß bei der Arbeit haben.

Der Listener-Prozess ist derjenige, der auf dem bei der Installation definierten Port »horcht«, ob von Client-Anwendungen Anfragen an die Datenbank gerichtet werden, und diese ggf. an die entsprechenden Prozesse bzw. den Oracle-Kernel weiterleitet.

2.3.4 Die Verzeichnisstruktur

Nachdem wir nun die wichtigsten Komponenten der Oracle-XE-Installation in Form der DBA-Benutzergruppe ORA_DBA und die verschiedenen Dienste kennen gelernt haben, ist es an der Zeit, einen Blick auf die Verzeichnisstruktur zu werfen. Im Gegensatz zu den umfangreicheren und (auch teureren) Oracle-Datenbank-Produkten war die Installation recht einfach, es wurden nur wenig Parameter abgefragt. Als Resultat davon sieht die Verzeichnisstruktur relativ ähnlich auf allen Systemen aus. Tabelle 2.1 listet die wichtigsten Ordner auf.

Verzeichnis	Bedeutung
c:\oraclexe	Root-Verzeichnis der Installation
c:\oraclexe\app	Anwendungsverzeichnis von XE
c:\oraclexe\app\oracle\admin\xe	Dump-Verzeichnisse der Installation
c:\oraclexe\app\oracle\doc	HTML-Dokumentation zum <i>Getting Started Guide</i>
..\app\oracle\flash_recovery_area	Zielverzeichnis für alle Dateien, die etwas mit dem Backup und Recovery der Datenbank zu tun haben
c:\oraclexe\app\oracle\product	Programmverzeichnis des XE-Servers
...\product\bin	Verzeichnis der ausführbaren Dateien, Programmbibliotheken und Batch-Dateien von XE. Dieses Verzeichnis wurde bei der Installation auch mit in den Suchpfad des Systems eingetragen.
...\product\config	Enthält in Unterordnern einige Konfigurations- und Log-Dateien, die nach der Installation ausgeführt werden.
...\product\ctx	Installations- und Konfigurationsdateien für die Volltextsuche in Oracle. Auch bekannt als <i>Oracle Text</i> .
...\product\database	Log- und Error-Verzeichnis, in das die DB beim Start oder auch während des Betriebs schreibt, wie sie sich fühlt
...\product\dfs	In dem Verzeichnis liegt die Datei <code>spfilexe.ora</code> . Es handelt sich dabei um eine Parameterdatei, in der systemweite Initialisierungsparameter abgelegt werden und die beim Start gelesen wird. Diese wurden bis zur Version 8 von Oracle in einer <code>init.ora</code> abgelegt. Die neue Datei kann jedoch direkt aus der Datenbank heraus (<code>alter system ...</code>) beschrieben werden
...\product\demo	In einem Unterordner befinden sich SQL-Skripte zur Erzeugung der Beispieldatenbank HR.
...\product\doc	Lizenzvereinbarung von Oracle zu XE
...\product\jdbc	Java-Libraries, die eine Kommunikation mit der Datenbank erlauben
...\product\jlib	Java-Bibliothek zur Zeichensatzformatierung
...\product\ldap	Meldungstexte, die angezeigt werden (können), wenn XE mit <i>LDAP (Lightweight Directory Access Protocol)</i> zusammenarbeitet

Tabelle 2.1: Die Verzeichnisstruktur einer Oracle-XE-Installation

Verzeichnis	Bedeutung
... \product\lib	Bibliotheksdateien, die beispielsweise in C/C++-Anwendungen mit eingebunden werden müssen, um mit der DB kommunizieren zu können
... \product\log	Log-Verzeichnis, in dem Client-spezifische Meldungen protokolliert werden
... \product\network	Wichtiges Verzeichnis zur Konfiguration der Datenbank
... \product\network\admin	Verzeichnis, in dem die wichtigsten Konfigurationsdateien jeder Oracle-Datenbank (<code>listener.ora</code> , <code>sqlnet.ora</code> und <code>tnsnames.ora</code>) liegen. Im Unterordner <code>\sample</code> sind Beispieldateien dieser Dateien zu finden.
... \product\network\log	Log-Verzeichnis für den Listener-Prozess
... \product\network\mesg	Verzeichnis, in dem lokalisierte Fehlermeldungen liegen, die bei Bedarf vom Listener-Prozess in die Log-Dateien geschrieben werden
... \product\network\trace	Bei Bedarf kann der Listener-Prozess mitgetraced werden. Etwaige Trace-Meldungen werden in dieses Verzeichnis geschrieben.
... \product\nls	<i>NLS</i> steht für <i>National Language Support</i> .
... \product\OCI	<i>OCI</i> steht für <i>Oracle Call Interface</i> . Es handelt sich hierbei um eine Reihe von Bibliotheken/Routinen, die in eigene C/C++-Anwendungen eingebunden werden können. Über das OCI-Interface können diese Anwendungen dann direkt mit der Datenbank kommunizieren.
... \product\odbc	Hilfe-Datei, Release Notes und lokalisierte Meldungen für den Oracle ODBC-Treiber. Der Treiber selbst liegt in Form der Datei <code>SQORA32.DLL</code> im <code>..\server\bin</code> -Verzeichnis.
... \product\ODE	Release Notes für die <i>Oracle Database Extensions for .NET</i>
... \product\ODP.NET	Dokumentation und Bibliotheken für die <i>Oracle Data Provider for .NET</i>
... \product\oledb	Include- und Library-Datei für die Oracle OLE DB-Komponenten
... \product\oracore	Lokalisierte (Fehler-) Meldungen von Oracle
... \product\oramts	Skript zur Installation des Pakets <code>utl_oramts</code> . Steht im Zusammenhang mit dem Microsoft Transaction Server.

Tabelle 2.1: Die Verzeichnisstruktur einer Oracle-XE-Installation (Forts.)

Verzeichnis	Bedeutung
... \product\sqlplus	Konfigurationsverzeichnis für SQL*Plus. Hier liegt beispielsweise die Datei <code>glogin.sql</code> , die bei jedem Start von SQL*Plus ausgeführt wird.
... \product\xdk	Enthält Dateien für lokalisierte Fehlermeldungen.
... \oracle\oradata	Verzeichnis, in dem die Datendateien von Oracle XE liegen. Im Unterverzeichnis XE findet man die Dateien CONTROL.DBF SYSAUX.DBF SYSTEM.DBF TEMP.DBF UNDO.DBF USERS.DBF

Tabelle 2.1: Die Verzeichnisstruktur einer Oracle-XE-Installation (Forts.)

2.3.5 Die Registry-Einträge

Nachdem wir nun schon zwei Blicke unter die Motorhaube geworfen haben, darf ein dritter und letzter nicht fehlen. Gerade bei der Installation von Software unter Windows darf man nicht die Registry vergessen. Neben einigen anderen Stellen finden Sie unter `HKEY_LOCAL_MACHINE | SOFTWARE | ORACLE | KEY_XE` einige interessante Einträge (vgl. Abb. 2.9).

