

# Oracle und Industrie 4.0 – zurück in die Zukunft

Gerhild Aselmeyer, Analysen, Konzepte & Anwendungsentwicklung für EDV

Im Rahmen von Industrie 4.0 könnte zukünftig häufiger mal die Anforderung nach Kenntnissen älterer Oracle-Versionen (7 bis 9) gestellt werden. Aufgrund der sehr viel längeren Lebenszyklen industrieller Anlagen und Maschinen werden diese häufig auf Industrie-Rechnern mit älteren Betriebssystemen als Bedienerkonsolen betrieben. Produktionsdaten lassen sich auch unter diesen Umständen automatisch in eine aktuelle Oracle-Datenbank übertragen, wenn man gewisse Besonderheiten beachtet.

Dieser Artikel zeigt, welche Möglichkeiten die Oracle-Software bietet, um Rechner mit geringerer Hardware-Ausstattung und älterem Betriebssystem (DOS/ Windows NT/ 2000/ XP) als Client für eine aktuelle Oracle-Datenbank einzusetzen. Grundlegend dafür sind die von Oracle gepflegte Rückwärts-Kompatibilität sowie die Werkzeug-Konstanz für Netzwerk (SQL\*Net) und Programmierung (Precompiler PRO\*C, PRO\*Cobol und OCI). Außerdem unterstützt Oracle ein Betriebssystem über dessen gesamten Lebenszyklus mit neuen Versionen. Die Client-Version 10g hat die offizielle Freigabe; für die Professional- und Enterprise-Editionen Windows 2000, 11g ab Windows XP und für 12c benötigt man mindestens Windows 7.

Eines gilt es bei den nachfolgenden Ausführungen zu beachten: Bei den Clients 9i oder älter handelt es sich um nicht offiziell von Oracle unterstützte Versionen für Datenbanken unter neuer Server-Software (11g/12c). Die installierten Clients lassen sich (ausschließlich) für die angegebenen Operationen unter Beachtung der beschriebenen Beschränkungen nutzen.

## Welche Oracle-Version für welches Betriebssystem

Betrachten wir zunächst ältere Betriebssysteme, die Hardware-Voraussetzungen und die jeweils letzten dafür freigegebenen Versionen der Oracle-Client-Software (siehe Tabelle 1). Neueren Windows-Betriebssystemen braucht keine besondere Aufmerksamkeit geschenkt zu werden, da sich unter diesen Client-Software von Versionen einsetzen lässt, die keine Kompatibilitätsprobleme erwarten lassen. Die Interoperabilität ist laut der von Oracle veröffentlichten Matrix gegeben, auch wenn die älteren einsetzbaren Versionen nur noch bedingt unterstützt werden. Damit erhält man zwar keine Fehler-Korrekturen, doch ist es unwahrscheinlich, dass

Fehler bei einfachen Datentransfers (SQL, DML) bei korrekter Konfiguration von SQL\*Net überhaupt auftreten.

Für neuere Linux-Kernel (ab 2.4.21 / glibc 2.2.4) als RHEL- oder SuSE-Distribution steht bereits die Client-Software 9.2.0.4 oder neuer zur Verfügung. Allerdings gelten für einige Pakete besondere Anforderungen an deren Version – ob auch bei einer reinen Client-Installation, hat die Autorin bisher nicht getestet. An-

Betriebssystem	Hardware-Voraussetzungen	Oracle-Version
<b>DOS/ Windows 3.1 Windows NT 3.5.1 (16-bit)</b>	Prozessor 80486 RAM 8 MB Platte < 300 MB	7.3.4 (16-bit Required Support Files)
<b>Windows NT 4.0 SP1 - SP4</b>	Prozessor mindestens: 80486 empfohlen: Pentium um 133/ 166  RAM mindestens: 32 MB empfohlen: 54 MB  Platte < 300 MB	8i / 8.1.5
<b>Windows NT 4.0 SP5</b>	Prozessor mindestens: Pentium 166/ 200 empfohlen: Pentium 266  RAM mindestens: 128 MB empfohlen: 256 MB  Platte FAT: 450 MB (Runtime) NTFS: 200 MB	9.2.0.1
<b>Linux Kernel 2.2</b>	Prozessor X86 RAM 64 MB Platte (Programmer) 275 MB	8i / 8.1.7

Tabelle 1

dererseits hat sie im Zusammenhang mit Industrie-Rechnern noch nichts von Linux als Betriebssystem gehört. Unabhängig von Betriebssystem und Oracle-Client-Version gehören zu den notwendigen Hard- und Software-Voraussetzungen die Netzwerk-Fähigkeit des Rechners/der Bedienkonsole, also die Existenz einer Netzwerkkarte und Unterstützung von TCP/IP.

