SQL Report

Report Generator zu HP ELOQUENCE

ODBC Treiber

Ausgabe:
A.01.00 - August 1996
A.01.00 - September 1996

Übersicht

Dieses Handbuch beschreibt die Installation und Administration des **SQL/R** ODBC Treibers.

Das Handbuch ist in folgende Kapitel unterteilt:

Kapitel 1 **Einleitung**

Eine kurze Beschreibung von ODBC allgemeine Informationen über den **SQL/R** ODBC Treiber.

Kapitel 2 **Server Installation und Konfiguration**

Dieses Kapitel beschreibt die Installation, Konfiguration und Administration des ODBC Servers.

Kapitel 3 **Client Installation und Konfiguration**

Dieses Kapitel beschreibt die Installation und Konfiguration des ODBC Treibers auf dem Client System.

Kapitel 4 **Treiber Referenz**

Dieses Kapitel enthält Referenzinformationen zu dem **SQL/R** ODBC Treiber.

Typographische Konventionen

Solange nicht anders angegeben, werden in diesem Handbuch die folgenden symbolischen Konventionen verwendet:

`Computer Font`	Computer Font markiert Kommandos, Schlüsselworte, Optionen, Konstanten, Programmanweisungen, Ausgaben und Dateinamen.
[ ]	Das Symbol [ ] markiert eine Taste oder eine Schaltfläche auf dem Bildschirm („PushButton"), die mit der Maus angewählt wird. [CTRL] bezeichnet zum Beispiel die `Strg` („Control") Taste und [Weiter] eine Schaltfläche auf dem Bildschirm.
[CTRL]-[char]	[CTRL]-[char] markiert ein Steuerzeichen. Zum Beispiel [CTRL]-[Y] bedeutet, daß Sie *gleichzeitig* die Tasten `Strg` und `Y` auf der Tastatur drücken.
italics	Innerhalb von Syntaxbeschreibungen repräsentiert ein schräggestelltes Wort einen formalen Parameter oder ein Argument, das Sie durch den aktuellen Wert ersetzen müssen. Ersetzen Sie in dem folgenden Beispiel *filename* durch dem Namen der Datei, die sie drucken möchten: `lp` *filename*
[ ]	Innerhalb von Syntaxbeschreibungen werden optionale Elemente in eckigen Klammern eingeschlossen. In dem folgenden Beispiel bedeuten die eckigen Klammern um `[-d`*dev*`]`, daß der Parameter optional ist: `lp [-d`*dev*`]` *filename*

Zusätzliche Literatur

Für weiterführende Informationen möchten wir Sie auf folgende Zusatzliteratur hinweisen:

ODBC 2.0, Programmers Reference and SDK Guide

Die definitive Beschreibung des ODBC Standards. Das Buch beinhaltet eine komplette Beschreibung des ODBC API, ODBC Programmierbeispiele (in C) und behandelt das ODBC SDK.

Microsoft ODBC 2.0 programmer's reference and SDK guide : for Microsoft Windows and Windows NT. Microsoft Press, ISBN 1-55615-658-8

Dokumentation zu HP-UX (online)

Verweise im Handbuch in der Form `services`(4) beziehen sich auf den entsprechenden Eintrag (hier `services`) im angegebenen Abschnitt (hier Abschnitt 4) der HP-UX- Referenz Dokumentation. Mit Hilfe des `man` Kommandos ist diese Dokumentation auch onlione verfügbar. Für `services`(4) geben Sie bitte folgendes Kommando ein:

```
man 4 services
```

SQL Einführung

ODBC baut auf SQL auf. ODBC spezifiziert den Mechanismus, mit dessen Hilfe der Austausch von SQL-Befehlen und Daten zwischen den Client Programmen und der Datenbank erfolgt, und zwar in einer von der Datenbank unabhängigen Art und Weise. Die meisten ODBC Client Programme bieten weiterhin die Möglichkeit an direkte SQL-Befehle an die Datenbank zu senden. Direkt bedeutet, daß diese SQL-Befehle dem SQL-Dialekt der Datenbank entsprechen. Da ein SQL Tutorial die Ziele dieses Handbuches sprengen würde, sollte der Leser eine Einführung in SQL lesen, um eine optimale Nutzung von **SQL/R** ODBC zu erzielen.

Inhaltsverzeichnis

1 Allgemeines

1.1 SQL/R ODBC Treiber

Der **SQL/R** ODBC Treiber ist eine Erweiterung zu **SQL/R**, die Ihnen den online Zugriff auf HP Eloquence Datenbanken aus jeder ODBC kompatiblen Anwendung erlaubt.

Dies vermeidet einerseits redundante Datenhaltung und ermöglicht andererseits dem Benutzer den schnellen und einfachen Zugriff auf die aktuellen Firmendaten.

1.2 Einführung in ODBC

Aus historischen, strategischen und technologischen Gründen sind in Firmen oftmals Anwendungen und Daten auf unterschiedlichen Plattformen und mit unterschiedlichen Datenbanken installiert. Microsoft Open Database Connectivity (ODBC) ist die de facto standardisierte Schnittstelle für den Zugriff auf Daten in einer heterogenen Umgebung von relationalen und nicht relationalen Datenbank Systemen.

ODBC basiert auf einer Spezifikation, die von einem Konsortium aus über 40 Firmen (Mitglieder des SQL Access Group u.a.) entwickelt wurde und findet eine breite Unterstützung von Anwendungsentwicklern und Datenbankherstellern.

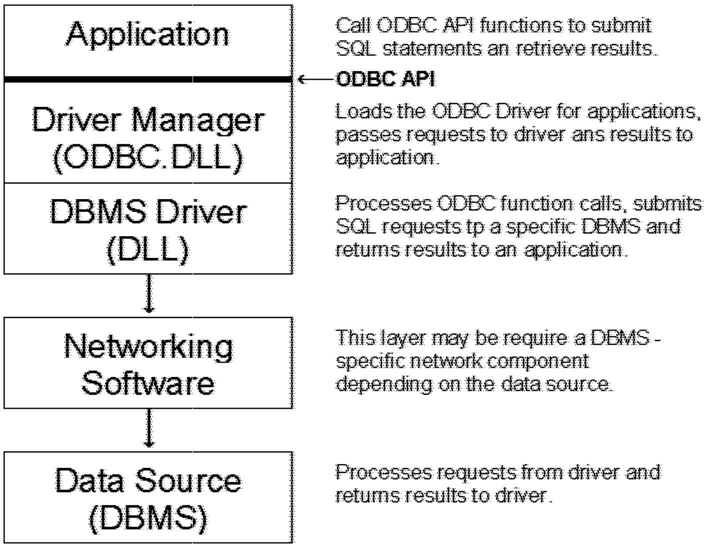
ODBC bietet eine offene und herstellerunabhängige Möglichkeit des Zugriffes auf Daten, die in unterschiedlichen Datenbanken und auf Plattformen (Personal Computer, Minicomputer und Mainframes) gespeichert sind. Mit ODBC kann eine Anwendung gleichzeitig auf Daten von mehreren unterschiedlichen Datenbanken zugreifen. ODBC bietet Entwicklern und Anwendern ferner viele signifikante Vorteile durch seinen offenen und standardisierten Datenzugriff an:

- ODBC erlaubt Firmen ihre existierenden Datenbanken zu verwenden und mittels ODBC aus gängigen Anwendungen auf die Datenbanken zuzugreifen.
- ODBC Anwender profitieren von der Tatsache, daß mehr und mehr Anwendungen externe Verbindungen mit zusätzlichen Datenbanken ermöglichen. Somit ist der einfache Zugriff auf eine riesige Menge von Firmendaten gegeben.

- ODBC erlaubt dem Anwender aus einer einzigen Anwendung (z.B. MS Access) heraus Zugriffe auf Daten aus mehreren Quellen (zum Beispiel mehreren Servern).
- ODBC ermöglicht Zugriffe auf Daten in unterschiedlichen Datenbank Systemen (wie z.B. DB2, Oracle, HP Eloquence und dBASE) aus der selben Anwendung heraus.
- ODBC stellt ein API bereit, das offen, standardisiert und Herstellerneutral ist.
- ODBC schützt die bereits erfolgten Investitionen von Firmen und Softwareherstellern in vorhandene Datenbanken.
- ODBC basiert auf der Schnittstelle (Call Level Interface, CLI) der Gruppe SQL Access Group (SAG) und stellt eine standardisierte SQL Sprache zur Verfügung, die auf dem ANSI-Standard basiert.

1.3 Wie funktioniert ODBC

ODBC definiert ein API. Jede Anwendung benutzt dieselben Befehle bzw. Funktionen, die in der API Spezifikation definiert sind, um mit Hilfe von Datenbank spezifischen Treibern mit unterschiedlichen Datenquellen zu kommunizieren. Die Kommunikation zwischen der Anwendung und dem eigentlichen Treiber erfolgt über den Treiber Manager. In Windows sind der Treiber Manager und der Treiber jeweils als dynamische Bibliotheken (DLLs) implementiert.



introa

Die Anwendung ruft die ODBC Funktionen auf, um eine Verbindung mit einer Datenquelle herzustellen, Daten zu empfangen, Daten zu senden und die Verbindung dann wieder zu schließen.

Der Treiber Manager stellt der Applikation zusätzliche Infomationen zur Verfügung wie z.B. die verfügbaren Datenquellen und lädt die benötigten Treiber dynamisch nach Bedarf. Darüber hinaus bietet er die Konfiguration von Datenquellen an, überwacht und steuert die Kommunikation der Anwendung und des entsprechenden Treibers.

Der Treiber, der unabhängig von unterschiedlichen Anwendungen entwickelt wurde, sitzt zwischen der Anwendung und dem Netzwerk. Der Treiber bearbeitet ODBC Funktionsaufrufe und überwacht den Datenaustausch zwischen einer Anwendung und einer spezifischen Datenbank. Ferner kann der Treiber die Standard SQL-Syntax in die spezifische SQL-Syntax der Zieldatenbank übersetzen. Alle SQL Übersetzungen liegen in der Verantwortlichkeit des Treiber-Entwicklers. Anwendungen sind nicht nur an einen Treiber für die Kommunikation gebunden.

Eine einzige Anwendung kann mehrere Verbindungen mit unterschiedlichen oder gleichen Datenquellen aufbauen.

Um den Zugriff auf eine neues Datenbank System zu ermöglichen, installiert der Benutzer oder Administrator einen Treiber für diese Datenbank. Der Benutzer braucht keine andere Version der Anwendung, um Zugriff auf diese neue Datenbank zu erlangen. Dies ist ein erheblicher Vorteil für den Anwender und bewirkt signifikante Kosteneinsparungen für EDV Abteilungen im Bereich Support und Entwicklung.