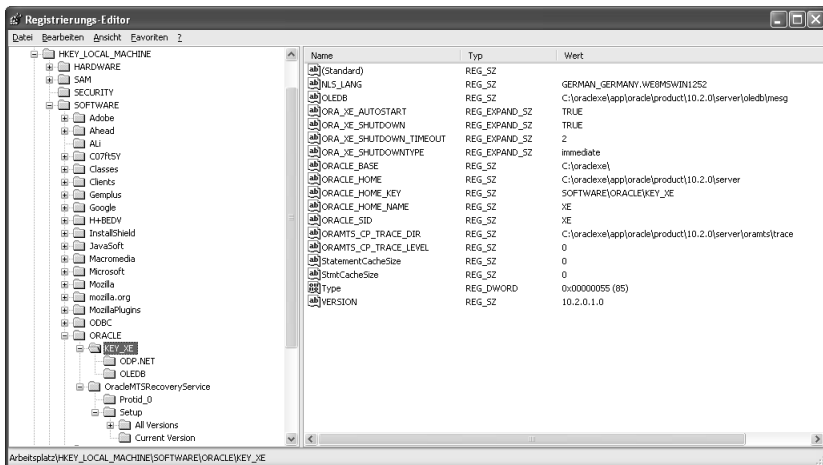


Abb. 2.9: Registry-Einträge von Oracle XE

Die folgende Tabelle gibt Auskunft über die wichtigsten Einträge:

Schlüssel	Bedeutung
NLS_LANG	Der Parameter ist nach folgender Nomenklatur aufgebaut: SPRACHE_REGION.ZEICHENSATZ SPRACHE definiert dabei die Sprache, in der jegliche Arten von Meldungen oder auch Tages- oder Monatsnamen angezeigt werden. REGION definiert, wie beispielsweise Datums- oder Währungsfelder angezeigt werden. ZEICHENSATZ definiert den Character Set. Darin ist z.B. festgelegt, welches Zeichen mit welchem Byte-Wert kodiert wird.
ORA_XE_AUTOSTART	Definiert, ob die Datenbank beim Start von Windows automatisch gestartet wird. Den automatischen Start bzw. Nichtstart könnten Sie auch über die Startart der beiden Dienste OracleServiceXE und OracleXETNSListener bestimmen.
ORA_XE_SHUTDOWN	Falls dieser auf TRUE steht, wird die Datenbank heruntergefahren, nachdem die im Parameter ORA_XE_SHUTDOWN_TIMEOUT definierte Zeit in Sekunden gewartet worden ist.
ORA_XE_SHUTDOWNTYPE	Definiert, wie die Datenbank heruntergefahren werden soll. Gültige Werte sind <code>immediate</code> , <code>normal</code> und <code>abort</code> und <code>transactional</code> . <code>immediate</code> bedeutet, dass die Datenbank sofort geschlossen wird, die Benutzerprozesse werden dabei kontrolliert beendet. Der Typ <code>normal</code> wartet so lange, bis alle Transaktionen der Benutzer beendet sind, neue Verbindungen werden allerdings nicht weiter zugelassen. <code>transactional</code> ist dem Typ <code>immediate</code> ähnlich, allerdings wird auch hier gewartet, bis alle Transaktionen geschlossen sind. <code>abort</code> fährt die Datenbank unverzüglich und ohne Rücksicht auf Verluste herunter.
ORACLE_BASE	Root-Verzeichnis der Installation von XE
ORACLE_HOME	Definiert die Umgebungsvariable ORACLE_HOME. Der Wert entspricht dem Server-Verzeichnis der Installation.
ORACLE_SID	Definiert den <i>System Identifier</i> . Je Datenbankinstanz wird ein eindeutiger System Identifier definiert. Die SID ist unter Windows auch jeweils mit in den Dienstnamen zu finden. Das bedeutet automatisch, dass Sie mehrere gleiche Dienste auf einem System finden, sofern dort mehr als eine Datenbankinstanz installiert ist.

2.4 Die Programme von Oracle XE

Oracle XE kommt mit einer Vielzahl von Programmen daher, wovon viele allerdings nur einen geringen Nutzen für den DBA oder Entwickler haben. Einige sind jedoch immens wichtig, sie werden bei der täglichen Arbeit benötigt. Diese möchte ich in den nächsten Abschnitten analysieren und Ihnen näherbringen.

2.5 Das Startmenü

Werfen wir zunächst einen Blick auf das Startmenü von Windows. Dort finden wir nach erfolgter Installation unter *Programme* den Eintrag *Oracle 10g Database Express Edition*. Hierunter verbergen sich einige weitere Menüpunkte:

- Hilfe aufrufen
- Backup der Datenbank
- Datenbank starten
- Datenbank stoppen
- Datenbank zurückschreiben
- Erste Schritte
- Gehe zu Datenbank-Homepage
- SQL-Befehlszeile ausführen

auf die ich im Folgenden noch näher eingehen werden. Erwähnenswert in diesem Zusammenhang ist die Tatsache, dass die Installation von Oracle Express die Menüeinträge unter *All Users* einträgt. Das bewirkt, dass alle anderen Benutzer des Windows-Systems die Menüeinträge zwar sehen, allerdings nichts damit anstellen können. Hier fehlt den Anwendern sowohl die Freigabe auf das Installationsverzeichnis als auch die Mitgliedschaft in der ORA_DBA-Gruppe. Die Reihenfolge der hier beschriebenen Programme entspricht der Reihenfolge im Startmenü. Ich würde es hier für sinnvoller halten (und habe es auf meinem System auch so konfiguriert), die Menüpunkte logisch zusammenhängend zu sortieren.

2.5.1 Hilfe aufrufen

Unter diesem Menüpunkt verbergen sich einige Menüeinträge, hinter denen sich letztlich jeweils eine URL verbirgt. Herauszufinden ist dies, indem Sie sich die Eigenschaften des Menüpunktes anschauen. Die entsprechenden Webadressen liegen in Form von URL-Dateien im Server-Verzeichnis Ihrer XE-Installation (vgl. Abb. 2.10). Teilweise verbergen sich hinter diesen Links Adressen auf der Website von Oracle, sodass Sie online sein müssen, um auf die entsprechenden Informationen zugreifen zu können. Teilweise enthalten die URLs auch lokale Adressen, so z.B. beim Menüpunkt *Online-Hilfe lesen*. Dessen URL sieht wie folgt aus:

```
[InternetShortcut]
URL=http://127.0.0.1:8080/apex/
www_flow_help.show_help?p_flow_id=4500&p_step_id=1000
```

Wie Sie sehen, verweist dieser Link auf die lokale Adresse der Datenbank-Homepage. Stellen Sie vor Zugriff hierauf sicher, dass die entsprechenden Dienste gestartet und die Datenbank damit hochgefahren ist. Nur dann erhalten Sie Zugriff auf die entsprechenden Dokumente.

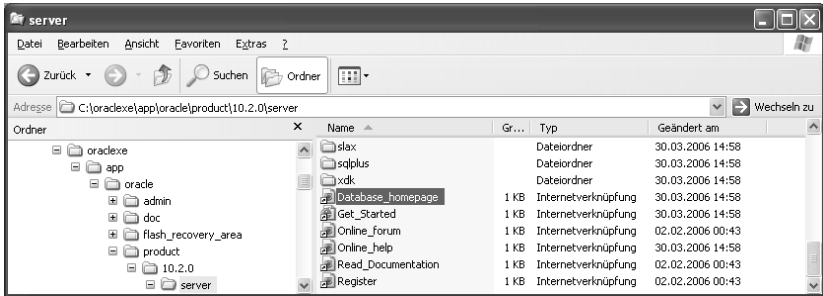


Abb. 2.10: Die URLs des Hilfe-Menüs

2.5.2 Backup der Datenbank

Auch wenn das Backup einer Datenbank sicherlich zu den fortgeschrittenen Aufgaben eines DBAs gehört und wir uns diesem Thema erst in einem späteren Kapitel widmen werden, möchte ich hier dennoch auf die wesentlichen Aspekte dieses Menüpunktes eingehen.