## Was sich kombinieren lässt

Die Frage nach den möglichen Kombinationen aus Oracle-Client- und Server-Versionen/-Editionen muss sowohl technisch als auch lizenzrechtlich betrachtet werden. Bei der technischen Interoperabilität von Versionen benötigen die Editionen (XE, PE, SE1, SE, EE) keine gesonderte Behandlung, da die Code-Basis für alle identisch ist; lizenzrechtlich hingegen müssen die Bedingungen für einen weiteren Client gemäß den Bestimmungen für die einzelnen Editionen beachtet werden. Zunächst die technischen Voraussetzungen: *Abbildung 1* zeigt die in My Oracle Support veröffentlichte Matrix für die Unterstützung zur Interoperabilität.

Es gibt folgende generelle Anmerkungen, die Oracle zur Interoperabilität macht:

- Für Datenbank-Links muss die Interoperabilität in beide Richtungen gewährt sein.
- Nicht unterstützte Kombinationen können funktionieren. Fehlermeldungen hierzu werden aber nicht bearbeitet.
- Da neuere Datenbank-Server mit einem begrenzten Set älterer OCI-Clients kompatibel sind, muss die Client-Software nicht unbedingt gleichzeitig mit der Server-Software ein Update erfahren. Allerdings lassen sich neue Eigenschaften der Server-Software dann nicht nutzen.
- Die Matrix gilt auch über verschiedene Plattformen und Prozessor-Releases (32/64-Bit) hinweg, außer wenn die von Oracle veröffentlichte Notiz zur Unterstützung einer Plattform oder eines Prozessors etwas anderes besagt.

Im Folgenden gilt das Hauptaugenmerk den Client-Versionen, die in der Mat-

rix für Datenbankserver der Versionen 11g/12c mit „No“ gekennzeichnet sind, da hierfür nie Unterstützung von Oracle geboten wurde. Die übrigen Markierungen geben lediglich den Stand der von Oracle zum gegenwärtigen Zeitpunkt noch geleisteten Unterstützung wieder; es waren oder sind aber von Oracle freigegebene Kombinationen.

Auf den ersten Blick scheinen die Entwickler bei Oracle von den bisher beobachteten Prinzipien der Rückwärts-Kompatibilität bei der neuesten Version 12c abzuweichen, wenn man „#8“ in der Legende zu *Abbildung 1* liest. Ein Blick auf die Beschreibung der Parameter für SQL\*Net in der „sqlnet.ora“ lässt die Hoffnung zu, dass zumindest Clients bis zurück zu 8i (8.1.7) konfigurierbar sind:

- SQLNET.ALLOWED\_LOGON\_VERSION (ab 10g)
- In 12c ersetzt durch
- SQLNET.ALLOWED\_LOGON\_VERSION\_SERVER
- SQLNET.ALLOWED\_LOGON\_VERSION\_CLIENT

Ob und welche Probleme mit noch älteren Client-Versionen auftreten, hat die Autorin bisher nicht verifiziert. In diesem Fall lässt sich mithilfe einer Gateway-Datenbank der Version 11g der Datentransport bewerkstelligen; die Verbindung von 11g- und 12c-Datenbanken über Datenbank-Link wird nämlich unterstützt. Mit Gateway-Datenbanken zur Überbrückung von Versionen beim Einsatz von Datenbank-Links hat die Autorin bereits positive Erfahrungen gemacht. Aber auch bei der Konfiguration von SQL\*Net für eine 11g-Datenbank müssen spezielle Einstellungen für ältere, nicht offiziell unterstützte Client-Anwendungen vorgenommen beziehungsweise überprüft werden. Das betrifft vor allem die Datei „sqlnet.ora“.

