

Datendirektzugriffe – Teil 1

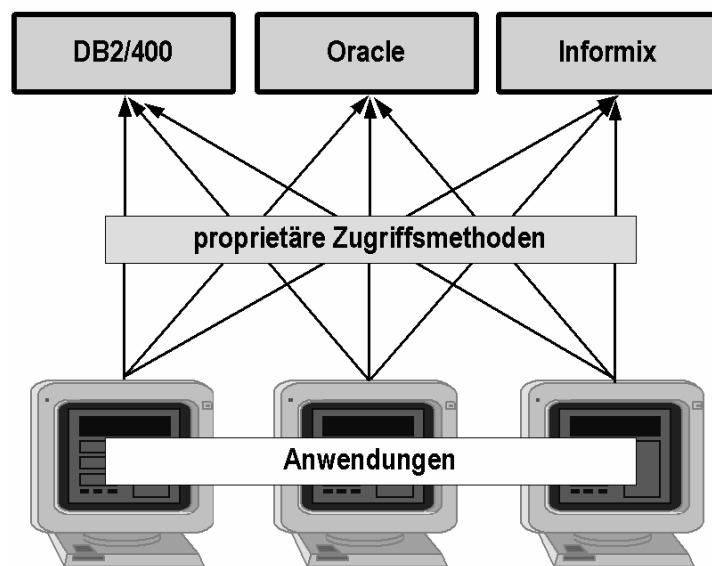
Eine Art, iSeries-Daten in Excel und anderen Office-Programmen zu verarbeiten, besteht darin, die zu bearbeitenden Daten nicht via Dateitransfer zum Beispiel in ein Arbeitsblatt hineinzustellen, sondern auf die betreffenden Daten direkt zuzugreifen. Es werden also keine lokalen Datenkopien der DB2-Daten via Filetransfer erzeugt, sondern die Daten werden dort belassen, wo sie eigentlich auch am besten aufgehoben sind – auf dem System i Server.

Access für Windows unterstützt in diesem Umfeld folgende Zugriffsverfahren

- ODBC
- OLE DB
- ADO.NET

ODBC

ODBC ist ein maßgeblich von Microsoft entwickeltes Standardverfahren, um Datenbankzugriffe zu vereinheitlichen. Als Anwendungen entwickelndes Unternehmen war Microsoft schon frühzeitig durch den Umstand betroffen, dass seine Software mit diversen Datenbankmanagementsystemen (DBMS) koexistieren musste. Da aber jedes der verfügbaren DBMS seine eigenen individuellen Zugriffsschnittstellen bereitstellte, musste ein Anwendungsentwickler für jedes der betreffenden DBMS eigene Zugriffsmodule bereitstellen.

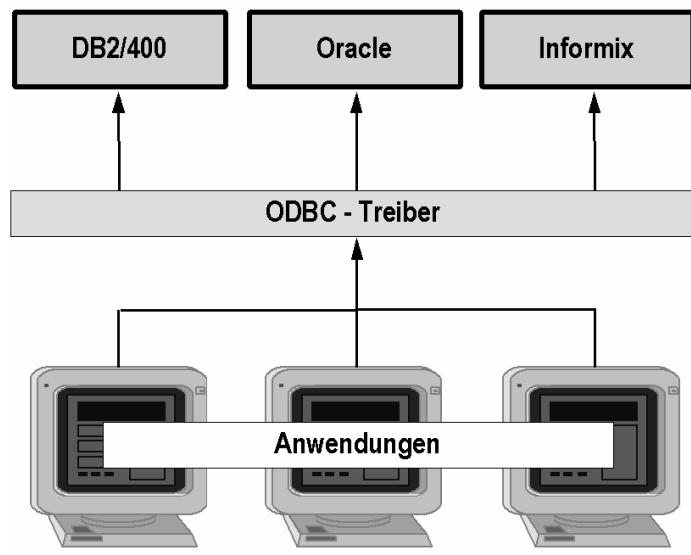


Die historische Situation

Dieses führt im Rahmen der Softwareentwicklung zu quasi redundanten Entwicklungen, denen nicht unbedingt ein entsprechender Return on

Investment gegenübersteht. Microsoft war diese Situation schon frühzeitig klar und bat darum alle namhaften DB-Hersteller, gemeinsam mit Microsoft ein standardisiertes Zugriffsverfahren zu entwickeln.

IBM trat diesem Gremium relativ spät bei, entwickelte aber mit dem Produkt Client Access für DB2/400 seine ersten ODBC-Treiber.



Der ODBC-Treiber

Basis des ODBC-Standards ist der Sachverhalt, dass alle relationalen DB-Systeme Datenbankzugriffe via SQL realisieren.