Der Vollständigkeit halber seien hier aber auch weitere Möglichkeiten der Datensicherung erwähnt:

■ Kopie der Datendateien

Die simpelste besteht darin, einfach das Verzeichnis /oradata aus dem Installationsverzeichnis an einen sicheren Ort zu kopieren, sodass Sie diese Dateien ggf. in das entsprechende Verzeichnis (auf dem gleichen System) zurückkopieren können.

■ Kopie des Installationsverzeichnisses

Eine etwas komplexere Methode wäre, das komplette Installationsverzeichnis zu sichern. Dieses könnten Sie theoretisch auf einen anderen Rechner kopieren (auf dem Sie zuvor schon einmal eine leere XE-Datenbank installiert haben). Dadurch wäre es möglich, mit dem gleichen Datenstand (und gleichen Einstellungen) die Datenbank auf einem Rechner zu Hause und auf einem Notebook unterwegs zu betreiben.

■ Nutzung von exp und imp

Bei `exp` und `imp` handelt es sich um zwei ausführbare Dateien im `bin`-Ordner der Installation. Jetzt raten Sie doch mal, welches Programm exportiert und welches importiert. Richtig, `exp` ist für den Export zuständig. Nach dem Start müssen Sie sich – sofern nicht über Kommandozeilenparameter angegeben – anmelden und einige Parameter eingeben. Daraufhin startet der Export in eine Datei; voreingestellt ist hier `EXPDAT.DMP`. Diese erscheint, sofern nichts weiter spezifiziert ist, im gleichen Programmordner wo auch das Export-Programm zu finden ist. Ebenso einfach ist das Import-Programm (`imp` ☺) zu bedienen. Nach einigen wenigen Parametern startet der entsprechende Import.

Hinweis

Die beschriebenen Methoden zur Sicherung der Datenbank über das Dateisystem funktionieren nur, wenn die Datenbank (bzw. Instanz) heruntergefahren ist.

Doch nun zurück zum Menüpunkt *Backup*. Gestartet wird über diesen Punkt die Batch-Datei `Backup.bat`, zu finden im Server-Verzeichnis von Oracle XE. Die Datei selbst ist ein wenig zu lang, um Sie hier komplett abdrucken und analysieren zu können. Vereinfacht gesagt, passiert durch dieses Skript das Folgende: Es generiert und speichert einige SQL-Skripte (die im Temp-Ordner abgelegt werden) und führt diese mithilfe von SQL*Plus aus. Die gesamte Datenbank wird daraufhin in Unterordnern des Verzeichnisses `<Installationsverzeichnis>\app\oracle\flash_back_area` abgelegt. Per Default legt dieses Script ein Zwei-Generationen-Backup in der Flash-Back-Area an, d.h., dort liegen immer die letzte und die vorletzte Sicherung. Wenn ein drittes Backup erzeugt wird, wird das älteste gelöscht.

Log-Files der angelegten Backups finden Sie in `<Installationsverzeichnis>\product\10.2.0\server\database`. Die beiden Dateien (zu dem letzten und vorletzten Backup) lauten `OXE_BACKUP_CURRENT.TXT` und `OXE_BACKUP_PREVIOUS.TXT`.

Wenn Sie diesen Menüpunkt aufrufen, öffnet sich ein Eingabefenster, in dem die Datei `Backup.bat` ausgeführt wird. Dort werden Sie nach einer kurzen Gedenkminute mit der folgenden Meldung konfrontiert:

```
Warning: Log archiving (ARCHIVELOG mode) is currently disabled. If
you restore the database from this backup, any transactions that take
place between this backup and the next backup will be lost. It is
recommended that you enable ARCHIVELOG mode before proceeding so
that all transactions can be recovered upon restore. See the section
'Enabling ARCHIVELOG Mode...' in the online help for instructions.
Backup with log archiving disabled will shut down and restart the
database. Are you sure [Y/N]?
```

Der ArchiveLog-Mode ist per Voreinstellung nach der Installation ausgeschaltet. Moment mal, was ist eigentlich der ArchiveLog-Mode? Nun, die oben beschriebene Backup-Methode legt eine komplette Kopie der Datenbank an. Nehmen wir einmal an, dies geschieht jede Nacht um drei Uhr morgens. Um acht beginnen die Anwender wieder mit der Arbeit und um zwölf Uhr mittags stürzt das System ab und die letzte Sicherung muss zurückgeholt werden. Damit ist die Arbeit der Anwender von acht bis zwölf Uhr verloren. Würde man die Datenbank im ArchiveLog-Mode fahren, könnte man die Datensicherung von letzter Nacht nehmen und die Log-Dateien (in der die Transaktionen der Benutzer von acht bis zwölf gespeichert sind) wieder nachfahren. Sobald Sie also mindestens eine Sicherung haben, bietet sich der ArchiveLog-Mode zur Sicherung an.

2.5.3 Datenbank starten

Sofern die Datenbank nicht mit dem Start von Windows automatisch gestartet wird, können Sie mit diesem Menüpunkt die Datenbank starten. Hinter dem Menüpunkt verbirgt sich die DOS-Batch-Datei `StartDB.bat` aus dem `bin`-Verzeichnis der Server-Installation. Die Datei enthält insgesamt drei Zeilen:

```
net start OracleXETNSListener
net start OracleServiceXE
@oradim -startup -sid XE -starttype inst > nul 2>&1
```

Die beiden ersten Zeilen bzw. Anweisungen starten die beiden wichtigsten Dienste von Oracle Express. Mit der letzten Zeile wird das Programm `oradim` mit einigen Parametern aufgerufen. Zur Erläuterung dieser Anwendung muss ich ein wenig weiter ausholen. Bislang war immer die Rede vom Starten bzw. Hochfahren einer Datenbank. Diese Formulierung war nicht ganz korrekt. Nicht die Datenbank selbst, sondern eine Instanz davon (zu der Instanz gehören beispielsweise noch Steuerdateien o.Ä.) wird hoch- bzw. heruntergefahren. Die Datenbank (einer Instanz) wird geöffnet oder geschlossen.