1.4 Zusammenfassung

ODBC definiert eine Standard SQL-Syntax und eine Reihe von Funktionsaufrufen, die auf der SAG CLI Spezifikation basieren. Diese werden als core grammar („Kernsyntax") und core functions („Kernfunktionen") bezeichnet. Wenn eine Applikation nur die Kernfunktionalität (core functionality) verwendet, ist es nicht erforderlich, spezifische Eigenschaften zu behandeln.

Mit der Kernfunktionalität kann eine Anwendung

- eine Verbindung mit einer Datenquelle aufbauen, SQL Befehle ausführen und die Ergebnisse empfangen.
- Standardisierte Fehlermeldungen empfangen.
- eine gemeinsame Login-Schnittstelle für den Benutzer bereitstellen.
- standardisierte Datentypen verwenden, die durch ODBC definiert sind.
- eine standardisierte Syntax benutzen, die durch ODBC definiert ist.

ODBC definiert zusätzlich erweiterte Funktionen um Anwendungen den Zugriff auf erweiterte Fähigkeiten einer Datenbank auf eine standardisierte Art und Weise zu ermöglichen. Außerdem unterstützt ODBC die Verwendung der Datenbank spezifischen SQL Syntax, damit Anwendungen besondere Fähigkeiten der jeweiligen Datenbank verwenden können.

1.5 Voraussetzungen

Folgende Voraussetzungen müssen für die Installation des **SQL/R** Treibers erfüllt sein.

Voraussetzung auf der Server-Seite (HP 9000):

- HP 9000 Serie 800 oder 700
- HP-UX Release 9.0 oder größer
- HP-UX ARPA Services (TCP/IP Netzwerk unterstützung)
- HP ELOQUENCE Version A.03.10 oder größer
- Mindestens 4MB an freiem Speicherplatz im /opt File-System
- DDS Laufwerk (1.3 GB)
- **SQL/R** Version A.01.50 oder größer

Voraussetzungen auf der Client-Seite:

- PC mit Windows 3.11, Windows for Workgroups 3.11, Windows 95 oder Windows NT
- TCP/IP Netzwerk Protokoll-Stack kompatibel mit dem `WINSOCK 1.1` Standard.

Die Auslieferung erfolgt nur auf 60m DDS Kassetten (für die Server Software) und auf 3,5" (1,44 MB) Disketten (für die Client software).

1.6 Lizenz

Jeder Server (HP 9000) benötigt eine Lizenz für **SQL/R** und **SQL/R** ODBC. Der **SQL/R** ODBC Server kann nur auf dem System ausgeführt werden, für das die Lizenz erworben wurde.

Die Client (PC) Software kann auf eine unbegrenzten Anzahl von Systemen innerhalb einer Firma oder Institution installiert werden.

Bitte entnehmen Sie den Lizenzbedingungen die genauen Voraussetzungen für den Einsatz von **SQL/R** und **SQL/R** ODBC.

SQL/R ODBC kann zum Zweck der Evaluierung einen Monat getestet werden. Die Evaluierungsversion hat den vollständigen Lieferumfang und die volle Funktionalität.

1.7 SQL/R im Internet

Aktuelle Informationen über **SQL/R** finden Sie im World Wide Web (WWW) unter folgender Adresse:

`http://www.msede.com/public/sqlr/`

2

Server Installation und Konfiguration

Dieses Kapitel beschreibt die Installation von **SQL/R** ODBC auf dem Server System (HP 9000).

Es ist in folgende Bereiche untergliedert:

- Installation von **SQL/R** ODBC auf dem Server
- Konfiguration des Servers
- System Konfiguration
- Administration

2.1 Installation von SQL/R ODBC auf dem Server

Die **SQL/R** ODBC Server Software ist Teil des **SQL/R** Softwarepakets. Wenn Sie bereits **SQL/R** Software Release A.01.50 oder größer installiert haben, wurde die **SQL/R** ODBC Server Software bereits auf Ihrem System installiert. Hinweise zur Installation des gesamten Softwarepaketes finden Sie in der Ergänzung zur **SQL/R** Dokumentation für Version A.01.41 im Kapitel 1.

Um mit **SQL/R** ODBC arbeiten zu können, benötigen Sie einen Lizenschlüssel. Sofern Sie Ihn nicht zusammen mit der **SQL/R** Software erhalten haben, können Sie Ihn von Ihrem Vertriebspartner oder mit beiliegendem Fax Formular direkt bei der Marxmeier Software Entwicklung GmbH anfordern.

Um den Lizenzschlüssel von ODBC einzugeben, gehen Sie wie nachfolgend beschrieben vor:

1. Melden Sie sich als Superuser (root) an.
2. Tragen Sie den Lizenzschlüssel mit einem Editor Ihrer Wahl (z.B. vi) in die Lizenzdatei mit dem Namen `licence` ein. Abhängig von der Version Ihres HP-UX Betriebssystemes finden Sie die Lizenzdatei unter :

   HP-UX 9.x `/opt/sqlr/etc/licence`
   HP-UX 10.x `/etc/opt/sqlr/licence`

   Mit Hilfe des Programms /opt/sqlr/etc/chklic können Sie Ihre Lizenzdatei überprüfen.

Bitte beachten Sie: Der Lizenzschlüssel muß genauso eingegeben werden, wie er auf dem Lizenzblatt ausgedruckt ist.

2.2 Konfiguration des Servers

Dieser Abschnitt ist in die folgenden Bereiche untergliedert:

- Konfiguration des Servers
- Server Sicherheit
- Netzwerk Optionen
- Server Optionen
- Server Protokoll Optionen

2.2.1 Konfiguration des Servers

Die **SQL/R** ODBC Server wird mit Hilfe der Datei `odbc.cfg` konfiguriert. Abhängig von der HP-UX Version finden Sie die Datei `odbc.cfg` in folgenden Verzeichnissen:

HP-UX 9.x `/opt/sqlr/etc/odbc.cfg`
HP-UX 10.x `/etc/opt/sqlr/odbc.cfg`

Die Konfigurationsdatei wird beim Start des **SQL/R** ODBC Server eingelesen.

Die Konfigurationsdatei `odbc.cfg` besteht aus einzelnen Abschnitten mit Konfigurationseinträgen. Jeder Abschnitt beginnt mit einer Überschrift, die in eckigen Klammern (`[...]`) steht. Momentan wird nur der Abschnitt `[config]` verwendet.

Die Konfigurationseinträge werden in einzeiligen Textzeilen abgelegt. Zeilen die mit einem Gatterzeichen (`#`) beginnen sind Kommentar.

2.2.2 Server Sicherheit

Wir empfehlen Ihnen, eine eigene Benutzerkennung (login) und eine Gruppe für den **SQL/R** ODBC Servers anzulegen, zum Beispiel „`sqlr`".

Wenn Sie eine bereits existierendes Benutzerkennung bzw. Gruppe verwenden möchten, sollten Sie keinesfalls eine verwenden, die normalerweise für Administrationsaufgaben benutzt wird wie z.B. „`bin`", „`daemon`" etc. Der ODBC Server verwendet ein eigenes Authorisierungsschema und obwohl keine Sicherheitsprobleme des ODBC Servers bekannt sind, ist es generell nicht empfehlenswert komplexe Programme mit besonderen Berechtigungen zu starten.

Der **SQL/R** ODBC Server nutzt standardmäßig die Rechte des Benutzers der den Server gestartet hat. Sofern der **SQL/R** ODBC Server mit Superuser-Berechtigung (root capabilities) gestartet wurde, gibt er seine erweiterten Rechte auf und nimmt die Rechte des Benutzers (login) an, der in der Konfigurationsdatei festgelegt wurden. Dies gilt auch für die Rechte der Gruppe.

Eintrag	Beschreibung
`User`	Benutzerkennung (oder Benutzer Nummer) unter der der Server betrieben werden soll. Dieser Eintrag wird ignoriert, sofern der Server nicht mit root Brechtigung gestartet wird. Ein Eintrag ist notwendig sofern der Server mit root Berechtigung gestartet wird.
`Group`	Name der Gruppe (oder Gruppen Nummer) unter der der Server betrieben werden soll. Dieser Eintrag wird ignoriert, sofern der Server nicht mit root Brechtigung gestartet wird. Ein Eintrag ist notwendig sofern der Server mit root Berechtigung gestartet wird.
`Security`	Dieser Konfigurationseintrag definiert die Server Sicherheits Strategie. Es legt fest, wie Benutzer-Namen und Kennwörter geprüft werden. Folgende Optionen stehen zur Verfügung: `Security = odbc` Der Server prüft Kennwörter und Benutzernamen unter Berücksichtigung der Datei `odbc.user`. Dies ist die Voreinstellung.

Zum Beispiel:

```
User = user
Group = group
Security = odbc
```

Wobei *user* der Benutzer ist *group* der Name der Systemgruppe mit deren Berechtigung der Server laufen soll.

2.2.3 Netzwerk Optionen

Eintrag	Beschreibung
`Service`	Der Service-Name (wie in /etc/services definiert) oder die Port Nummer unter der der Server Anfragen erwartet. Der Standardeintrag ist `sqlrodbc`.
`UseKeepAlive`	Ein numerisches Kennzeichen, welches steuert, ob die KEEP ALIVE Socket Option genutzt wird. Mögliche Werte sind 1 (Ja) und 0 (Nein). Die Voreinstellung ist 1. Sofern diese Option aktiviert ist, wird nach einer vom Betriebssystem vorgegebenen Zeit ohne Netzwerkverkehr geprüft, ob der Client noch erreichbar ist.

Zum Beispiel:

```
Service = sqlrodbc
UseKeepAlive = 1
```

2.2.4 Server Optionen

Eintrag	Beschreibung
`dsnFile`	Name und Verzeichnis der Datei odbc.dsn. In dieser Datei werden alle Datenquellen des Servers definiert. Abhängig von Ihrer HP-UX Revision befindet sich die Datei unter: **HP-UX 9.x** `/opt/sqlr/etc/odbc.dsn` **HP-UX 10.x** `/etc/opt/sqlr/odbc.dsn` Weiterführende Informationen über die Datei `odbc.dsn` finden Sie im Abschnitt 2.4.2.