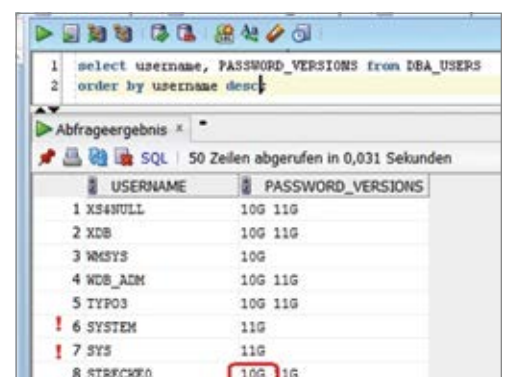
Da mit Version 8i eine Zuverlässigkeitsprüfung („Authentisierung“) bei der Netz-Kommunikation eingeführt wurde, gibt es seitdem den Parameter „SQLNET.AUTHENTICATION\_SERVICES“, den ältere Clients nicht kennen. Außerdem haben sich die zulässigen Werte von 8i über 9i bis 10g teilweise geändert:

- 8i  
beq, none, all, kerberos5, cybersafe, securid, identix

- 9i  
beq, none, all, kerberos5, cybersafe, radius
- 10g  
beq, none, all, kerberos5, radius, nts

Client Version	Server Version				
	12.1.0	11.2.0	11.1.0	10.2.0	10.1.0
12.1.0	Yes	Yes	ES	LES #7	No #3
11.2.0	Yes	Yes	ES	LES #7	No
11.1.0	ES	ES	ES	LES #7	Was #6
10.2.0	LES #7	LES #7	LES #7	LES	Was
10.1.0(#4)	No	Was #6	Was #6	Was	Was
9.2.0	No #0	Was #5	Was #5	Was #5	Was
9.0.1	No	No	No	No	Was
8.1.7	No	No	No	Was	Was
8.1.6	No	No	No	No	No
8.1.5	No	No	No	No	No
8.0.6	No	No	No	No	No
8.0.5	No	No	No	No	No
7.3.4	No	No	No	No	No

*Abbildung 1: Aktuelle Unterstützung für Interoperabilität. Legende: „Yes“ = unterstützt, „ES“ = Bugfixes nur für Kunden mit „Extended Support“, „LES“ = Bugfixes nur für Kunden mit „Limited Extended Support“- oder „Market-Driven Support“-Vertrag, „Was“ = Unterstützung für mindestens eine Seite ausgelaufen, „No“ = keine Unterstützung. Spezielle zusätzliche Anmerkungen: „#3“ = Korrekte Anzeige des Fehlers ORA-3134, „#4“ = Probleme, wenn eine der beiden Seiten auf EBCDIC Characterset basiert, „#5“ = Client-Version min. 9.2.0.4, „#6“ = Patchsets bei Datenbank-Links beachten, „#7“ = Patchsets bei Datenbank-Links beachten, „#8“ = der Zugriffsversuch von Client 9.2 auf eine Datenbank 12.1 schlägt fehl mit der Meldung „ORA-28040: No matching authentication protocol“ error.*



*Abbildung 2: Passwort-Versionen*

Betreibt man den Datenbankserver mit einem Linux-Betriebssystem, dann gibt es für die serverseitige Einstellung von „SQLNET.AUTHENTICATION\_SERVICES“ mehrere Möglichkeiten:

- SQLNET.AUTHENTICATION\_SERVICES=(all)
- SQLNET.AUTHENTICATION\_SERVICES=(beq, all)
- SQLNET.AUTHENTICATION\_SERVICES=(nts, all)
- SQLNET.AUTHENTICATION\_SERVICES=(none)

Mit „all“ lässt man alle Netz-Authentisierungen zu, die der Server kennt, mit „none“ keine außer der Datenbank-gestützten mit Username/Passwort, „nts“ steht für das proprietäre Verfahren NTLM zur Authentisierung von Microsoft und „beq“ für die native Authentisierung über das Betriebssystem für alle außer Microsoft Windows.

Damit steht für Client-Anwendungen basierend auf älteren Oracle-Clients der Versionen 7.3.x, 8.0.x immer die ihnen bekannte Anmeldeform zur Verfügung.

Läuft der Datenbank-Server mit einem Windows-Betriebssystem, gibt es für Client-Anwendungen der Version 7.3.x/8.0.x nur die Einstellungsmöglichkeit „SQLNET.AUTHENTICATION\_SERVICES=(none)“. Alle anderen Einstellungen führen dazu, dass vor der Datenbank-gestützten Anmeldeform eine Authentisierung über NTLM versucht wird; dies bewirkt allerdings auf Client-Seite ein unerwartetes Ende der Kommunikation (ORA-03113), der Datenbank-Prozess beansprucht daraufhin 100 Prozent der Server-CPU und legt diese damit lahm. Bei etwas jüngeren Client-Anwendungen (8i, 9i) kann dann auch „kerberos5“ zum Einsatz kommen.