Genau diese Aufgaben leistet das Program `oradim` – teilweise findet man in der Literatur auch noch den Namen *Instance Manager* – unter Windows. Hier die vollständige Syntax:

```
C:\tmp>oradim
ORADIM: <command> [options]. Siehe Handbuch.
Geben Sie einen der folgenden Befehle ein:
Erstellen Sie eine Instance, indem Sie die folgenden Optionen angeben:
    -NEW -SID sid | -SRVC srvc | -ASMSID sid | -ASMSRVC srvc [-SYSPWD pass]
    [-STARTMODE auto|manual] [-SRVCSTART system|demand] [-PFILE file | -
    SPFILE]
```

```

[-SHUTMODE normal|immediate|abort] [-TIMEOUT secs] [-RUNAS osusr/ospass]
Bearbeiten Sie eine Instance, indem Sie die folgenden Optionen angeben:
    -EDIT -SID sid | -ASMSID sid [-SYSPWD pass]
[-STARTMODE auto|manual] [-SRVCSTART system|demand] [-PFILE file | -
SPFILE]
[-SHUTMODE normal|immediate|abort] [-SHUTTYPE srv|inst] [-RUNAS osusr/
ospass]
Löschen Sie Instances, indem Sie die folgenden Optionen angeben:
    -DELETE -SID sid | -ASMSID sid | -SRVC srv | -ASMSRVC srv
Fahren Sie Services und Instance hoch, indem Sie die folgenden Optionen
angeben:
    -STARTUP -SID sid | -ASMSID sid [-SYSPWD pass]
[-STARTTYPE srv|inst|srv,inst] [-PFILE filename | -SPFILE]
Fahren Sie Service und Instance herunter, indem Sie die folgenden Optionen
angeben:
    -SHUTDOWN -SID sid | -ASMSID sid [-SYSPWD pass]
[-SHUTTYPE srv|inst|srv,inst] [-SHUTMODE normal|immediate|abort]
Fordern Sie Hilfe an, indem Sie die folgenden Parameter angeben: -? | -h |
-help

```

2.5.4 Datenbank stoppen

Analog zur StartDB.bat gibt es eine StopDB.bat. Der Inhalt dieser Datei beschränkt sich auf eine Zeile:

```
net stop OracleServiceXE
```

Hierüber wird zwar der Kern von Oracle Express gestoppt, nach Ausführung dieser Datei bleibt allerdings immer noch der Dienst OracleXETNSListener gestartet. Angehende DBAs werden vielleicht zunächst auf die Menüpunkte zurückgreifen, später vermutlich jedoch eher eigene Skripte zur Administration der Datenbank schreiben. Ich halte es für relativ sinnfrei, den OracleXETNSListener auf einem System zu starten bzw. gestartet zu lassen, während der OracleServiceXE heruntergefahren ist. Mein Tipp deshalb: Fügen Sie die folgende Zeile der Datei StopDB.bat hinzu:

```
net stop OracleXETNSListener
```

2.5.5 Datenbank zurückschreiben

Was nützt die beste Sicherung, wenn man nicht in der Lage ist, diese Sicherung zum Leben zurück zu erwecken? Sofern wenigstens eine Sicherung mittels der Batch-Datei Backup.bat erstellt worden ist, kann diese auf einfache Weise über die

Datei `Restore.bat` zurückgesichert werden. Sollten zwei Sicherungen in der Flashback-Area vorhanden sein, wird die aktuellste genommen.

2.5.6 Erste Schritte

Dieser Punkt ist schnell erklärt. Hinter ihm verbirgt sich – ähnlich der Hilfe-Menüpunkte weiter oben – eine URL. Hier zeigt sie auf die Datei `c:/oracle/app/oracle/doc/getting_started.htm`.

2.5.7 Gehe zu Datenbank-Homepage

Dieser Menüpunkt ist ähnlich schnell wie der vorherige erklärt. Der Aufruf dieses Punktes startet einen Webbrowser mit der Adresse `http://127.0.0.1:8080/apex/f?p=4550:11:1858938069472623::N0::`. Hierunter verbirgt sich die Startseite oder auch Datenbank-Homepage genannt. Erreichbar ist diese Seite allerdings nur, wenn die Datenbank-Instanz gestartet ist.

2.5.8 SQL-Befehlszeile ausführen

Dieser Menüpunkt startet SQL*Plus. Dieses Programm stellt sozusagen die SQL-Schnittstelle zur Oracle-Datenbank dar. Einmal gestartet und angemeldet, kann man dort Abfragen formulieren und an die Datenbank schicken oder auch DML-Anweisungen, also solche, die Daten verändern, durchführen. Auf den ersten Blick mag die Benutzerschnittstelle etwas antiquiert erscheinen, SQL*Plus bietet jedoch einen mächtigen Funktionsumfang (und ist zudem auf jedem Betriebssystem verfügbar, auf dem Oracle-Datenbanken laufen), den ich Ihnen in einem späteren Kapitel näherbringen möchte.

Wenn Sie sich die Eigenschaften des Menüpunktes anschauen, werden Sie feststellen, dass SQL*Plus mit dem Parameter `/nolog` gestartet wird. `/nolog` sorgt dafür, dass Sie sich nicht explizit an der Oracle-Datenbank anmelden müssen, sondern die Zugangsdaten aus der Windows-Anmeldung (sofern Sie Mitglied der Gruppe `ORA_DBA` sind) übernommen werden.

2.6 Der XE-Client

Nachdem Sie nun erfolgreich die Datenbank auf Ihrem Rechner zum Laufen gebracht haben, wird irgendwann die Zeit kommen, sich mit der Frage zu beschäftigen, wie man von anderen Rechnern Zugriff auf die Oracle-Express-Datenbank erlangt. Der Trend geht ja ohnehin zum Zweitrechner, diesen könnten Sie jetzt einer wirklich sinnvollen Nutzung zuführen. Notwendige Voraussetzung dafür ist natürlich, dass die Rechner in einem Netzwerk miteinander verbunden sind. Ist

das der Fall, benötigen Sie nur noch die XE-Client-Software. Diese ist ebenfalls frei verfügbar und kann über die Oracle-Website (<http://www.oracle.com/technology/software/products/database/xe/index.html>) heruntergeladen werden oder noch einfacher: Sie wechseln in der Ordner Oracle-Software der Buch-CD. Dort finden Sie die Datei `OracleXEClient.exe`, die selbstextrahierende Installationsdatei.

Tipp

Wenn Sie noch nicht wirklich vertraut sind mit Oracle bzw. Oracle Express, macht es an dieser Stelle vielleicht Sinn, zunächst das Kapitel 3 durchzuarbeiten. Sie finden wertvolle Informationen zum Umgang mit der Datenbank und ihrer Administration sowie eine Beschreibung einiger nützlicher Programme.

2.6.1 Die Installation

Die Installation ist denkbar einfach, starten Sie einfach die ausführbare Datei `OracleXEClient.exe`, den Rest erledigt der Installer für Sie (vgl. Abb. 2.11).



Abb. 2.11: Installation des XE-Clients

Nach wenigen Klicks über *Weiter* und der Bestätigung der obligatorischen Lizenzvereinbarung gelangen Sie in den einzigen Dialog der Installationsroutine, in dem Sie Parameter eingeben können. Wie in Abbildung 2.12 zu sehen, bestimmen Sie hier das Installationsverzeichnis.