Eintrag | **Beschreibung**

Eintrag | **Beschreibung**

`userFile`

Name und Verzeichnis der Datei `odbc.user`. In dieser Datei werden alle Benutzer hinterlegt, die den **SQL/R** ODBC Server verwenden dürfen.
Abhängig von Ihrer HP-UX Revision befindet sich Datei unter:
HP-UX 9.x `/opt/sqlr/etc/odbc.user`
HP-UX 10.x `/etc/opt/sqlr/odbc.user`
Weiterführende Informationen über die Datei `odbc.user` finden Sie im Abschnitt 2.4.1

`tmpDir`

Angabe des Verzeichnisses unter dem Temporärdateien erstellt werden sollen. Sofern hier nichts festgelegt wurde, wird das vom HP-UX System vorgegebene Verzeichnis verwendet. Weitere Informationen hierzu finden Sie in der online Hilfe zu HP-UX unter tempnam(3).
SQL/R ODBC hat unter Umständen einen größeren Bedarf an temporären Plattenplatz, sofern Sie komplexe Abfragen auf große Datenbanken starten. Um eventuelle Störungen Ihrer anderen Anwendungen zu vermeiden (z.B. durch einen möglichen Überlauf des Dateisystemes) sollten Sie die Temporärdateien von SQL/R in einem von Ihrer Datenbank getrennten Dateisystem erstellen lassen.
Bitte beachten Sie: Sofern Temporärdateien im selben Dateisystem wir Ihre Datenbank erstellt werden, kann es zu Datenbankfehlern kommen, sofern das Dateisystem überläuft.

`panic`

Dieser Eintrag steuert das Verhalten des Servers, wenn ein kritischer Fehler auftritt. Folgende Möglichkeiten stehen zur Verfügung:

`Panic = exit`

Der Prozeß wird abgebrochen. Dies ist die Voreinstellung.

`Panic = dump`

Der Prozeß wird abgebrochen und ein core dump erzeugt.

Eintrag	Beschreibung
`Lang`	Dieser Eintrag definiert die Sprachumgebung, die der Server benutzt. Der Voreinstellung ist „`C`".
`Messages`	Dieser Eintrag definiert die Sprache, in der Meldungen des Servers ausgegeben werden. Sofern nichts anderes definiert wird, wird die Einstellung von Lang übernommen.

Zum Beispiel:

```
dsnFile = /etc/opt/sqlr/odbc.dsn
userFile = /etc/opt/sqlr/odbc.user
tmpDir = /tmp
Panic = exit
Lang = C
Messages = C
```

2.2.5 Server Protokoll Optionen

Der **SQL/R** ODBC Server kann wahlweise das Syslog des Betriebssystems oder eine Datei zur Aufzeichnung ausgewählter Ereignisse verwenden.

Jeder Protokolleintrag hat eine zugeordnete Herkunft und Gewichtung. Der Konfigurationseintrag `LogFlags` steuert welche Einträge ausgegeben oder unterdrückt werden.

Von folgenden Bereichen sind Protokolleinträge verfügbar:

Flag	Bechreibung
`*`	Alle Bereiche
`N`	Netzwerk Kommunikation
`P`	Client / Server Protokoll
`T`	Cursor Steuerung
`S`	Befehls Steuerung und Ausführung
`C`	Konfigurations Subsystem
`K`	SQL Kernel
`Y`	Syntax Prüfung
`I`	Isam Subsystem
`O`	Query Optimizer

Folgende Gewichtungen sind verfügbar:

Gewichtung	Bechreibung
0	Fehlermeldungen
1	Informationen
2	Fehlersuche
3	Ausführliche Fehlersuche

Sofern Syslog verwendet wird, werden die Gewichtungen den folgenden Prioritäten zugeordnet:

Gewichtung	Syslog Priorität
0	LOG_ERR
1	LOG_NOTICE
2	LOG_DEBUG
3	LOG_DEBUG

Weiterführende Informationen hierzu finden Sie in der HP-UX online Dokumentation zu syslogd(1M).

Eintrag	Beschreibung
`LogFile`	

Dieser Konfigurationseintrag spezifiziert entweder den Dateinamen und das Verzeichnis einer Protokolldatei oder eines der nachfolgenden Schlüsselwörter:

`LogFile = console`

Das Protokoll wird auf der Console ausgegeben.

`LogFile = syslog`

Das Protokoll wird zum syslog daemon gesendet.

`LogFile = default`

Die Loginformationen werden in eine vorgegebende Datei geschrieben:

HP-UX 9.x	`/usr/adm/sqlrodbc.log`
HP-UX 10.x	`/var/adm/sqlrodbc.log`

Um diese Option verwenden zu können ist es notwendig den Server mit Root-Berechtigung zustarten.

Der vorgegebene Eintrag ist „syslog".

Eintrag	Beschreibung
`SysIdent`	Syslog Identifier. Der vorgegebene Wert für diesen Eintrag ist „sqlrodbc". Weitere Informationen finden Sie unter syslogd(1M).
`SysFacility`	Syslog Facility (USER/DAEMON/LOCAL0..LOCAL7). Der vorgegebene Wert für diesen Eintrag ist „USER". Weitere informationen finden Sie unter syslogd(1M).
`LogFlags`	Dieser Konfigurationseintrag legt fest welche Ereignisse protokolliert werden sollen. Der vorgegebene Wert für diesen Eintrag ist „*0".

Zum Beispiel:

```
LogFile = syslog
SysIdent = sqlr
SysFacility = USER
LogFlags = "*0"
```

oder

```
LogFile = /tmp/sqlrodbc.log
LogFlags = "*0S1Y1"
```

2.2.6 Beispiel für eine Konfigurationsdatei

Die meisten Konfigurationseinträge müssen nicht explizit definiert werden, da die Vorgabewerte normalerweise ausreichend sind. Das folgende Beispiel zeigt eine vollständige Server Konfigurationsdatei:

```
[config]
User = user
Group = group
tmpDir = /tmp
LogFile = default
LogFlags = "*0"
```

Hierbei ist *user* der System account (Benutzerkennung) and *group* ist der Name der System Gruppe deren Berechtigungen der Server verwenden soll.

2.3 System Konfiguration

Dieser Abschnitt ist in die folgenden Bereiche untergliedert:

- Konfiguration des TCP/IP Transportes
- Konfiguration des Syslog Daemon
- Starten des Servers
- Server Kommandozeilen Argumente
- Automatischer Start des Servers

2.3.1 Konfiguration des TCP/IP Transportes

Der Server und die Clients sind unter Verwendung des TCP/IP Protokolles miteinander verbunden. Um mittels TCP/IP einen Zugriff auf ein Netzwerk zu ermöglichen, muß jeder Service Name einer eindeutigen Port Nummer zugeordet werden.

Dazu fügen Sie bitte auf dem Server folgende Zeile in der Datei `/etc/services` ein:

```
sqlrodbc  8003/tcp  # SQL/R ODBC
```

Dieser Eintrag hat folgende Bedeutung:

- *sqlrodbc* ist der Service name
- *8003* ist eine eindeutige Port Nummer
- *tcp* gibt den Protokoll Namen an
- alle Zeichen nach dem '#' sind Kommentar

Beachten Sie bei der Auswahl der Port Nummer die folgenden Punkte:

- Verwenden Sie *keine* Port Nummer kleiner als 1024.
- Verwenden Sie *keine* Port Nummer, die bereits durch einen anderen Service belegt ist.

- Verwenden Sie *keine* Port Nummer zwischen 5800 und 7500; diese werden vom X-Window System belegt.

Weitere Informationen finden Sie in der HP-UX online Dokumentation unter `services`(4).

Bitte beachten Sie: Die verwendete Port Nummer auf dem Server und dem Client muß identisch sein.

2.3.2 Konfiguration des Syslog Daemon

Sofern die Protkollierung über den Syslog Deamon erfolgt, können die **SQL/R** Meldungen in eine getrennte Datei umgelenkt werden. Dazu ändern Sie bitte die Datei `/etc/syslog.conf`.

Wenn Sie die folgende Zeile in der Syslog Konfigurationsdatei `/etc/syslog.conf` einfügen, werden alle Meldungen der syslog facility `local0` in die Datei `/var/adm/syslog/odbc.log` geschrieben.

```
local0.* /var/adm/syslog/odbc.log
```

In diesem Fall solten Sie `local0` als syslog facility in der Konfigurationsdatei eintragen.

Bitte beachten Sie: Wenn Sie die Syslog Konfigurationsdatei ändern, ist es erforderlich ein `HUP` Signal zum `syslogd` Prozeß senden um Ihm mitzuteilen, daß die Konfigurationsdatei erneut gelesen werden muß.

Dies geschieht mit folgendem Befehl:

```
$ kill -HUP ‘cat /etc/syslog.pid‘
```

Weitere Informationen finden Sie untr syslogd(1m) in der HP-UX online Dokumentation.

Starten des Servers

Der **SQL/R** ODBC Server kann entweder von Internet Daemon Prozeß (inetd) oder als Hintergrundprozeß (daemon) gestartet werden. Bei Verwendung des `inetd` wird bei jedem Zugriff auf den sqlrodbc Service ein sqlrodbcd Prozess gestartet. Falls **SQL/R** ODBC als Hintergrundprozeß gestartet wird, ist er stäendig aktiv und bearbeitet eingehende Verbindungen selbstständig.

Falls **SQL/R** ODBC als Hintergrundprozeß gestartet werden soll, kann der Server entweder manuell oder automatisch beim Start des Systems aktiviert werden.

Ein Hintergrundprozeß ist in der Lage, eine Verbindung schneller aufzubauen, da die Einrichtung des Servers beim Start nur einmal ausgeführt werden muß. Sofern Sie **SQL/R** ODBC täglich und mit mehreren Anwendern nutzen, empfehlen wir Ihnen den Server als Hintergrundprozeß zu starten.

2.3.3 Server Kommandozeilen Argumente

Die **SQL/R** ODBC Server Konfiguration wird durch die Server Konfigurationsdatei `odbc.cfg` bestimmt. Die Kommandozeilen Argumente können dazu verwendet werden, die Konfigurationseinträge zu übergehen.

```
Syntax: sqlrodbcd [Option] [Transport]
Optionen:
 -help       = Alle Möglichkeiten (Diese Liste)
 -d          = Fehlersuche (debug mode)
 -i          = Aufruf über inetd
 -l name     = Name der Logdatei (oder console/syslog/default)
 -f          = Läuft im Vordergrund (foreground)
 -s name     = Service Name (tcp/ip transport)
 -F facility = Syslog facility (USER/DAEMON/LOCAL0..LOCAL7)
 -I ident    = Syslog Bezeichner (identifier)
transport:
 TCPIP       = TCP/IP sockets
```

Folgende Optionen stehen zur Verfügung:

Option	**Beschreibung**
`-help`	Mit dieser Option erhalten Sie eine Liste aller verfügbaren Argumente.
`-d`	Diese Option kann verwendet werden, um abweichende `LogFlags` anzugeben; z.B. zur Eingrenzung eines Problemes. Die Option überlagert den Eintrag `LogFlags`.
`-i`	Sofern der ODBC Server durch `inetd` gestartet wird, muß diese Option gesetzt sein.
`-l` *name*	Diese Option wird verwendet um eine abweiche Protokolldatei (`LogFile`) zu verwenden. Diese Option überlagert den Eintrag `LogFile`.