Ab der Datenbank-Version 10g gibt es weiterhin den Parameter „SQLNET.ALLOWED\_LOGON\_VERSION“ mit „8“ als

Voreinstellung bis 11g, ab 12c ist die Voreinstellung dann „11“. In der von Oracle mitgelieferten Beispieldatei sucht man diesen vergeblich; aber wenn er in der „sqlnet.ora“ steht (beziehungsweise ab 12c), muss der Eintrag „SQLNET.ALLOWED\_LOGON\_VERSION=8“ lauten, um älteren Client-Anwendungen den Zugriff zu ermöglichen.

Für die Konfiguration von SQL\*Net auf dem Client-Rechner können weder „sqlnet.ora“ noch „tnsnames.ora“ einfach vom Server kopiert werden – das wohl häufigste Vorgehen beim Nutzen dieser Methode. Die „sqlnet.ora“ lässt sich unbearbeitet aus der Installation nutzen. Die Einträge für Aliasse mit TCP/IP in der „tnsnames.ora“ haben sich nicht umfangreich verändert, es sind nur weitere Möglichkeiten der Adressierung hinzugekommen. Daher lassen sich die aktuellen Einträge für die notwendigen Datenbanken weitgehend übernehmen, jedoch ist der Eintrag „GLOBAL\_NAME“ zu entfernen, denn globale Namen gibt es erst seit Version 8i. Zuletzt gilt es noch, mit der Abfrage „select username, PASSWORD\_VERSIONS from DBA\_USERS“ die Passwort-Version für das Schema zu prüfen, auf das die älteren Client-Anwendungen zugreifen sollen (siehe Abbildung 2).

Zugriff mit älteren Client-Anwendungen gibt es nur, wenn das entsprechende Schema auch eine Passwort-Version „10G“ besitzt, wie das Schema „STRECKO“ in Abbildung 2. Da es sich hierbei um den schwächsten Verschlüsselungs-Algorithmus handelt, sollten mindestens die Schemata „SYS“ und „SYSTEM“ ausschließlich die höchste Passwort-Version besitzen; deshalb auch die Markierung „!“ bei diesen Schemata. Nachdem SQL\*Net jetzt die entsprechende Konfiguration besitzt, kann als erster Test eine Anmeldung mit SQL\*Plus 3.3.3 (Datenbank-Version 7.3.3) erfolgen (siehe Abbildung 3).

Da der Datenbank-Katalog immer nur erweitert wurde und die Basis-Tabellen dafür unverändert blieben, lassen sich bekannte Katalog-Views wie „USER\_TABLES“ fehlerfrei abfragen. Neuere Katalog-Tabellen und damit die darüberliegenden Views enthalten allerdings auch für das alte SQL\*Plus Unverständliches (siehe Abbildung 4).

Anwender-Regeln haben frühestens mit Version 8 Einzug gehalten, denn bei dem undefinierten Datentyp für „RULE\_CONDITION“ handelt es sich um einen CLOB, diesen Datentyp gibt es ab Version 8.0.4. Weitere Details zu Einschränkungen bei Cli-