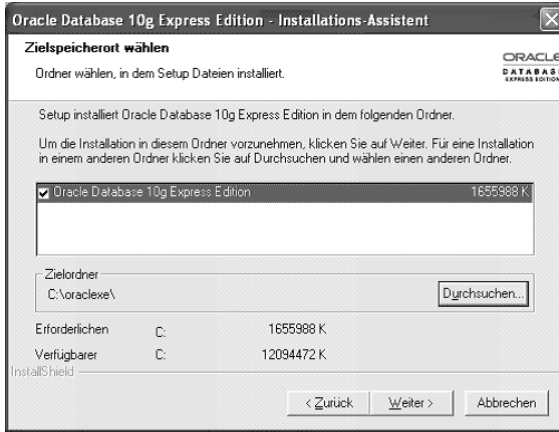


Abb. 2.12: Definition des Installationsverzeichnis

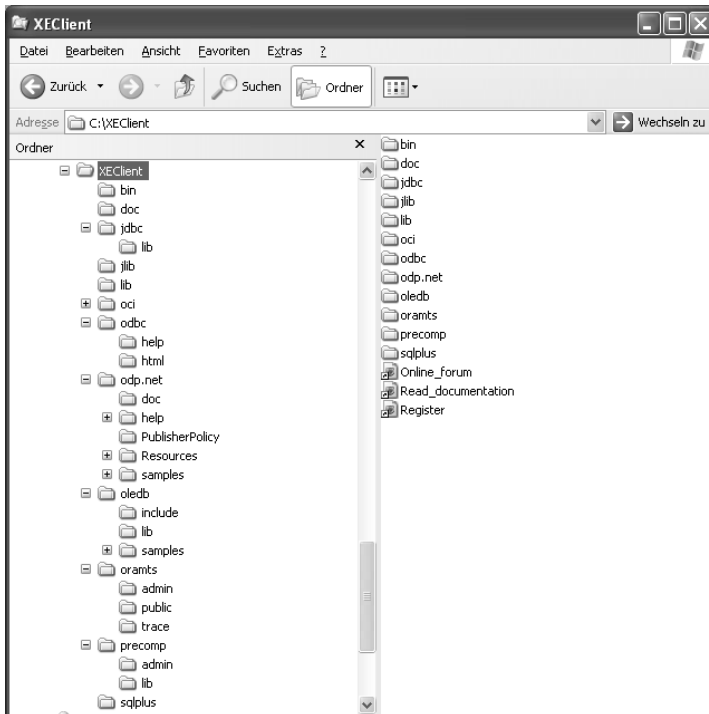


Abb. 2.13: Verzeichnisstruktur des XE-Clients

Nach drei weiteren Dialogen, die Sie einmal über *Weiter* und dann über *Installieren* erreichen, ist die Installation auch schon fertig. Vorausgesetzt, Sie haben den Zielordner bei `c:\oracle\exe` belassen, finden Sie nach erfolgter Installation dort eine Verzeichnisstruktur analog zu der in Abbildung 2.13.

Im `bin`-Verzeichnis finden Sie neben einigen anderen, weniger wichtigen Programmen unter anderem auch SQL*Plus. In den anderen Verzeichnissen finden Sie Bibliotheks-Dateien bzw. Treiber für die unterschiedlichsten Programmiersprachen. Angefangen bei dem JDBC-Treiber für Java über ODBC bis hin zu .NET finden Sie hier die entsprechenden Bibliotheken. Teilweise sind die entsprechenden DLLs selbst allerdings auch im `bin`-Ordner zu finden.

Im Startmenü wurden vier Punkte eingetragen:

- *Dokumentation lesen*
- *Für Online-Forum registrieren*
- *Gehe zu Online-Forum und*
- *SQL-Befehlszeile ausführen*

Tipp

SQL*Plus ist umfassend im Abschnitt 3.2 des folgenden Kapitels beschrieben.

Die oben beschriebene Verzeichnisstruktur definiert allerdings nicht die einzige Stelle, an der der Installer Daten bzw. Dateien abgelegt hat. Analog zu der Installation des Servers selbst schreibt die Installationsroutine in das `SystemRoot`-Verzeichnis (in der Regel `c:\windows`) eine Log-Datei (`OracleDatabaseXEClient-Install.log`), in der die Installation protokolliert ist.

Darüber hinaus wird das `bin`-Verzeichnis mit in den Suchpfad eingetragen. Ebenfalls über die Installation wird die Umgebungsvariable `SQLPATH` auf `c:\xeclient\sqlplus` gesetzt. Hier ist allerdings Vorsicht angebracht. Sollte die Variable vor Installation schon gesetzt sein, wird sie gnadenlos überschrieben!

Zu guter Letzt werden auch noch einige Parameter in der Registrierungsdatenbank hinterlegt. Unter `HKEY_LOCAL_MACHINE | SOFTWARE | ORACLE | KEY_XE-CLIENT` finden Sie Einträge, die auch in Abbildung 2.14 zu sehen sind. Lassen Sie sich in der Abbildung nicht von den anderen Einträgen unter `ORACLE` irritieren. Diese werden Sie in dieser Form auf Ihrem Rechner wohl nicht vorfinden. Es handelt sich hier um Relikte aus anderen Oracle-Installationen, die ich auf dem Entwicklungsrechner vorgenommen habe.

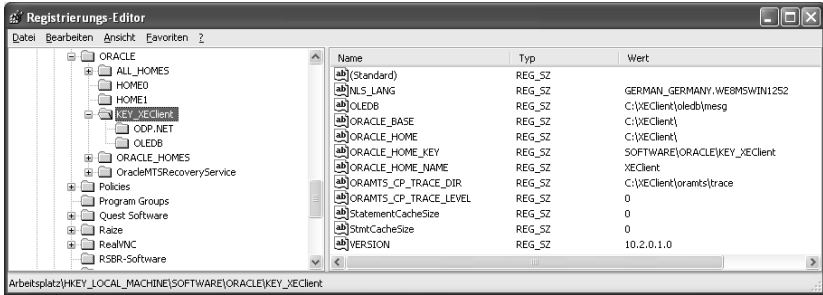


Abb. 2.14: XE-Client-Einträge in die Registrierungsdatenbank

2.6.2 Oracle XE – bitte melden

Nachdem nun die Client-Software vollständig auf einem weiteren Rechner installiert ist, geht es jetzt um die Herstellung einer Kommunikation mit der Datenbank auf dem Server. Da mit der Client-Software auch SQL*Plus auf dem Rechner Einzug gehalten hat, werden wir zunächst damit eine Verbindung herstellen und eine einfache Abfrage durchführen.

Kommunikation über SQL*Plus

In Abschnitt 2.4.2 haben Sie sich schon SQL*Plus der Server-Installation (und die Zugehörigkeit zur Gruppe ORA_DBA) zunutze gemacht, die Anwendung gestartet und sich mithilfe der Anweisung

```
connect / as sysdba;
```

an der Datenbank angemeldet. Eine Gruppe ORA_DBA wird mit Installation der Client-Software nicht angelegt. Das führt dazu, dass in der connect-Anweisung anstelle des »/« mindestens eine Benutzerkennung nebst Passwort mit angegeben werden muss. Das allein reicht allerdings noch nicht aus. Man muss zusätzlich definieren, wohin die Reise gehen soll, also mit welchem Rechner bzw. mit welcher Datenbank man eine Kommunikation aufbauen möchte.

Hier die vollständige Syntax der connect-Anweisung:

```
username/passwort@host[:port] [/service_name]
```

host ist der Name des Rechners (bzw. seine IP-Adresse), auf dem die Oracle-XE-Datenbank läuft. Optional dazu kann noch die Portnummer mit angegeben werden. Die Standardinstallation von XE erstellt einen Listener-Prozess, der auf der Portnummer 1521 »horcht«. An dieser Stelle darf ich auf den Abschnitt 2.3, die

Installation des Datenbank-Servers, verweisen. Optional kann zusätzlich ein Service-Name angegeben werden, mit dem der Client sich verbinden soll. Je Rechner/Portnummer kann es also mehrere Services geben. Die Angabe eines Service-Namens ist optional. Wird keiner angegeben, wird der *Default Database Service* während der Anmeldung ermittelt.