Option	Beschreibung
`-f[f]`	Mit dem Setzen dieser Option wird der Server im Vordergrund gestartet. Normalerweise läuft der Server im Hintergrund.
`-s` *service*	Überlagert den Eintrag `Service`.
`-F` *facility*	Überlagert den Eintrag `SysFacility`.
`-I` *indent*	Überlagert den Eintrag `SysIdent`.

Der einzige unterstützte Transport ist `TCPIP`.

Weiterführende Informationen finden Sie im Abschnitt zur Server Konfiguration.

2.3.4 Automatischer Start des Servers

Sofern der Server als Hintergrundprozeß (daemon) gestartet wird, können Sie `sqlrodbcd` so konfigurieren, daß er automatisch beim Start des Betriebssystems gestartet wird. Verwenden Sie hierzu eine der nachfolgend aufgeführten Methoden:

HP-UX 9.x

Sie können den Server mit Hilfe von `/etc/rc` starten. In diesem Fall ist es erforderlich die Datei `/etc/rc` zu verändern. Fügen Sie die folgende Zeile im Abschnitt `localrc` oder am Ende der Datei ein:

```
/opt/sqlr/bin/sqlrodbcd
```

HP-UX 10.x

Soll der **SQL/R** ODBC Server durch das Betriebssystem automatisch gestartet oder beendet werden setzen Sie `START_SQLRODBCD=1` in der **SQL/R** Startup Konfigurationsdatei `/etc/rc.config.d/sqlr`.

Um den **SQL/R** ODBC Server manuell zu starten können Sie folgende Anweisung verwenden:

```
/sbin/init.d/sqlr start
```

Um den **SQL/R** ODBC Server manuell zu beenden, können Sie folgende Anweisung verwenden:

```
/sbin/init.d/sqlr stop
```

2.3.5 Server Start mit inetd

Sie können den `sqlrodbcd` Server mit dem `inetd` starten. In diesem Fall müssen Sie in die Konfigurationsdatei `/etc/inetd.conf` folgende Zeile einfügen:

```
sqlrodbc stream tcp nowait user /opt/sqlr/bin/sqlrodbcd sqlrodbcd -i
```

Dieser Eintrag hat folgende Bedeutung:

- *sqlrodbc* ist der Service Name
- *stream* ist der Socket Typ
- *tcp* gibt den Protokoll Namen an
- *nowait* ist die empfohlene Option für stream sockets
- *user* ist der Login Name
- option `-i` für einen Aufruf über `inetd`

Weitere Informationen finden Sie in der HP-UX online Dokumentation unter `inetd.conf`(4).

Bitte beachten Sie: Sofern Sie **SQL/R** ODBC Aufrufe über inetd realisieren, können Sie den Server nicht mehr manuell starten. Die Methoden (inetd und Hintergrundprozeß) schließen sich gegenseitig aus.

2.4 Verwaltung des Servers

Dieser Abschnitt ist in die folgenden Bereiche untergliedert:

- Verwaltung der Benutzer
- Verwaltung der Datenquellen

2.4.1 Benutzerverwaltung

Der **SQL/R** ODBC Server verwendet ein eigenes Authorisierungsschema. Dieses ermöglicht es, einen Datenbankzugriff zu erlauben, ohne notwendigerweise eine eigene HP-UX Butzerkennung für jeden einzelnen Anwender zu haben. Alle Benutzer, die dem Server bekanntgegeben werden, müssen in in der Konfigurationsdatei für Benutzer `odbc.user` definiert werden.

Der Standort der Datei `odbc.user` ist abhängig von der Revision des HP-UX Betriebssystemes:

HP-UX 9.x `/opt/sqlr/etc/odbc.user`
HP-UX 10.x `/etc/opt/sqlr/odbc.user`

Der Dateiname und das Verzeichnis kann durch den Konfigurationseintrag `userFile` in der Serverkonfigurationsdatei geändert werden.

Dateiformat

Die Benutzer Konfigurationsdatei `odbc.user` ist eine Textdatei und kann mit einem Texteditor wie zum Beispiel dem `vi` geändert werden. Leerzeilen und Zeilen, die mit einem Gatterzeichen (`#`) beginnen werden ignoriert.

Die Datei ist in einzelne Abschnitte mit Konfigurationseinträgen unterteilt. Jeder Abschnitt beginnt mit einer Überschrift, die in eckigen Klammern (`[...]`) steht. Die Konfigurationseinträge gelten nur für den entsprechenden Abschnitt. Bei den Überschriften und Konfigurationseinträgen wird die Groß-/Kleinschreibung ignoriert.

Für jeden Benutzer gibt es einen eigenen Abschnitt. Die ODBC Benutzerkennung wird als Überschrift verwendet.

Entry	Description
`Name`	Der volle Name des Benutzers. Dieser Eintrag wird zur Zeit vom Server nicht verwendet.
`Password`	Kennwort des Benutzers. Dieses Kennwort muß vom ODBC Client zusammen mit dem Namen des Benutzers bereitgestellt werden, um den Benutzer zu legitimieren.

Entry	Description

Beispiel

Das folgende Beispiel definiert zwei ODBC Benutzer.

Den Anwender „sample" mit dem Kennwort „secret" und den Benutzer „demo" ohne Kennwort.

```
[sample]
Name = Ein X-beliebiger ODBC Benutzer
Password = secret

[demo]
Name = Demo-Benutzer
```

Bitte beachten Sie: Da unverschlüsselte Kennwörter in der Benutzerkonfigurationsdatei abgelegt sind, sollten Sie den Zugriff auf diese Datei auf den ODBC Server beschränken.

Sie können mit folgenden Befehlen den Besitzer der Datei auf *sqlr* ändern und die Zugriffsrechte auf 400 setzen:

```
chown sqlr odbc.user
chmod a=,u=r odbc.user
```

Wenn der Server als `root` gestartet wurde, können Sie den Besitzer auf root ändern und die die Zugriffsrechte auf 400 setzen:

```
chown root odbc.user
chmod a=,u=r odbc.user
```

2.4.2 Verwaltung der Datenquellen

Der **SQL/R** ODBC Server arbeitet mit „Datenquellen" anstatt Datenbanken. Eine Datenquelle besteht aus einer oder mehreren Datenbanken sowie zusätzlichen Informationen wie Zugriffsrechten und Repository Informationen.

Die dem ODBC Server bekannten Datenquellen müssen in der Datei `odbc.sdn` definiert werden.

Der Standort der Datei `odbc.dsn` ist abhängig von der Revision des HP-UX Betriebssystemes:

HP-UX 9.x `/opt/sqlr/etc/odbc.dsn`
HP-UX 10.x `/etc/opt/sqlr/odbc.dsn`

Der Dateiname und das Verzeichnis kann durch eine Änderung des Eintrages `dsnFile` in der Serverkonfigurationsdatei verändert werden.

Dateiformat

Die Konfigurationsdatei für die Benutzer `odbc.user` ist eine Textdatei und kann mit einem Texteditor wie zum Beispiel dem `vi` geändert werden. Leerzeilen und Zeilen, die mit einem Gatterzeichen (`#`) beginnen werden ignoriert.

Die Datei ist in einzelne Abschnitte mit Konfigurationseinträgen unterteilt. Jeder Abschnitt beginnt mit eine Überschrift, die in eckigen Klammern (`[...]`) steht. Die Konfigurationseinträge gelten nur für den entsprechenden Abschnitt. Bei den Überschriften und den Konfigurationseinträgen wird die Groß-/Kleinschreibung ignoriert.

Jede Datenquelle ist ein eigener Abschnitt. Die Groß-/Kleinschreibung wird ignoriert. Abgesehen davon muß der Name der Datenquelle auf dem Client genau so geschrieben werden wie auf dem Server (einschließlich Leerzeichen).

Entry	Description
`Database`	Name und Verzeichnis der Datenbank.
`Password`	

Entry	Description
	Kennwort der Datenbank. Dies ist ein optionaler Eintrag; der Eintrag kann entfallen, sofern kein Kennwort für den Zugriff auf die Datenbank benötigt wird. Dieser Eintrag bezieht sich auf die Datenbank die vorher im Eintrag `Database` definiert wurde.
`Alias`	Alternativer Name für die Datenbank. Der Eintrag ist optional. Dieser Eintrag bezieht sich auf die Datenbank die vorher im Eintrag `Database` definiert wurde. Wenn ein alternativer Name definiert wird, wird er dem ODBC Client anstatt des eigentlichen Namen der Datenbank zu Verfügung gestellt. Der Name der Datenbank wird vom Client dazu verwendet eine Tabelle in einer Datenbank zu qualifizieren. Durch den alternativen Namen können Sie einen logischen Namen anstatt des eigentlichen Datenbanknamens verwenden (Zum Beispiel "VERKAUF" anstatt "SGLD25". Manche Clients (wie zum Beispiel MS Access) legen Struktur Informationen über die Datenbank lokal ab. Wenn Sie den Namen der Datenbank später ändern (zum Beispiel wegen einer neuen Revision Ihrer Datenbank), können einige Auswertungen fehlschlagen.
`Repository`	Name und Verzeichnis einer Repository Datei. Repository Dateien beziehen sich auf die Datenquelle und werden bearbeitet sofern alle Datenbanken erfolgreich geöffnet wurden. Die Verwendung mehrerer Repository Dateien ist möglich. Der Eintrag ist optional. Eine Repository Datei beinhaltet **SQL/R** Statements wie z. B. `FIELD` und kann zusätzliche Informationen über logische Felder und Formate bereitstellen.
`UserList`	Liste der Benutzer, durch Kommata getrennt. Nur Benutzer, die in dieser Liste aufgeführt sind, sind berechtigt auf die Datenquelle zuzugreifen. Die Benutzerkennung muß in der Datei odbc.user definiert sein. Sofern eine Datenquelle keinen UserList Eintrag enthält kann keine Benutzer darauf zugreifen. Mehrere UserList Einträäge sind möglich. Die Benutzerkennung „`public`" wird vom ODBC Server vordefiniert und muß nicht mehr in der Konfigurationsdatei der Benutzer definiert werden. Sofern die Benutzerkennung „public" in der `UserList` eingetragen wird, kann dies dazu verendet werden, einen anonymen Zugriff auf die Datenquelle zu ermöglichen.