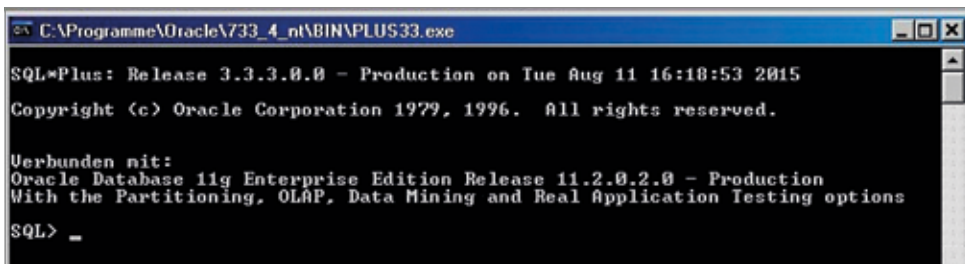


Abbildung 3: Anmeldung mit SQL\*Plus 3.3.3 an einer Oracle-Datenbank 11g

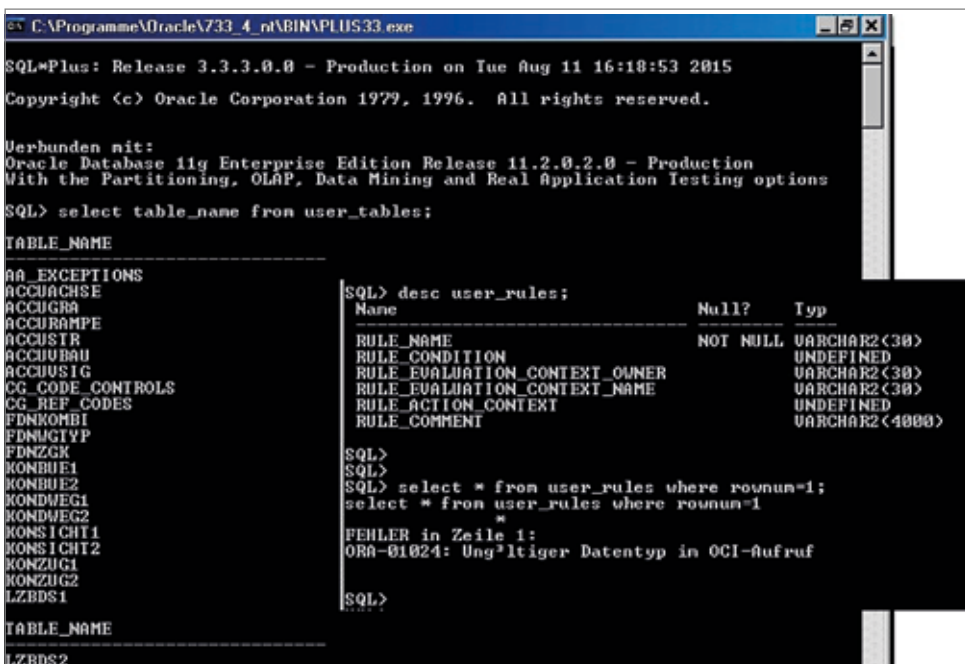


Abbildung 4: Katalog abfragen mit SQL\*Plus 3.3.3 bei einer Datenbank 11g

ent-Anwendungen, basierend auf den Bibliotheken älterer Oracle-Versionen, folgen später, da dies bei der Entwicklung von Anwendungen in Programmiersprachen der dritten Generation (3-GL) zu beachten ist.

## Die Lizenzierung

Wenden wir uns an dieser Stelle zunächst noch der lizenzrechtlichen Seite zu. Am einfachsten sind die beiden Fälle Personal Edition (PE) und Enterprise Edition (EE) mit Prozessor-Lizenz:

- **PE**  
Hier kann wohl zu Testzwecken (quasi unter OTN-Lizenz) ein älterer, noch verfügbarer Client beziehungsweise SQL\*Net und SQL\*Plus installiert sein, ein Betrieb ist ausgeschlossen.
- **EE/Prozessorlizenz**  
Liegt ein Installationsmedium für eine ältere Version vor, können die Client-Software installiert und Client-Anwendungen betrieben werden – solange der Server dafür nicht um einen Prozessor erweitert wird.

Etwas genauer hinsehen muss man bei EE mit Lizenzierung benannter Anwender (Named User Plus) sowie bei Standard Edition (SE) und Standard Edition One (SE1). Die über die Konsole (Industrie-Rechner) gesteuerte Maschine gilt für die Oracle-Lizenzierung als ein Benutzer; daher ist für diese Installation eine zusätzliche Nutzer-Lizenz erforderlich. Im Fall von SE und SE1 muss noch überprüft werden, ob man durch den zusätzlichen Nutzer eine vorgegebene Höchst-Anzahl für die jeweilige Edition überschreitet. Eine dafür notwendige Erweiterung des Servers mit zusätzlicher CPU kann zur Folge haben, dass die lizenzierte Edition nicht mehr ausreicht.

Liegt eine gültige Lizenz vor, dann ist die Installation einer älteren Software-Version ein Downgrade, was lizenzrechtlich unbedenklich sein sollte, allerdings benötigt man natürlich noch ein Installationsmedium. Firmen, die bereits seit Langem Oracle-Datenbanken einsetzen, haben ältere Versionen möglicherweise im Archiv. Wer einen gültigen Support-Vertrag besitzt, kann über „My Oracle Support“-Medien oder einen Link zum Herunterladen ältere Software-Versionen anfordern.