Werfen Sie dazu einen Blick in die `tnsnames.ora` der XE-Installation:

```
XE =
  (DESCRIPTION =
    (ADDRESS = (PROTOCOL = TCP)(HOST = obelix)(PORT = 1521))
    (CONNECT_DATA =
      (SERVER = DEDICATED)
      (SERVICE_NAME = XE)
    )
  )
```

Hier sehen Sie die Konfiguration meiner persönlichen XE-Installation. Der Datenbankserver ist auf einem Rechner namens `obelix` installiert, der Listener-Prozess horcht auf dem Port 1521 und als Service-Name ist XE konfiguriert. Hinter dem Rechnernamen verbirgt sich eine IP-Adresse. Wenn Sie nun von einem Client-Rechner heraus eine Verbindung zu – in diesem Fall – `obelix` aufbauen möchten, müssen Sie sicherstellen, dass der Client zu dem Namen eine IP-Adresse auflösen kann. Dies überprüfen Sie am einfachsten, indem Sie auf dem Client an der Kommandozeile die Anweisung

```
ping obelix
```

eingeben. Es sollte eine Ausgabe ähnlich der folgenden erscheinen:

```
C:\>ping obelix

Ping obelix [192.168.0.3] mit 32 Bytes Daten:

Antwort von 192.168.0.3: Bytes=32 Zeit<1ms TTL=128
Antwort von 192.168.0.3: Bytes=32 Zeit<1ms TTL=128
Antwort von 192.168.0.3: Bytes=32 Zeit<1ms TTL=128
Antwort von 192.168.0.3: Bytes=32 Zeit<1ms TTL=128

Ping-Statistik für 192.168.0.3:
    Pakete: Gesendet = 4, Empfangen = 4, Verloren = 0 (0% Verlust),
    Ca. Zeitangaben in Millisek.:
```

```
Minimum = 0ms, Maximum = 0ms, Mittelwert = 0ms
```

```
C:\>
```

Erscheint jedoch eine Meldung wie »Ping-Anforderung konnte Host "xyz" nicht finden, überprüfen Sie den Namen und versuchen Sie es erneut.«, ermitteln Sie die IP-Adresse des Servers (z.B. über die Anweisung `ipconfig` an der Kommandozeile) und geben diese dann als Parameter der `ping`-Anweisung an:

```
ping <IP-Adresse>
```

Damit sollte die Kommunikation aber in jedem Fall funktionieren; falls nicht, haben Sie ein Netzwerkproblem.

Starten Sie jetzt SQL*Plus über das Startmenü. Am SQL-Prompt geben Sie jetzt ein:

```
connect system/<Passwort>@<Rechnername>
```

Dabei müssen Sie natürlich `<Passwort>` durch Ihr bei der Installation definiertes Passwort und `<Rechnername>` durch Ihren Rechnernamen des Servers ersetzen.

Sollten Sie an dieser Stelle eine Fehlermeldung über einen »Timeout« erhalten, dass also keine Verbindung aufgrund einer Zeitüberschreitung hergestellt werden konnte, sollten Sie die Sicherheitseinstellungen, insbesondere die einer evtl. installierten Firewall, überprüfen. Beispielsweise erhalten Sie unter Windows XP mit Installation des Service Packs 2 eine Firewall, die nach der Installation aktiv gesetzt ist und per Voreinstellung keine Verbindungsanforderungen von außen zulässt. Das genau ist aber erforderlich und deshalb muss hier mindestens der Port 1521 freigeschaltet werden. Sie könnten nun die gesamte Firewall deaktivieren oder nur einen bestimmten Port zusätzlich freischalten. Ich empfehle die letztere Alternative.

Gehen Sie dazu in die Systemsteuerung und rufen Sie dort den Punkt *Windows-Firewall* auf (wie gesagt, diesen Eintrag finden Sie dort nur, wenn das Service Pack 2 installiert ist). Auf der Karteikarte *Ausnahmen* (vgl. Abb. 2.15) klicken Sie nun auf *Port*. In dem hier erscheinenden Dialog können Sie für den freizuschaltenden Port einen Namen vergeben. Natürlich ist hier auch die Portnummer einzutragen (vgl. Abb. 2.16).

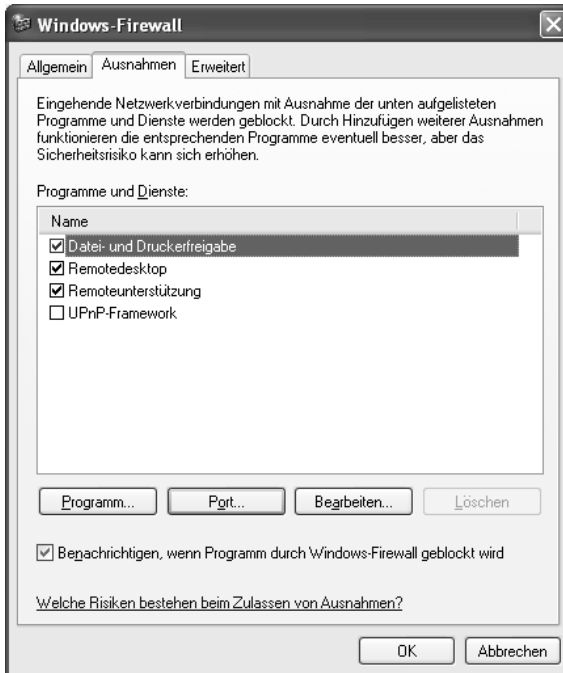


Abb. 2.15: Konfiguration der Windows-Firewall

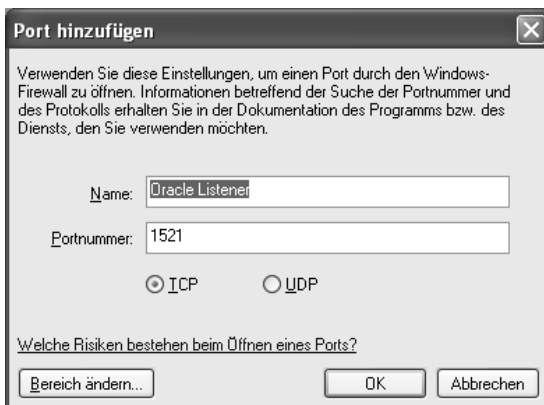


Abb. 2.16: Port des Oracle-Listeners freischalten

Danach sollte die Verbindung zur Datenbank über diesen Port möglich sein.

Kommunikation über die Datenbank-Homepage

Sie können selbstverständlich auch über andere Wege mit der Datenbank von einem Client-Rechner aus kommunizieren. So ist es z.B. auch möglich, alle Tools der Datenbank-Homepage zu nutzen. Lokal auf dem Server wird die Datenbank-Homepage über die URL

```
http://127.0.0.1:8080/apex
```

gestartet. Von einem Client aus können Sie die (gleiche) Homepage über die Adresse

```
http://<Rechnername>:8080/apex
```

erreichen, wobei der <Rechnername> analog zu der Kommunikation via SQL*Plus durch Ihren konkreten Rechnernamen bzw. die IP-Adresse des Servers zu ersetzen ist. Allerdings sind vor einem erfolgreichen Start der Datenbank-Homepage von einem Client aus noch zwei Dinge zu erledigen.

- Freischaltung des Ports
- Freischaltung des http-Zugriffs innerhalb der Datenbank

Im vorherigen Abschnitt haben Sie gelernt, den Port 1521 in der Firewall freizuschalten. Ähnliches gilt hier auch. Der Port 8080 muss bei einer evtl. installierten Firewall freigeschaltet werden, bevor die Datenbank-Homepage von einem entfernten Rechner gestartet werden kann.