Entry	Description

Beispiel

Das folgende Beispiel definiert die Datenquelle „SQL/R Example". Diese Datenquelle beinhaltet eine Datenbank ohne Kennwort. Der Alias Name lautet „SAMPLE" und wurde für die Datenbank „DB" vergeben. Die Datenquelle hat eine zusätzliche Repository Datei. Nur den Benutzern public und demo ist der Zugriff auf die Datenbank erlaubt.

```
[SQL/R Example]
Database   = /opt/sqlr/share/db.g/db
Password   =
Alias      = SAMPLE
Repository = /opt/sqlr/share/db.g/db.rep
UserList   = public,demo
```

Das folgende Beispiel definiert die Datenquelle „Accounts". Sie beinhaltet zwei Datenbanken (SGL1, SGL2). Jede Datenbank hat ein Kennwort und einen Alias Namen.

```
[Accounts]
Database   = /path/to/db/sgl1
Password   = ALL
Alias      = GL1
Database   = /path/to/db/sgl2
Password   = ALL
Alias      = GL2
Repository = /path/to/file/gl1.rep
Repository = /path/to/file/gl2.rep
Repository = /path/to/file/gl.rep
UserList   = sample
UserList   = gl,stat
```

Bitte beachten Sie: Da in der Datenbankkonfigurationsdatei die Kennwörter der Datenbank unverschlüsselt abgelegt sind, sollten Sie den Zugriff auf dese Datei auf den ODBC Server beschränken.

Sie können mit folgenden Befehlen den Besitzer der Datei auf *sqlr* ändern und die Zugriffsrechte auf 400 setzen:

```
chown sqlr odbc.dsn
chmod a=,u=r odbc.dsn
```

Wenn der Server als `root` gestartet wurde, können Sie den Besitzer auf root ändern und die die Zugriffsrechte auf 400 setzen:

```
chown root odbc.dsn
chmod a=,u=r odbc.dsn
```

3

Client Installation und Konfiguration

Dieses Kapitel beschreibt die Installation von **SQL/R** ODBC auf dem Client-System (PC). Es ist in folgende Abschnitte unterteilt:

- Installation von **SQL/R** ODBC auf dem Client.
- Konfiguration des Client-Systems.
- Administration der Datenquellen.

3.1 Installation von SQL/R ODBC auf dem Client

Die **SQL/R** ODBC Client Software ist nicht kopiergeschützt und darf beliebig oft innerhalb einer Firma oder Institution installiert werden.

Es ist eine 16 Bit und eine 32 Bit Version des **SQL/R** ODBC Treibers verfügbar. Installieren Sie den 16 Bit Treiber mit **Windows 3.x** und den 32 Bit Treiber bei Verwendung von **Windows 95** und **Windows NT**. Wenn Sie 16 Bit Anwendungssoftware auf einer 32 Bit Plattform verwenden (z.B. MS Access 2.0 auf Windows NT), dann benutzt die Anwendungssoftware ebenfalls den 32 Bit Treiber.

Zur Installation der **SQL/R** ODBC Client Side führen bitte Sie folgende Schritte aus:

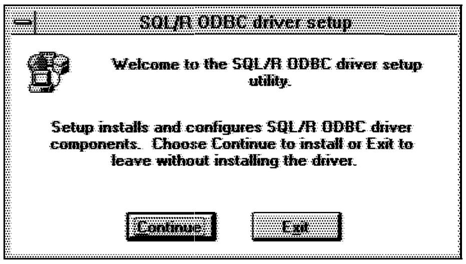
- Starten Sie Windows. Wenn Windows bereits gestartet ist, beenden Sie alle momentan aktiven Windows-Anwendungen.
- Wenn Sie die Client Software auf einer Diskette bekommen haben, enthält diese zwei verschiedene Verzeichnisse für den 16 Bit und 32 Bit Treiber.

  Um den 16Bit Treiber zu installieren, führen Sie das `SETUP.EXE` Programm im Verzeichnis **SETUP16** auf der Diskette aus. Um den 32Bit Treiber zu installieren, führen Sie das `SETUP.EXE` Programm im Verzeichnis **SETUP32** auf der Diskette aus.
- Wenn Sie die Software in einem selbst extrahierenden Archiv bekommen haben, können Sie das entsprechende Softwarearchiv entpacken, indem Sie es ausführen. Das 16 Bit Archiv heißt `ODBC16.EXE` und das 32 Bit Archive heißt `ODBC32.EXE`.

  Beim Start des Entpackens der Archivdatei erscheint ein Windows Dialogfenster, das einerseits die Auswahl eines temporären Verzeichnises ermöglicht, in dem die Dateien entgepackt werden sollen, und andererseits die Option für einen automatischen Start des Setup-Programms.

  Klicken Sie auf die Schaltfläche `Unzip` um die Dateien aus dem Archive auszupacken und das SETUP Programm zu starten. Klicken Sie auf die Schaltfläche `Close` um die Installation abzubrechen.

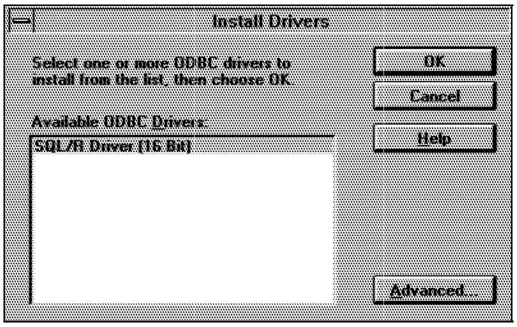
Nach dem Start des Setup-Programms erscheint das Installations-Dialogfeld auf dem Bildschirm.



odbca

Klicken Sie auf die Schaltfläche `Continue`, um die Installation fortzusetzen.

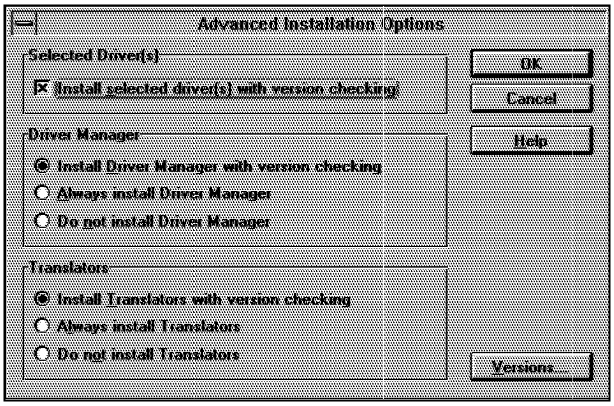
Das folgende Dialogfeld zeigt eine Liste der vorhandenen ODBC Treiber an, die vom `SETUP` Programm installiert werden können.



odbcb

Wählen Sie bitte den **SQL/R** Treiber aus und klicken Sie auf die Schaltfläche `OK`, um die Installation fortzusetzen.

Eine benutzerdefinierte Installation kann durch Anwahl von `Advanced` durchgeführt werden. Normalerweise ist es allerdings nicht notwendig, die Optionen der Grundeinstellung zu ändern.



odbcc

Folgende Installationsoptionen stehen zur Auswahl:

- `Selected Driver(s) (Ausgewählte Treiber)`

  Wenn die Prüfoption gesetzt ist, dann wird standardmäßig eine Versions-Überprüfung durchgeführt. Sie werden aufgefordert, die Installation des **SQL/R** Treibers zu bestätigen, sofern er dieselbe oder eine frühere Versionsnummer besitzt als die des bereits installierten Treibers.Sofern Sie den **SQL/R** ODBC Treiber zum ersten Mal installieren, hat diese Versionsüberprüfung keine Auswirkung. Ist die Versionüberprüfung nicht aktiviert, erfolgt keine Überpfüfung des zu installierenden Treibers, er wird unbeachtet seiner (ggf. veralteten) Versionsnummer installiert.

- `Driver Manager (Treiber Manager)`

  Diese Optionen gibt an, inwieweit der ODBC Treiber Manager installiert werden soll.

  - Installation des Treiber Managers mit Versionsüberprüfung
    Der ODBC Treiber Manager wird nur installiert, wenn er dieselbe oder eine neuere Versionsnummer als die des bereits existierenden Treiber Managers hat.
  - Installation des Treiber Managers ohne Überprüfung
    Der Treiber Manager wird unbeachtet seiner Version installiert.
  - Keine Installation des Treiber Manager bei Versionskonflikten
    Der Treiber Manager wird nicht installiert, sofern dieselbe oder eine neuere Version als die zu installierende bereits installiert ist.

- `Translator (Übersetzer)`

  Der **SQL/R** Treiber unterstützt keine Übersetzter. Sie können diese Optionen ignorieren.

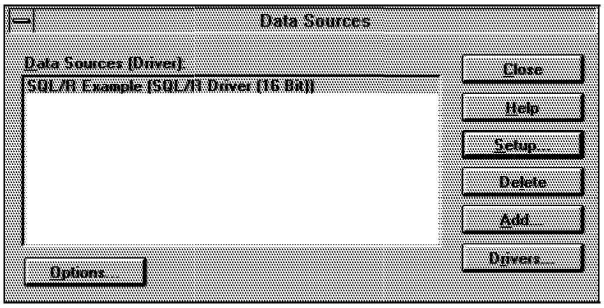
- `Versions (Versionen)`

  Klicken Sie diese Option an, um sich Informationen über die Versionen der verschiedenen ODBC Komponenten in einem Dialogfeld anzeigen zu lassen.

Schließen Sie das Dialogfeld durch Auswahl der Schaltfläche `OK` oder `Cancel`. Im Hauptmenü wählen Sie den **SQL/R** Treiber aus und klicken Sie auf die Schaltfläche `OK`, um die Installation fortzusetzen

Nachdem die Dateien kopiert worden sind, erscheint der ODBC Treiber Manager (Bestandteil von Microsofts ODBC) auf dem Bildschirm. Der Manager kann zum späteren Zeitpunkt aus der Windows Systemsteuerung aus gestartet werden.