Aufgabe des ODBC-Treibers ist es, einen SQL-Befehl einer Clientanwendung an den ODBC-Treiber zu übergeben, der dafür Sorge trägt, dass der Befehl an die Zugangsschnittstelle des jeweiligen DBMS herantransportiert wird. Kurz gesagt, der Input für einen ODBC-Treiber ist der SQL-Befehl eines Clients in Form einer Zeichenkette, der Output ist ein validierter, aufbereiteter SQL-Befehl, der von dem betreffenden DBMS korrekt verarbeitet werden kann.

Diese Verarbeitungslogik bietet dann der Anwendungsseite eine eindeutige, standardisierte Schnittstelle. Jedoch muss jeder DB-Hersteller für sein DB-System den passenden ODBC-Treiber bereitstellen.

Mit Access für Windows steht Ihnen ein performanter, solider ODBC-Treiber zur Verfügung, der es Ihnen erlaubt, aus ODBC-fähigen Anwendungen heraus auf DB2 UDB zuzugreifen.

Technische Merkmale der Version V5R4 und aufwärts

Seit Version V5R4 des Access für Windows werden folgende ODBC-Merkmale unterstützt:

Verfügbar für 32- und 64-Bit Windows; kompatibel zur Microsoft ODBC Version 3 und somit ausgestattet mit den Funktionen der aktuellen Version 3.5.

Beachten Sie: 32-Bit-Anwendungen können auch nur mit einem 32-Bit ODBC-Treiber kommunizieren. Auch 64-Bit-Anwendungen können lediglich mit 64-Bit ODBC arbeiten.

Folgende Verfahrensmerkmale wurden seit V5R2 eingeführt:

V5R4 /V6R1/V7R1	V5R3	V5R2
Feldnamenlänge 128 Byte	UTF8 und UTF16	Maximale Befehslänge 64 k
Maximale Befehslänge 2 GB	Binäre und Varbinäre Datentypen	Zusätzliche Descriptor- Informationen
Lock Sharing zwischen Transaktionen	Höhere Werte für die Anzahl der Nachkommastellen	MTS Unterstützung
Verbesserter Support für "zeichenbegrenzte " Namen	Verbesserter MTS Support	
	Core Level 1999 SQL wird unterstützt	

ODBC-Konfiguration: System i

Bevor Sie jedoch ODBC nutzen können, müssen Sie einige vorbereitende Konfigurationsschritte ausführen.

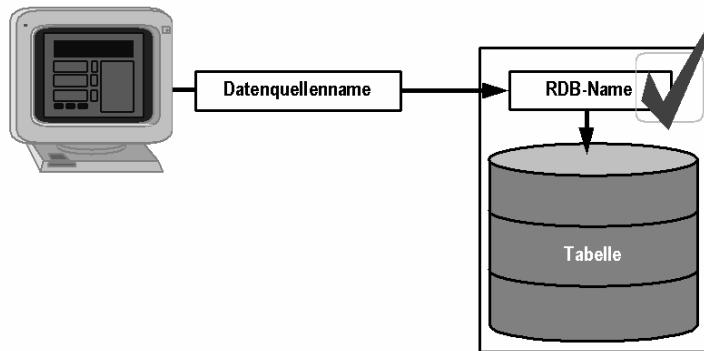
Der erste Teil der Konfiguration gehört zur DB-Konfiguration des System i.

Der Zugriff zu einer DB2–Datei führt über folgenden Pfad:

DB-Name.Bibliothek.Datei

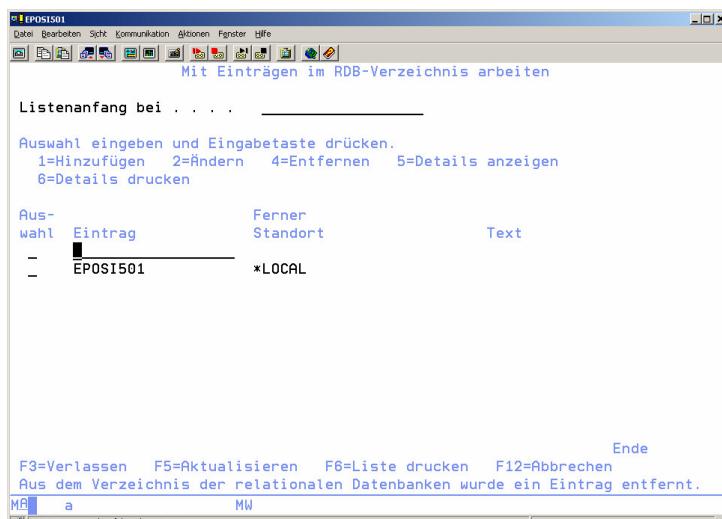
Da es sich bei ODBC um eine SQL-Zugriffsmethode handelt, wird grundsätzlich immer die erste Teildatei einer Datei verarbeitet.

Wird ein System i Server neu ausgeliefert, so wird für DB2 der von IBM konfigurierte Systemname als Datenbankname verwendet. Es hat sich als sinnvoll herausgestellt, dass ein selbstvergebener Name die Arbeit mit ODBC fördert. Dieser Name muss in der Datenbank erzeugt werden, um einen problemfreien ODBC-Betrieb zu ermöglichen.