Das reicht aber in diesem Fall nicht. Innerhalb der Datenbank müssen Sie zusätzlich noch den http-Zugriff auf diese freischalten. Melden Sie sich dazu an der Datenbank-Homepage (auf dem Server selbst) mit dem `system`-Account an. Rufen Sie dann das Menü *Administration* auf. Auf dieser Seite finden Sie rechts in der Tasks-Liste den Link *HTTP-Zugriff verwalten*. Über diesen Link gelangen Sie in einen Dialog, in dem Sie den entsprechenden Zugriff freischalten können (vgl. Abb. 2.17).

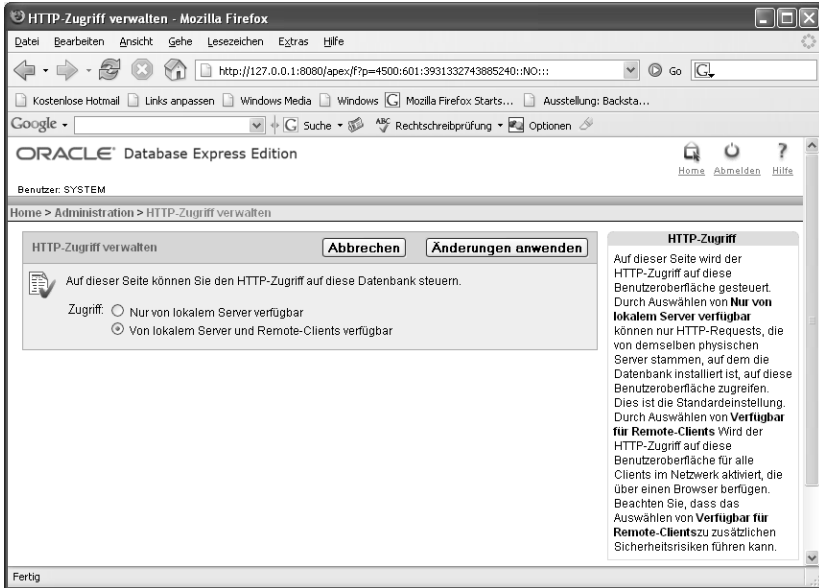


Abb. 2.17: Verwaltung der http-Zugriffe

Danach sollten Sie die Datenbank-Homepage bequem von jedem beliebigen Client aus erreichen.

2.7 Der Instant-Client

Die Installation der XE-Client-Software war schon recht einfach, dennoch fragt man sich manchmal, weshalb denn auf jedem Rechner, der mit einer XE-Datenbank kommunizieren soll, eine 75-MB-Installation vorgenommen werden muss. Mir ging es zumindest so. Stellen Sie sich vor, Sie haben eine kleine Datenbankanwendung geschrieben, bei der XE als Server-System zum Einsatz kommt. Neben der Installation des Servers müssen Sie beim Kunden auch das Frontend auf beispielsweise 25 Rechnern zum Laufen bringen. Auf jedem dieser Rechner müssen Sie also auch den XE-Client installieren. Oracle bietet zu diesem Zweck auf seiner Homepage Installationsdateien an, die eine so genannte *Silent Installation* ermöglichen. Eine solche *stille Installation* erfordert keine Benutzereingriffe mehr. Alle vom Benutzer eingebaren Parameter, z.B. das Installationsverzeichnis, können in einer Skript- oder Parameter-Datei hinterlegt werden, die die Installationsroutine dann ausliest. Dadurch könnten Sie die Verteilung des XE-Clients zwar automatisieren, aber wie sieht die Sache aus, wenn Ihr Frontend z.B. auch noch eine Vielzahl von Dateien umfasst. Hierfür müssten Sie eine eigene Installationsroutine schreiben, die nach erfolgter XE-Client-Installation ausgeführt werden muss.

Langer Rede, kurzer Sinn: Es geht auch einfacher, auch wenn das auf den ersten Blick nicht so scheinen mag und auch wenn die im Folgenden beschriebene Technologie noch nicht ganz ausgereift zu sein scheint.

Hinweis

Die auf den folgenden Seiten beschriebenen Installationen beziehen sich auf die Version 10.2.0.2 des Instant Clients. Es ist nicht auszuschließen oder gar zu hoffen, dass die beschriebenen Workarounds, insbesondere die der fehlenden DLLs, in höheren Versionen nicht mehr auftreten.

Oracle hat wohl selbst erkannt, dass die Client-Installation der Software – dies gilt übrigens auch für die »großen« Datenbankservers – mittlerweile recht aufgebläht und kompliziert ist. Hier steuert Oracle mit dem Instant Client entgegen. Hierbei handelt es sich um ein bzw. mehrere kleine Programmpakete, die nur wenige Dateien, genauer gesagt, die nur wirklich zum Kommunikationsaufbau benötigten Dateien enthalten. Abhängig von der Art der Kommunikation – eine Java-Anwendung über JDBC oder ein C-Programm über das *Oracle Call Interface* –, laden Sie einfach eine Datei herunter, entpacken diese in ein Verzeichnis, hängen dieses Verzeichnis mit in den Suchpfad und schon kann es losgehen.

Unter <http://www.oracle.com/technology/tech/oci/instantclient/index.html> gibt es die entsprechende Software zum Download. Hier wird grundsätzlich unterschieden zwischen den folgenden Paketen:

- Basic
- Basic Lite
- JDBC Supplement
- SQL*Plus
- ODBC Supplement
- SDK

Das Paket *Basic* umfasst einen Satz an Dateien, die im Prinzip bei jeder Art von Kommunikation mit einer Datenbank benötigt werden. Es schlägt mit rund 30 MB zu Buche. Weniger umfangreich, dafür nicht mit lokalisierten Fehlermeldungen (und einem anderen kleinen Unterschied, auf den ich noch zu sprechen komme) kommt das Paket *Basic Lite* daher. Eines dieser beiden Pakete benötigen Sie immer, falls Sie sich für eine Installation des Instant Clients entscheiden. Darauf aufbauend benötigen Sie zusätzlich noch eines der Pakete *JDBC* oder *ODBC Supplement*, *SQL*Plus* oder gar das *SDK*.

Auf den folgenden Seiten möchte ich die Installation des Instant Clients erläutern, um darüber eine ODBC-Verbindung zur Datenbank aufzubauen. Auf der Buch-CD

finden Sie sämtliche Pakete im Ordner *Instant Client*. Packen Sie sie in ein beliebiges Verzeichnis. Diese Zip-Datei enthält die folgenden Dateien:

- classes12.jar
- ojdbc14.jar
- oci.dll
- ocijdbc10.dll
- ociw32.dll
- oranzsbb10.dll
- oraocci10.dll
- oraociei.dll

Die letzte Datei ist der wohl größte Unterschied zur Basic-Lite-Version; dort ist die letztgenannte Datei nicht enthalten, dafür eine andere. Allerdings wird genau diese letzte Datei später bei der Installation des ODBC-Treibers benötigt. Entgegen der Beschreibung auf der Website kann man auf der Basic-Lite-Version 10.2.0.2 keinen ODBC-Treiber aufsetzen! Sie benötigen also zwingend das Basic-Paket. Laden Sie jetzt zusätzlich das Paket *Instant Client Package – ODBC* herunter und entpacken Sie es in das gleiche Verzeichnis wie zuvor. In der in diesem Paket enthaltenen ODBC_IC_Readme_Win.html können Sie nachlesen, dass die ODBC-Registrierung nach den Dateien oci.dll, oraociei10.dll, oranzsbb10.dll sucht und Letztere gibt es nun einmal nur in dem großen Paket.