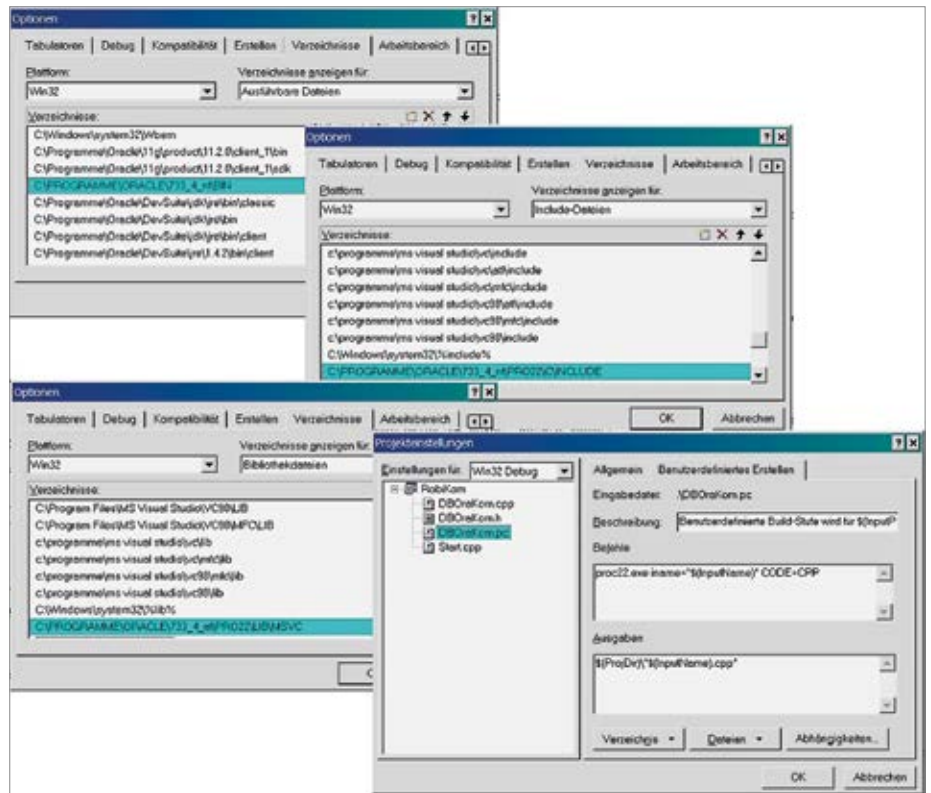


Abbildung 5: Pro\*C 2.2 einbinden in Visual Studio

Eine Besonderheit bildet Express Edition (XE). Es gibt zwar keine Benutzer-Beschränkung, aber die Lizenz enthält nicht die Precompiler (Pro\*C, Pro\*Cobol) sowie SQLJ. Damit berechtigt diese Lizenzierung nicht zur Programmierung. Nach Auskunft von Oracle LMS gibt es aber keine Beschränkungen bezüglich der eingesetzten Client-Version für extern entwickelte Software, wenn entsprechende Installationsmedien vorliegen.

## Eingebettetes SQL in 3-GL-Programmiersprachen

Zunächst benötigt man die Programmierumgebung der alten Oracle-Version, bestehend aus Header-Dateien, Bibliotheken und Pre-Compiler. Dafür muss das zur Oracle-Version gehörende Pro\*C (Pro\*Cobol ist wohl eher ungeeignet) in der modernen Entwicklungsumgebung installiert sein.

Da Microsoft wenig von Rückwärts-Kompatibilität hält, hat die Autorin die Installation unter Windows 7 nur über den Umweg über einen Rechner mit Windows XP und nachträglichem Kopieren der Installation (Ordner sowie die notwendigen Einträge in der Registrierung) geschafft.

Anschließend ließ sich der Pre-Compiler dann aber anstandslos ins Visual Studio einbinden. Dabei ist darauf zu achten, dass man alle Pfade berücksichtigt und beim benutzerdefinierten Erstellen für die PC-Dateien besser die meist auch vorhandene „PROC[n].EXE“ verwendet, um sicher zu sein, den richtigen Pre-Compiler anzusprechen (siehe Abbildung 5).

Leider erhält man unter Windows 7 bei der Ausführung der erzeugten Programme schon bei der Ausführung des eingebetteten Befehls „EXEC SQL CONNECT :psUser IDENTIFIED BY :psPW USING :psDB;“ eine „nicht abgefangene Ausnahme in ... (NTDLL.DLL): 0xC0000008; Invalid Handle“. Daher ist das Debuggen höchstens remote möglich.