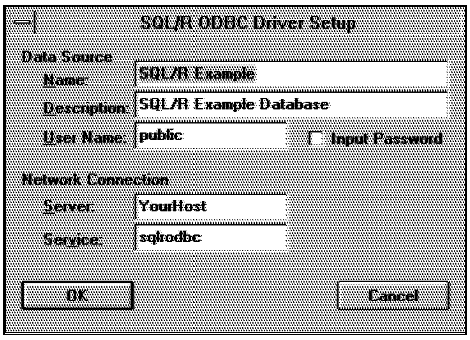
Eine vollständige Beschreibung über die Administration von ODBC Treibern und insbesonders über Datenquellen folgt am Ende dieses Kapitels im Abschnitt 3.4.



odbcd

Sie können nun eine neue Datenquelle definieren oder eine existierende Datenquelle ändern. Das Installationsprogramm installiert als Beispiel eine Datenquelle mit dem Namen "**SQL/R** Example", das die bei **SQL/R** mitgelieferte Beispieldatenbank benutzt.

Um die Beispiel-Datenquelle benutzen zu können, müssen Sie die Definition vervollständigen. Klicken Sie auf die Schaltfläche `Setup...`, um das **SQL/R** ODBC Treiber Setup Menü aufzurufen.



odbce

Die Hardcopy zeigt die Grundeinstellung der Datenquelle "**SQL/R** Example". Bitte tragen Sie den Name des Servers (Host) in das Feld "Server" ein.

Klicken Sie auf die Schaltfläche `OK`, um Ihren Eintrag abzuspeichern und in das vorherige Menü zurückzukehren.

Klicken Sie auf die Schaltfläche `Close`, um den Dialog zu schließen und und die Installation zu beenden.

3.2 Konfiguration des Client-Systems

Um eine TCP/IP Netzwerkverbindung aufzubauen, müssen Sie den Servernamen in die Datei `HOSTS` und den Servicenamen in die Datei `SERVICES` eintragen. Wo die beiden Dateien sich befinden, hängt von dem verwendeten TCP/IP Protokoll Stack ab. Wenn Sie den Microsoft TCP/IP Protokoll Stack benutzen, können Sie die Dateien in den folgenden Verzeichnissen finden:

MS Windows 3.x, WfW 3.11	`\WINDOWS`
MS Windows 95	`\WINDOWS`
MS Windows NT	`\WINNT\DRIVERS\ETC`

Bitte tragen Sie eine Zeile in die Datei `HOSTS` auf Ihrem PC ein, die folgenden Inhalt haben muß.

```
192.168.1.1          Server Name
```

Ersetzen Sie die *IP Adresse* und den *Server Namen* mit den für Ihr System gültigen Angaben. Fragen Sie bei Unklarheiten Ihren Systemadministrator nach dem Servernamen und der IP Adresse.

Bitte tragen Sie eine Zeile in die Datei `SERVICES` ein, die folgenden Inhalt haben muß.

```
sqlrodbc 8003/tcp      # {SQLR} ODBC
```

Klären Sie die Auswahl der Port Nummer mit Ihrem Systemadministrator ab.

Bitte beachten Sie, daß die Port Nummern des Clients und des Servers übereinstimmen müssen.

3.3 Treiber-Optionen

Zusätzliche Treiberoptionen können in der Datei `SQLRODBC.INI`, die sich in Ihrem Windows Verzeichnis befindet, konfiguriert werden. **Normalerweise ist es nicht notwendig, die Grundeinstellung des Treibers zu ändern.**

Die Datei `SQLRODBC.INI` wird nicht vom Setup-Programm installiert. Stattdessen wird eine Muster Konfigurationsdatei (`SQLRODBC.SAM`) in das Windows Verzeichnis kopiert. Entweder können Sie sich analog dazu eine eigene `SQLRODBC.INI` Datei erstellen oder die Muster Konfigurationsdatei `SQLRODBC.SAM` in `SQLRODBC.INI` umkopieren und mit einem Editor Ihrer Wahl (z.B. Notepad) modifizieren.

Die Konfigurationsdatei `SQLRODBC.INI` besteht aus mehreren Abschnitten. Jeder Abschnitt fängt mit einem Namen an, der in eckigen Klammern steht. Jeder Abschnitt kann eigene Konfigurationseinträge beinhalten. Der Konfigurationseintrag und sein entsprechender Wert sind durch ein Gleichheitszeichen (=) getrennt; Kommentare fangen mit einem Semikolon (`;`) an.

3.3.1 Der Abschnitt Config

Eintrag	**Bedeutung**
`Ctl3D`	In der Grundeinstellung verwendet **SQL/R** ODBC CTL3DV2 oder CTL3D32.DLL um eine dreidimensionale Darstellung von Dialogen zu erreichen. Diese Einstellung kann mit `Ctl3D=0` abgeschaltet werden. Die Grundeinstellung ist `Ctl3D=1`.
`UseKeepAlive`	Wenn Sie wolllen, daß **SQL/R** ODBC die TCP KEEPALIVE Option aktiviert, dann setzen Sie `UseKeepAlive=1`. Mit dieser Option kann **SQL/R** ODBC feststellen, ob die Verbindung zum Server aktiv ist, Die Grundeinstellung ist `UseKeepAlive=0`.

3.3.2 Der Abschnitt Debug

Entry	Meaning
`LogFile`	Sofern eine Datei `LogFile` definiert ist, werden darin vom **SQL/R** ODBC Treiber Debug Informationen protokolliert. Der Umfang und Art der Debug Informationen ist vom `LogMode` (siehe unten) abhängig. Wenn das `LogFile` nicht definiert ist, findet keine Protokollierung statt. In der Grundeinstellung ist das `LogFile` nicht definiert.
`LogMode`	Der `LogMode` setzt sich aus Flags zusammen, die eine Protokollierung für bestimmte Ausführungsarten und Gewichte erlauben. Das Logging-Level definiert, was und wieviel an Debug-Informationen protokolliert wird. Wenn kein `LogFile` definiert wurde, findet keine Protokollierung statt. In der Grundeinstellung ist das `LogFile` nicht definiert und der `LogMode="*0"`. Log-Flags: * Bezieht sich auf alle Flags A API Schnittstelle N Netzwork P Protokoll Log-Level: 0 Fehlermeldungen (Error Messages) 1 Informationen („Gut zu wissen") 2 Wichtige Debug-Informationen 3 Ausführliche Debug-Informationen

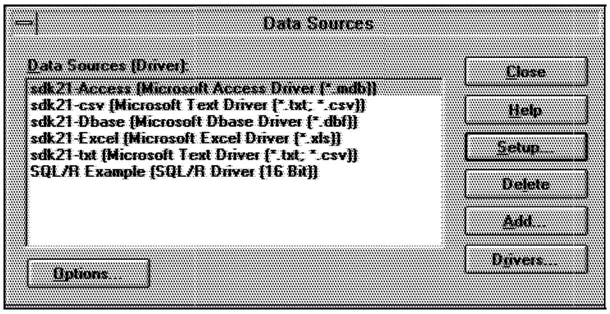
3.3.3 Die Musterdatei der Treiberkonfiguration

```
; sqlrodbc.ini

[Config]

; By default SQLRODBC uses the CTL3DV2 or CTL3D32 DLL to achieve a
; 3 dimensional appearance. You can disable the use of the CTL3D
; library by setting Ctl3D=0. The default is Ctl3D=1.
```

```
; Ctl3D=1

; If you want SQLRODBC to activate the TCP KEEPALIVE option to
; determine if the server is still alive and well, you can set
; UseKeepAlive=1. The default is UseKeepAlive=0.

; UseKeepAlive=0


[Debug]
; LogMode consists of log flags which enables logging for specific
; execution paths and a logging level, which determines what
; (actually how many :-) is logged.
; If the LogFile is undefined, no logging will be done.
; The default is LogFile undefined (no logging) and LogMode="*0".
;
; Log Flags:
;  * - Applies to all log flags
;  A - API
;  N - Network
;  P - Protocol
;
; Log Level:
;  0 = Error Messages
;  1 = Informational ("nice to know")
;  2 = Debug
;  3 = Verbose debug

; LogMode="*0"
; LogFile=c:\odbc.log
```

3.4 Administration der Datenquellen

Der ODBC Treiber Manager ist eine Microsoft ODBC-Komponente, die für die Verwaltung der ODBC Treiber und Datenquellen verantwortlich ist. Sofern man von einer Windows Anwendung eine Verbindung zu einer ODBC Datenquelle aufbauen möchte, wird der Treiber Manager aufgerufen, um den Zugriff auf die Datenquelle zu ermöglichen. Der ODBC Treiber Manager kann auch allein von der Systemsteuerung von Windows aus gestartet werden, um Datenquellen für einen Zugriff zu definieren.



adma

Das Hauptmenü des Treiber Managers zeigt eine Liste aller definierten Datenquellen mit den zugehörigen Treibern an.

Folgende Aktionen stehen zur Verfügung:

`Close` Verlassen des ODBC Treiber Managers.

`Help` Aufruf der Online Hilfe.

`Setup...` Konfiguration einer existierenden Datenquelle.

`Delete` Löschen einer existierenden Datenquelle. Wählen Sie die zu löschende Datenquelle aus der Liste aus und klicken Sie auf die Schaltfläche `Delete`.

`Add...` Hinzufügen einer neuen Datenquelle.

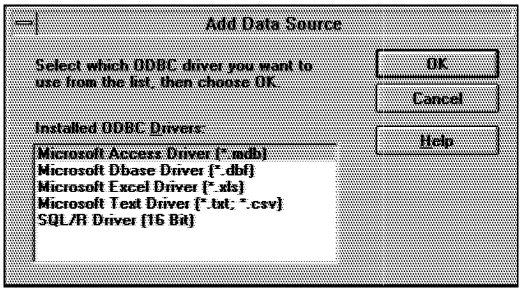
`Drivers...` Installation oder Löschen eines ODBC Treibers. Mit dieser Funktion kann auch die Version des ODBC Treibers abgefragt werden.

`Options...` Setzten von Treiber Manager Optionen.

3.4.1 Einfügen einer neuen Datenquelle

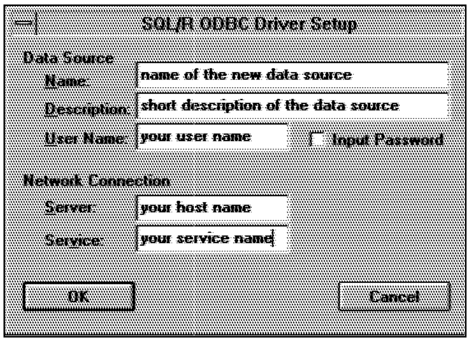
Um eine neue Datenquelle einfügen zu können, klicken Sie im Hauptmenü auf die Schaltfläche `Add...`. Der Treiber Manager öffnet dann ein Fenster mit einer Liste der vorhandenen Treiber.