Erweitern Sie jetzt noch die Pfad-Umgebungsvariable auf Ihrem System um das zuvor gefüllte Verzeichnis. Sie müssen jetzt nur den ODBC-Treiber im Windows-System registrieren. Dies geschieht einfach durch Ausführung von `odbc_install.exe` aus dem zuvor eingerichteten Verzeichnis.

Das war es schon (fast). Um eine Kommunikation zur Datenbank zu ermöglichen, muss neben dem Instant Client natürlich noch ein Stück individuelle Software in Form einer neu zu erstellenden Datei mitgegeben werden. Erstellen Sie zu diesem Zweck die (Text-)Datei `tnsnames.ora`. Sie können diese im gleichen Installationsverzeichnis des Instant Clients ablegen. Fügen Sie der Datei die folgenden Zeilen hinzu:

```
XE =
(DESCRIPTION =
  (ADDRESS = (PROTOCOL = TCP)(HOST = obelix)(PORT = 1521))
  (CONNECT_DATA =
    (SERVER = DEDICATED)
    (SERVICE_NAME = XE)
  )
)
```

Beachten Sie dabei, dass der Host-Name `obelix` auf Ihren Systemen durch den jeweils korrekten Host-Namen zu ersetzen ist. Der Port 1521 muss natürlich entsprechend angepasst werden, sofern der Listener-Prozess auf Ihrem Datenbank-Server auf einem anderen Port lauscht.

Damit die im Folgenden beschriebene Einrichtung des XE-ODBC-Treibers diese Datei mit den entsprechenden Einstellungen auch findet, bedarf es einer weiteren Umgebungsvariablen. Erzeugen Sie dazu die Umgebungsvariable `TNS_ADMIN` und weisen Sie ihr den Pfad zu, in dem die zuvor erzeugte `tnsnames.ora` liegt.

Damit sind alle Voraussetzungen – na ja, fast alle – geschafft, wir können uns an die Einrichtung des ODBC-Treibers machen. Die Einrichtung eines ODBC-Treibers erfolgt in der Computerverwaltung unter dem Eintrag *Datenquellen (ODBC)*. Wenn Sie darauf klicken starten Sie den *ODBC-Datenquellen-Administrator*. Über *Hinzufügen* gelangen Sie in einen Dialog ähnlich dem in Abbildung 2.18.



Abb. 2.18: Den XE-ODBC-Treiber einrichten

Wenn Sie hier die Liste nach unten scrollen, finden Sie dort den Eintrag *Oracle in instantclient_10_2*. Dieser wurde automatisch generiert. Wenn Sie jetzt auf *Fertig stellen* klicken, dürfte die gleiche Fehlermeldung wie in Abbildung 2.19 erscheinen. Es handelt sich hier um einen bekannten Bug des Instant Clients. Grund hierfür ist, dass zwei DLLs in dem Instant-Client-Verzeichnis fehlen: `mfc71.dll` und `mscvc71.dll`. Das ist aber im Prinzip kein Problem, denn beide DLLs finden Sie auf Ihrem Datenbankserver im `bin`-Verzeichnis. Kopieren Sie diese einfach in das zuvor eingerichtete Client-Verzeichnis auf dem entsprechenden Rechner.

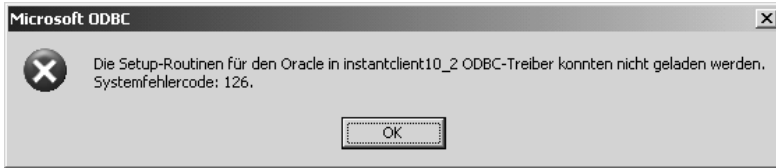


Abb. 2.19: Fehlende DLLs des Instant Clients

Sobald diese DLLs vorhanden sind, probieren Sie erneut, den Oracle-Treiber hinzuzufügen. Jetzt gelangen Sie in den nächsten Dialog, der auch in Abbildung 2.20 abgebildet ist. Neben der Vergabe eines Namens sowie einer Beschreibung ist vor allem die Definition eines *TNS Service Namens* wichtig. Diesen können Sie allerdings nicht manuell eingeben, sondern nur aus der Listbox auswählen. Hier erscheint der Eintrag *XE* deshalb, weil Sie zuvor die *tnsnames.ora* angelegt und über die Umgebungsvariable *TNS_ADMIN* bekannt gemacht haben.

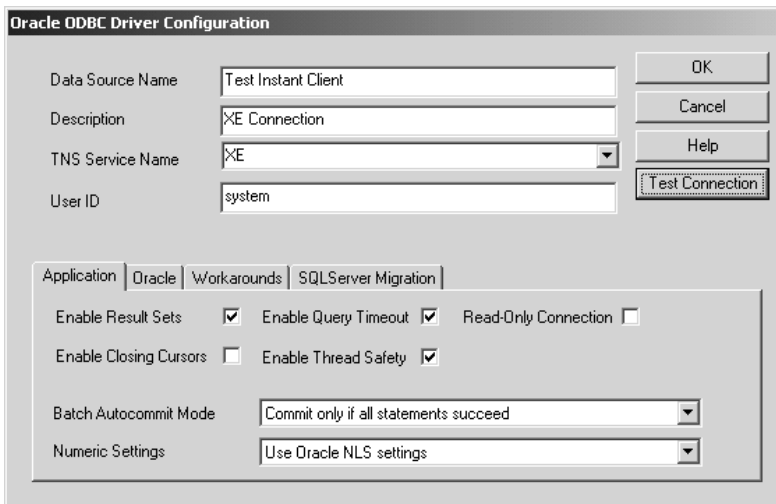


Abb. 2.20: Konfiguration der Parameter

Damit sind im Prinzip alle notwendigen Einstellungen vorgenommen. Probieren Sie es doch einfach über *Test Connection* aus. Sie werden noch kurz nach Benutzer und Passwort gefragt, doch dann sollte ein Dialog wie in Abbildung 2.21 erscheinen.



Abb. 2.21: Verbindung hergestellt!

Erlauben Sie mir noch ein kurzes Fazit. Auch wenn die Installation des Instant Clients noch einige kleinere Haken und Ösen haben mag, halte ich es dennoch für ein Stück gelungene Software. Im Hinblick auf eine anstehende Softwareverteilung einer Datenbankanwendung, also Client-Software nebst Frontend, haben Sie hiermit die Möglichkeit, alle Komponenten in eine eigene Installationsroutine mit Tools wie *InstallShield* oder *InnoSetup* zu gießen. Darüber hinaus – und das halte ich für noch wichtiger – behält man als Entwickler den Überblick darüber, wie die Komponenten miteinander im Zusammenspiel stehen und welche Dateien wofür benötigt werden. In Zeiten immer komfortablerer Entwicklungs- und Integrations-tools und einhergehend damit immer mehr Software-Schichten kann dieses Know-how von unschätzbarem Wert sein. Gerade wenn es um Themen wie Fehlerbereinigung oder Troubleshooting geht, wenn es also darum geht, ein Problem einzukreisen und zu eliminieren, benötigt man Spezialisten, die wissen, was unter der Motorhaube abgeht.