## Die Programmierung

Bei der Programmierung gilt die besondere Beachtung den verfügbaren Datentypen und Datenbankobjekten. In allen Versionen ab 7.3.3 gibt es folgende Datentypen:

- VARCHAR2 bis mindestens 2.000 Zeichen
- CHAR bis mindestens 255 Zeichen

- NUMBER
- LONG bis 2 GB
- DATE
- RAW bis mindestens 255 Zeichen
- LONG RAW bis 2 GB
- ROWID

Ab Version 8.0.4 können ein „VARCHAR2“ bis 4.000 Zeichen sowie „CHAR“ und „RAW“ bis 2.000 Zeichen Länge definiert sein. Außerdem gibt es ab dieser Version zusätzlich „NCHAR“ bis 2.000 Byte sowie „CLOB“, „NCLOB“, „BLOB“ und „BFILE“ mit jeweils bis 4 GB Fassungsvermögen. In Version 8i (8.1.5) existiert zusätzlich „UROWID“ mit einer Länge bis zu 4.000 Byte.

Ab Version 9i (9.0.1) muss bei den Längen-Angaben für „VARCHAR2“ und „CHAR“ noch nach „BYTE“- und „CHAR“-Semantik unterschieden werden. Außerdem gibt es jetzt auch „TIMESTAMP“ sowie die Datums-/Zeit-Intervalle. Auf diese Beschränkungen ist je nach der eingesetzten Client-Version nicht nur bei der Deklaration im Quellcode zu achten, sondern auch beim Anlegen der für das Projekt infrage kommenden Datenbankobjekte wie Tabellen, Views und Funktionen; weitere Objekte wie Prozeduren oder Packages hat die Autorin bisher nicht eingesetzt.

Bei Funktionen müssen die Ein- und Ausgabeparameter den Beschränkungen genügen; das Innere der Funktion interessiert lediglich die Datenbank und läuft damit in der Umgebung der Server-Software. Wenn man diese Einschränkungen beherzigt, kann mit einer Funktion die Stelle eines Werts in einer durch Doppelpunkte separierten Liste in einem SQL\*Plus der Oracle-Version 7.3.3 abgefragt werden (siehe Listing 1).

Obwohl zurzeit von Oracle 7.3.3 keiner etwas von HTMLDB/Apex auch nur ahnte, erhält man eine fehlerfreie Ausgabe, wie *Abbildung 6* zeigt. Die Funktion in Listing 2 ist Teil eines einfachen Daten-Transfers von Statusmeldungen eines Roboters in eine Datenbank-Tabelle „ROBI\_LOG“.

„Pre-compiliert“ mit „PROC22“ und gebunden mit den Bibliotheken „SQLLIB18.LIB2“ und „SQLLIB18.LIB“, liefert die Anwendung einwandfrei Daten aus einer alten 7er-Oracle-Client-Umgebung in einer 11g-Datenbank ab – allerdings nur unter älteren Windows-Betriebssystemen (Testumgebung: Windows XP). *Abbildung 7* zeigt die in der Tabelle „ROBI\_LOG“ abgespeicherten Daten.

```
create or replace FUNCTION GetIndex4Value
(ValueString IN varchar2, Value IN varchar2)
return number
AUTHID CURRENT_USER
is
    TableOfValues HTMLDB_APPLICATION_GLOBAL.VC_ARR2
                := HTMLDB_UTIL.STRING_TO_TABLE(ValueString);
    nInd          number := 1;
begin
    while ( nInd <= TableOfValues.count and
           TableOfValues(nInd) <> Value ) loop
        nInd := nInd +1;
    end loop;
    if ( nInd > TableOfValues.count ) then nInd := 0; end if;
    return nInd;
end;
```

Listing 1

```
int speicherLog(char* pcBez, double nWert)
{
    EXEC SQL BEGIN DECLARE SECTION;
    char* psBezeichnung = pcBez;
    double nWertangabe = nWert;
    double nZeitDiff;
    int nLfdNr;
    EXEC SQL END DECLARE SECTION;

    EXEC SQL select sysdate - RL_ZEITSTEMPEL Zeit_Diff,
                    max(RL_LFD_NR)
                    into nZeitDiff, nLfdNr
                    from robi_log
                    where RL_ZEITSTEMPEL
                        = (select max(RL_ZEITSTEMPEL)
                          from robi_log)
                    group by sysdate - RL_ZEITSTEMPEL;
    if ( nZeitDiff != 0 ) { nLfdNr = 0; }
    else { nLfdNr ++; }

    EXEC SQL insert into ROBI_LOG
                (RL_ZEITSTEMPEL, RL_LFD_NR, RL_BEZ, RL_WERT)
                values (sysdate, :nLfdNr, :psBezeichnung,
                       :nWertangabe);

    if ( sqlca.sqlcode == 0 )
    {
        EXEC SQL commit;
    }

    return sqlca.sqlcode;
}
```