Da jeder Treiber sein eigenes Setup verwendet, müssen Sie den Treiber, den Sie benutzen möchten, zuerst auswählen.



admc

Bitte wählen Sie den **SQL/R** ODBC Treiber aus der Liste aus und klicken Sie dann auf die Schaltfläche `OK`. Danach wird das Setup-Dialogfenster des **SQL/R** ODBC Treibers angezeigt.



admd

Um die neue SQL/R Datenquelle zu definieren, müssen Sie folgende Information eintragen:

- `Name`

  Tragen Sie den Namen der neuen Datenquelle ein. Dieser Name wird dann in der Liste der installierten Datenquellen erscheinen. Bitte beachten Sie, daß diese Datenquelle auch auf dem Server konfiguriert werden muß. **Der Name der Datenquelle muß auf dem Server und dem Client exakt gleich geschrieben werden**.

- `Description`

  Tragen Sie eine kurze Beschreibung der Datenquelle ein.

- `User Name`

  Tragen Sie Ihre Benutzerkennung ein. Dies wird vom **SQL/R** ODBC Server zur Überprüfung der Zugriffsberechtigung auf die entsprechende Datenquelle verwendet. Bitte beachten Sie, daß sich Ihre ODBC Benutzerkennung von Ihrer HP-UX Benutzerkennung (login) unterscheiden kann. Bitte fragen Sie im Zweifelsfall Ihren Systemadministrator.

- `Server`

  Tragen Sie den Host Namen (oder die IP Adresse) des Servers ein. Der Name des Servers muß in der Datei `HOSTS` definiert sein.

- `Service`

  Tragen Sie den Service Namen (oder die Port Nummer) ein, der vom Server benutzt wird. Als Grundeinstellung wird der Service Name `sqlrodbc` vorgegeben. Der Service Name muß in der Datei `SERVICES` definiert sein. Server und Client müssen dieselbe Portnummer (Service Port) benutzen.

- `Input Password`

  Wenn Sie diese Option aktivieren, fragt der **SQL/R** ODBC Treiber nach einem Zugangskennwort, um eine Verbindung mit dieser Datenquelle aufzubauen. Bitte beachten Sie, daß sich Ihr ODBC Zugangskennwort von Ihrem HP-UX Zugangskennwort (Paßwort) unterscheiden kann. Bitte fragen Sie Ihren Systemadministrator im Zweifelsfall nach Ihrem ODBC Zugangskennwort.

Klicken Sie auf die Schaltfläche [OK], um Ihre Eingaben abzuspeichern und zum Hauptmenü zurückzukehren. Klicken Sie auf die Schaltfläche [Cancel], um Ihre Eingaben zu verwerfen und zum Hauptmenü zurückzukehren.

Die neue Datenquelle erscheint nun in der Liste der installierten Datenquellen.

3.4.2 Konfiguration einer existierenden Datenquelle

Um die Konfiguration einer installierten Datenquelle zu modifizieren, wählen Sie die entsprechende Datenquelle aus der Liste und klicken Sie auf die Schaltfläche `Setup...`.

In einem neu geöffneten Fenster werden die Eigenschaften der Datenquelle angezeigt und Sie können die gewünschten Einstellungen der ausgewählte Datenquelle modifizieren. Eine Beschreibung des Setup-Menüs finden Sie im vorherigen Kapitel (3.4.1).

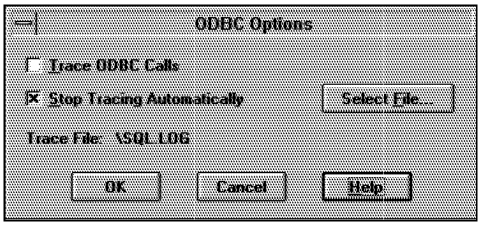
3.4.3 Treiber

Wenn Sie auf die Schaltfläche `Drivers...` im Hauptmenü klicken, erscheint ein Dialogfeld mit allen installierten ODBC Treibern.

Nun können Sie entweder einen neuen Treiber installieren, einen bestehenden Treiber löschen oder sich Informationen über einen installierten ODBC Treiber anzeigen lassen. Weiterführende Informationen hierzu erhalten Sie in der Online Hilfe.

3.4.4 ODBC Optionen

Wenn Sie im Hauptmenü auf die Schaltfläche `Drivers...` klicken, erscheint das Fenster "ODBC Options".



admb

Mit Hilfe dieses Dialogfeldes können alle ODBC Aufrufe des Treiber Managers protokolliert werden. Weiterführende Informationen hierzu erhalten Sie in der Online Hilfe.

4

Treiber Referenz

Dieses Kapitel enthält Referenzinformationen zum **SQL/R** ODBC Treiber und zum **SQL/R** ODBC Server.

Es besteht aus folgenden Abschnitten:

- Die Liste aller installierten Dateien
- Beschreibung der Repository Datei

4.1 Liste der Dateien

4.1.1 List der Dateien auf dem Server

Dieser Abschnitt enthält ein Verzeichnis aller Dateien die auf dem Server (HP9000) installiert werden und für **SQL/R** ODBC wichtig sind.

Datei/ Verzeichnis	Installiert unter	Beschreibung
sqlrodbcd	/opt/sqlr/bin	**SQL/R** ODBC Server
odbc.cfg	/opt/sqlr/etc	ODBC Server Konfigurationsdatei
odbc.dsn	/opt/sqlr/etc	Definition der Datenquellen
odbc.user	/opt/sqlr/etc	Benutzeridentifikation

Beispiele für die Konfigurationsdateien werden im Verzeichnis /opt/sqlr/newconfig/config installiert.

4.1.2 Liste der Dateien auf dem Client

In diesem Abschnitt finden Sie ein Verzeichnis aller Dateien, die auf dem Client System installiert werden. Das Installationsverzeichnis ist abhängig von der MS Windows Version.

Tabelle 1: Windows 3.x, WfW 3.11, Windows 95

Datei/ Verzeichnis	Installiert unter	Beschreibung
SQLRO16.DLL	\WINDOWS\SYSTEM	**SQL/R** ODBC 16 bit DLL
SQLRO32.DLL	\WINDOWS\SYSTEM	**SQL/R** ODBC 32 bit DLL
SQLRODBC.INI	\WINDOWS	Treiber Konfigurations Datei
SQLRODBC.SAM	\WINDOWS	Treiber Konfiguration Musterdatei

Tabelle 2: Windows NT

Datei/ Verzeichnis	Installiert unter	Beschreibung
SQLRO16.DLL	\WINNT\SYSTEM	**SQL/R** ODBC 16 bit DLL
SQLRO32.DLL	\WINNT\SYSTEM32	**SQL/R** ODBC 32 bit DLL
SQLRODBC.INI	\WINNT	Treiber Konfigurations Datei
SQLRODBC.SAM	\WINNT	Treiber Konfiguration Musterdatei

4.2 Die Repository Datei

Eine Repository Datei ist eine Textdatei die **SQL/R** ODBC Anweisungen enthält. **SQL/R** ODBC verwendet die Repository Datei um Zusatzinformationen zu einer Datenbank zu speichern.

Repository Dateien beziehen sich auf eine Datenquelle und werden bei jedem Öffnen der Datenquelle gelesen nachdem alle Datenbanken erfolgreich geöffnet werden konnten.

SQL/R Repository Dateien können verwendet werden für:

- Feldformatierung (`FIELD`)
- Alternative Feldnamen
- Virtuelle Felder
- Views (`CREATE VIEW`)
- Zugriffsrechte (`GRANT`/`REVOKE`)

Bitte beachten Sie: Das aktuelle **SQL/R** ODBC Release unterstützt nur die Verwendung von `FIELD` Statements in Repository Dateien. Eine Integration weiterer Funktionalität erfolgt in einem der folgenden Releases.

4.2.1 Glossar

Dieser Abschnitt dient der Erklärung bzw. Definition verschiedener, in diesem Abschnitt verwendeter Worte und Ausdrücke.

Feld (oder Item)
: Ein Feld ist die kleinste logische Einheit einer Datenbank. Seine Inhalte sind nicht auf ein Wort oder einen numerischen Eintrag beschränkt, es kann auch aus mehreren Wörtern bestehen.

: Zum Beispiel: Das Feld Strasse kann aus dem Eintrag "Martin Luther King Boulevard" bestehen. Felder entsprechen einer einzelnen Spalte einer Tabelle.

Array
: Ein Array ist eine Gruppe von Feldern des selben Typs (auch als Elemente bezeichnet), die mit dem selben Namen und einem Index gekennzeichnet werden.

SQL/R ODBC behandelt jedes Array Element wie ein eigenes Feld. Der Feldname wird aus dem Basisnamen sowie zwei Unterstrichen (__) und dem Index (beginnend mit 1) gebildet.

Zum Beispiel, wenn ein Array 12 Einträge füer den monatlichen Umsatz hat, wird der Monat Mai zu Umsatz__5 und der Monat Januar zu Umsatz__1.

Record (oder Datensatz)

Ein Record oder Datensatz ist eine Sammlung von Feldern. Jedes Feld in einem Record hat eine eindeutige Bezeichnung. Records werden in Tabellen gespeichert. Sie entsprechen einer Zeile einer Tabelle.

Tabelle (oder Dataset)

Eine Tabelle ist eine Sammlung von Records, die spaltenweise dargestellt werden.

View

Ein View ist eine virtuelle Tabelle, die aus ausgewählten Feldern einer oder mehrerer Tabellen besteht. Im einfachsten Fall besteht ein View aus einer einzigen Tabelle. Die `CREATE VIEW` Anweisung definiert einen permanenten Views. Die `SELECT` Anweisung erzeugt einen temporären View.

Korrelations Name

Ein Korrelations Name ist ein Alias Name oder ein Pseudonym einer Tabelle. Es wird verwendet um den Zugriff auf eine Tabelle zu vereinfachen oder um einen eindeutigen Tabellen-Namen zu definieren.

Instanz

In Fällen in denen eine einzelne Tabelle mehrfach in einem View verwendet wird (z.B. self join), gibt es mehrere Instanzen der selben Tabelle. Instanzen werden über unterschiedliche Korrelations Namen unterschieden.

Zum Beispiel:

```
SELECT A.PartNo, A.PartDesc, B.PartNo, B.PartDesc
FROM Parts A, Parts B
WHERE B.PartNo = A.PartReference;
```

Dies definiert einen temporären View, der zwei unterschiedliche Datensätze der Tabelle Parts enthält. Sie werden über die Korrelations Namen A und B unterschieden. A und B sind instanzen der Tabelle Parts.

Alias

Ein Alias ist ein Pseudonym (alternativer Name) für ein Feld in der Datenbank. Es wird mit einer `FIELD` Anweisung definiert.