Listing 2

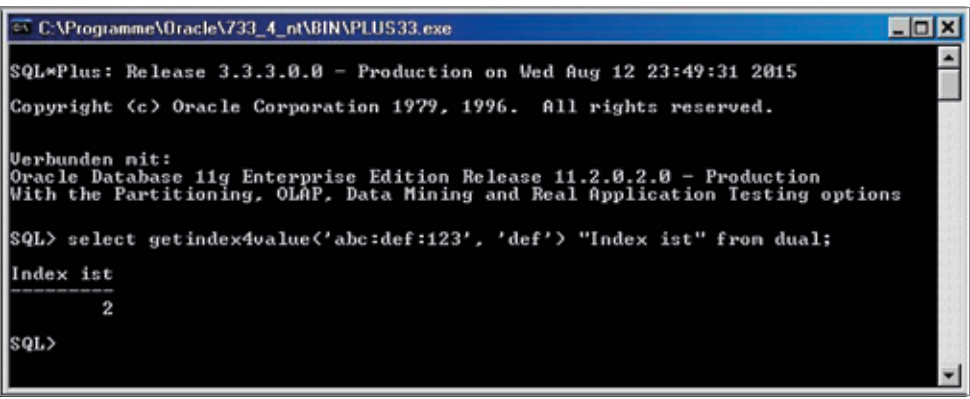


Abbildung 6: Funktionsabfrage in SQL\*Plus 3.3.3

```

1 alter session set nls_date_format = 'dd.mm.yyyy hh24:mi:ss';
2 select * from ROBI_LOG where to_char(RL_ZEITSTEMPEL, 'yyyymmdd') = 20150812
3 order by RL_ZEITSTEMPEL desc, RL_LFD_NR desc;
4

```

RL_ZEITSTEMPEL	RL_LFD_NR	RL_BEZ	RL_WERT
12.08.2015 20:49:44	10	Maschine Stop	1
12.08.2015 20:49:44	9	Aktueller Lärmspegel	541
12.08.2015 20:49:44	8	Aktueller Lärmspegel	517
12.08.2015 20:49:44	7	Aktueller Lärmspegel	539
12.08.2015 20:49:44	6	Aktueller Lärmspegel	552
12.08.2015 20:49:44	5	Spannung	5,324
12.08.2015 20:49:44	4	Maschine Start	0
12.08.2015 20:49:44	2	Maschine Stop	1
12.08.2015 20:49:44	1	Aktueller Lärmspegel	557
12.08.2015 20:49:44	0	Aktueller Lärmspegel	540
12.08.2015 20:47:18	10	Maschine Stop	1

Abbildung 7: Daten in der Tabelle „ROBI\_LOG“

Die Spalte „RL\_ZEITSTEMPEL“ ist selbstverständlich als „DATE“ deklariert; deshalb benötigt man zusätzlich die Spalte „RL\_LFD\_NR“, um eine exakte Abfolge der Meldungen zu erhalten.

### Fazit

Die meisten Probleme bereiten Windows-Betriebssysteme bei der Aufgabe, ältere Maschinen durch Datenbank-Anbindung

in eine automatisierte Umgebung zu integrieren. Erhält man von Oracle Unterstützung oder zumindest ein gewisses Entgegenkommen bei der Beschaffung der notwendigen Installationsmedien für ältere Oracle-Versionen (7 bis 9), dann ist die Aufgabe lösbar. Es lässt sich auch in solchen Umgebungen ein weiterer Schritt in Richtung „Industrie 4.0“ machen.



Gerhild Aselmeyer  
g.aselmeyer@aka-edv.de

Alles, was die SAP-COMMUNITY wissen muss, finden Sie monatlich im E-3 MAGAZIN.  
Ihr WISSENSVORSPRUNG im Web, auf iOS und Android sowie PDF und Print:  
[e-3.de/abo](http://e-3.de/abo)

# Wer nichts weiß, muss alles glauben!

Marie von Ebner-Eschenbach



SAP® ist eine eingetragene Marke der SAP AG in Deutschland und in den anderen Ländern weltweit.

[www.e-3.de](http://www.e-3.de)