4.2.2 Reservierte Worte

Reservierte Worte, sind durch **SQL/R** vordefinierte Worte mit einer speziellen Bedeutung.
Die Groß-/Kleinschreibung bleibt unberücksichtigt.

ABS	DAYOFYEAR	LEFT	REVOKE
ACOS	DEFAULT	LENGTH	RIGHT
ADD	DELETE	LIKE	RTRIM
ALL	DESC	LOCATE	SECOND
ALTER	DESCRIBE	LOG	SELECT
AND	DISTINCT	LTRIM	SET
ANY	DOUBLE	MAX	SIGN
AS	DROP	MICROSOFT	SIN
ASC	ESCAPE	MIN	SMALLINT
ASCII	EXISTS	MINUTE	SQRT
ASIN	EXP	MOD	SUBSTRING
ATAN	FIELD	MONTH	SUM
ATAN2	FLOAT	NOT	TABLE
AVG	FLOOR	NOW	TAN
BETWEEN	FN	NULL	TIME
BY	FOR	ODBC	TIMESTAMP
CASCADE	FOREIGN	OF	TO
CEIL	FROM	OJ	UCASE
CHAR	GRANT	ON	UNION
CHECK	GROUP	OPTION	UNIQUE
CONCAT	HAVING	OR	UPDATE
CONVERT	HOUR	ORDER	USER
COS	IFNULL	OUTER	VALUES
COT	IN	PI	VENDOR
COUNT	INDEX	PRIMARY	VIEW
CREATE	INSERT	PRODUCT	WEEK
CURDATE	INTEGER	PUBLIC	WHERE
CURRENT	INT	QUARTER	WITH
CURTIME	INTO	RAND	XOR
DATABASE	IS	REFERENCES	YEAR
DATE	JOIN	REPEAT	
DAYOFMONTH	KEY	REPLACE	
DAYOFWEEK	LCASE	RESTRICT	

4.2.3 Syntax Übersicht

Die Repository Datei ist eine Textdatei und kann mit einem Texteditior wie z.B. `vi` bearbeitet werden.

Hierbei gelten folgende Regeln:

- Leerzeilen und Zeilen, die mit einem Gatterzeichen (#) beginnen werden ignoriert.
- Eine Anweisung kann über mehrere Zeilen gehen und wird mit einem Semikolon abgeschlossen.
- Das Semikolon **muß** das letzte Zeichen in der Zeile sein. Lediglich Leerzeichen sind danach erlaubt.
- Bei Schlüsselwörtern und Feldnamen wird die Groß-/Kleinschreibung nicht berücksichtigt.

4.2.4 Konstanten

SQL/R unterstützt die folgenden Konstanten:

- **Zeichenkette**
  Zeichenketten werden in einfache oder doppelte Anführungszeichen eingeschlossen. Eine Zeichenkette muß mit dem gleichen Typ von Anführungszeichen enden, mit dem sie begonnen wurde. Um ein Anführungszeichen in eine Zeichenkette einzufügen, muß das Anführungszeichen mit einem Backslash (\) versehen werden.

  Zum Beispiel:

  ```
  "Zeichenkette"
  'Zeichenkette mit \' Anführungszeichen'
  ```

- **Numerische Werte**
  Es werden Integer (Ganzzahl) und Floating Point (Fließkommazahlen) unterstützt. Jeder Wert der außerhalb des Integer Gültigkeitsbereiches liegt ($2^{31} - 1$) oder einen Dezimalpunkt oder einen Exponent enthält werden zu einer Fließkommazahl konvertiert.

  Zum Beispiel:

```
123
123.45
3.1415E10
```

- **Datums Werte**
  SQL/R ODBC unterstützt die Microsoft ODBC Erweiterungen. Eine Datumskonstante muß wie folgt angegeben werden:

  ```
  {D "JJJJ-MM-TT"}
  ```

  Hierbei ist JJJJ eine vierstellige Jahreszahl (ein zweistelliger Wert wird relativ zu 1900 betrachtet), MM ist ein ein- oder zweistelliger Monat und TT ist ein ein- oder zweistelliger Tag.

  Zum Beispiel:

  ```
  {D "1996-08-30"}
  ```

- **Zeit Wert**
  SQL/R ODBC unterstützt die Microsoft ODBC Erweiterungen. Eine Zeit Konstante muß wie folgt angegeben werden:

  ```
  {T "HH:MM:SS"}
  ```

  Hierbei ist HH ein ein- oder zweistelliger Wert für die Stunde und, MM ist ein ein- oder zweistelliger Wert für die Minuten und SS ist ein ein- oder zweistelliger Wert für die Sekunden.

  Zum Beispiel:

  ```
  {T "14:30:50"}
  ```

- **Timestamp (Zeitpunkt)**
  SQL/R ODBC unterstützt die Microsoft ODBC Erweiterungen. Eine Timestamp Konstante muß wie folgt angegeben werden:

  ```
  {TS "JJJJ-MM-TT HH:MM:SS"}
  ```

  For example:

  ```
  {TS "1996-08-30 14:30:50"}
  ```

Alle Datums-, Zeit- und Timestamp Werte werden intern in Sekunden berechnet.

4.2.5 Bezeichner

Ein Bezeichner besteht aus Buchstaben, Ziffern oder einem Unterstrich (_). Ein Bezeichner muß immer mit einem Buchstaben beginnen. Bezeichner können wahlweise groß- oder kleingeschrieben werden. Deshalb werden die Bezeichner „Name", „NAME", „name" als gleich angesehen.

Bezeichner könne für alle Ausdrücke wie zum Beispiel Tabellen und Feldnamen verwendet wrden. Es darf nur kein von **SQL/R** reserviertes Schlüsselwort verwendet werden.

Bezeichner die einem reservierten Wort entsprechen, müssen in senkrechte Stiche (|) eingeschlossen werden. Darum wird zum Beispiel „|time|" als Bezeichner interpretiert und nicht als Schlüsselwort.

SQL/R ODBC behandelt jedes Element eines Arrays wie unterschiedliche Felder. Der Feldname wird aus dem Basisnamen sowie zwei Unterstrichen (_) und dem Index (beginnend mit 1) gebildet.

Aus einem Array 12 Einträgen füer den monatlichen Umsatz wird der Monat Mai zu Umsatz__5 und der Monat Januar zu Umsatz__1.

4.2.6 Die FIELD Anweisung

Die Field Anweisung wird verwendet um feldspezifische Eigenschaften wie Format oder Konvertierung festzulegen.

`FIELD` [*Bez.*.]*Tabelle*.*Feld*
`CONVERT TO` *Daten Typ*
[`DESCRIBE AS` "*Kommentar*"] **;**

Hierbei ist *Bez.* ein optionaler Bezeichner (wie zum Beispiel ein Alias Name der Datenbank), *Tabelle* ist der Name einer Tabelle und *Feld* ist ein Feld Name. *Kommentar* ist eine Zeichenkette mit einer Feldbeschreibung.

Bitte beachten Sie: Der Feld Name **muß** immer qualifiziert werden (zum Beispiel Kunden.Nummer). Unqualifizierte Feldnamen sind reserviert zur Verwendung als „virtuelle Felder". Die Feldbeschreibung wird zur Zeit nicht verwendet.

Folgende *Datentyp*en werden unterstützt:

`CHAR` [`(`*Breite*`)`]	Zeichenkette
`SMALLINT` [`(`*Breite*`)`]	Ganzzahl
`INTEGER` [`(`*Breite*`)`]	Ganzzahl
`FLOAT` [`(`*Breite* [`,` *Genauigkeit*]`)`]	Fließkommazahl
`DOUBLE` [`(`*Breite* [`,` *Genauigkeit*]`)`]	Fließkommazahl
`DATE` [`("`*Format*`"` [`,` *Jahr*]`)`]	Konvertierung eines numerischen Wertes in einen Datumswert.
`TIME` [`("`*Format*`")`]	Konvertierung eines numerischen Wertes in eine a Zeit Wert.
`TIMESTAMP`	Konvertierung eines numerischen Wertes in einen Zeitpunkt Wert

Das optionale Argument *Breite* definiert eine abweichende Spaltenbreite. Das optionale Argument *Genauigkeit* definiert eine abweichende Genauigkeit (wir zur Zeit nicht unterstützt). Das Argument *Format* definiert eine Datums- oder Zeitkonvertierung. Das optionale Argument *Jahr* legt eine Basisjahr fest.

Für weitere Informationen zu *Datentypen* verweisen wir auf die ODBC Referenz von Microsoft.

Datums Konvertierungsformate sind:

`"GREG"`	Konvertiert ein numerischens Datum (Anzahl Tage seit 1.1. des Basisjahrs). Sofern das optionale Argument *Jahr* nicht gesetzt wird, wird 1972 dafür eingesetzt.
`"YMD"`	Konvertiert das Datum aus dem Format [JJ]JJMMTT.

Zeit Konvertierungsformate sind:

`"HM"`	Konvertiert aus HHMM
`"HMS"`	Konvertiert aus HHMMSS

Zum Beispiel

```
FIELD Auftrag.Auftrnr CONVERT TO CHAR(8);
FIELD Auftrag.Auftragsdatum CONVERT TO DATE("YMD");
FIELD Zeileneintrag.Lieferdatum CONVERT TO DATE("YMD");
```

A

HP Eloquence Format-Nummern

HP Eloquence Format-Nummern werden entweder von `schema` oder `dbmods` für die Datenbank definiert. Bein Öffnen der Datenbank übersetzt **SQL/R** diese Format-Nummern in entsprechende Formate.

Die HP Eloquence Format-Nummer ist ein *additiver* Schlüssel. Für jede Gruppe bzw. Eigenschaft wird ein Wert addiert.

Group	Value	Comments
Query Write inhibit		
No write inhibit (default)	0	(ignored)
Write inhibit	1	(ignored)
Item type		
Date type	2	`DATE`("GREG",1972)
Currency	4	(ignored)
Undefined	6	(ignored)
Spacing		
Default	0	(ignored)
Comma every 3 digits	8	(ignored)
Post decimals		
Default	0	(ignored)
FIXED 0	16	(1)
FIXED 1	32	(1.0)
FIXED 2	48	(1.00)
FIXED 4	80	(1.000)
FIXED 3	64	(1.0000)
FIXED 5	96	(1.00000)
FIXED 6	112	(1.000000)

Bei den Typen `MONEY` und `DATE`, werden alle weiteren Einträge ignoriert.

Die Anzahl Nachkommastellen (post decimals) werden nur für Fließkommazahlen (float, double) berücksichtigt.

For example:

Datum = 2
Geldbetrag = 4
Wert mit 2 Nachkommastellen = 48

Index