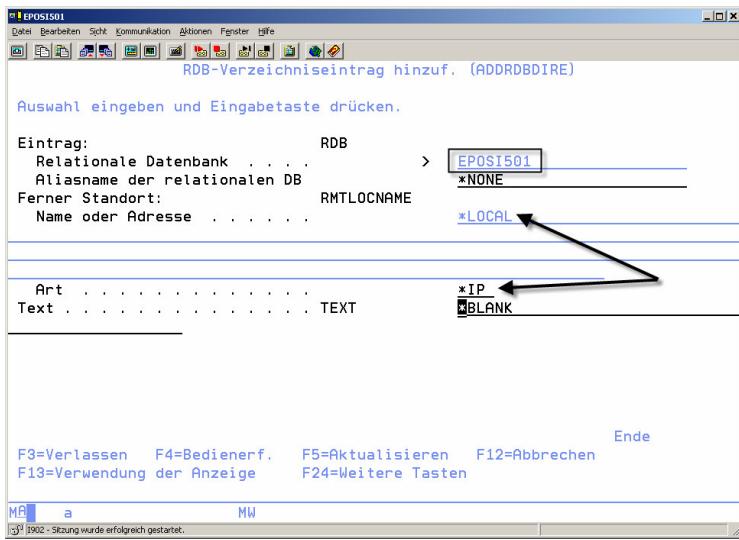
Der RDB-Name

Auf dem System i wird der Datenbankname unter dem Begriff RDB-Name (Relational Database Name) geführt. Um DB2 UDB einen eigenen RDB-Namen zu geben, müssen Sie den Befehl WRKRDBDI (Work with Relational Database Directory Entries) aufrufen.



WRKRDBDI (Work with Relational Database Directory Entries)

Innerhalb dieses Befehls führen Sie die Funktion 1=“Hinzufügen“ aus und erzeugen einen RDB-Namen.



1= Hinzufügen

Im oben dargestellten Bildschirm sehen Sie den Namen (RDB) EPOSI501. Dieses ist der RDB-Name der Datenbank des Systems.

Als Fernen Standort tragen Sie den Sonderwert *LOCAL ein.

Im Parameter Art tragen Sie *IP ein.

Sollte ein RDB-Name nicht vergeben worden sein (auch das ist möglich!), so werden Sie später beim Zugriff aus Excel heraus unter Umständen die Meldung "TOKEN * ungültig" sehen.

RDB-Namenskonflikte

Soll bei einem bestehenden System der System i RDB-Name geändert werden, so sind je nach verwendeter Client-Anwendung Adressierungsprobleme zu erwarten. Gerade MS Excel ist hier eine vergleichsweise kritische Anwendung.

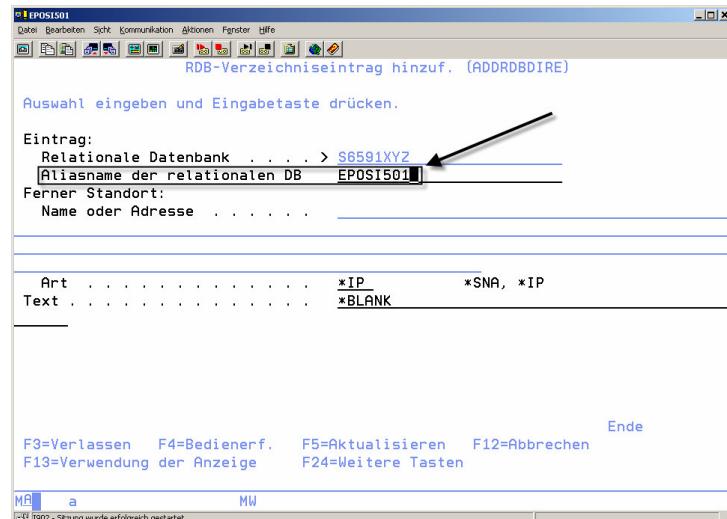
Gehen wir von folgendem Beispiel aus. Ihr System wurde von V5R2 oder V5R3 nach i5/OS V5R4 migriert. Im alten Betriebssystem wurde mit dem Standard-RDB-Namen S6591XYZ gearbeitet. Um sogenannten Powerusern das Konfigurieren einer ODBC-Datenquelle zu erleichtern, soll der RDB-Name auf EPOSDATEN geändert werden.

Jedoch gibt es schon eine ganze Reihe von bestehenden Excel-Arbeitsblättern, die mit dem alten RDB-Namen arbeiten. Diese Arbeitsblätter würden nicht mehr funktionieren, sollte der System i RDB-Name geändert werden.

Eine Möglichkeit, das Problem zu lösen, bestände darin, alle bereits erstellten Arbeitsblätter zu ändern. Sicherlich keine wirklich gute Lösung.

Eine bessere Lösung ist es, den alten RDB-Namen zu belassen und den neuen Namen als Alias sozusagen darüberzulegen.

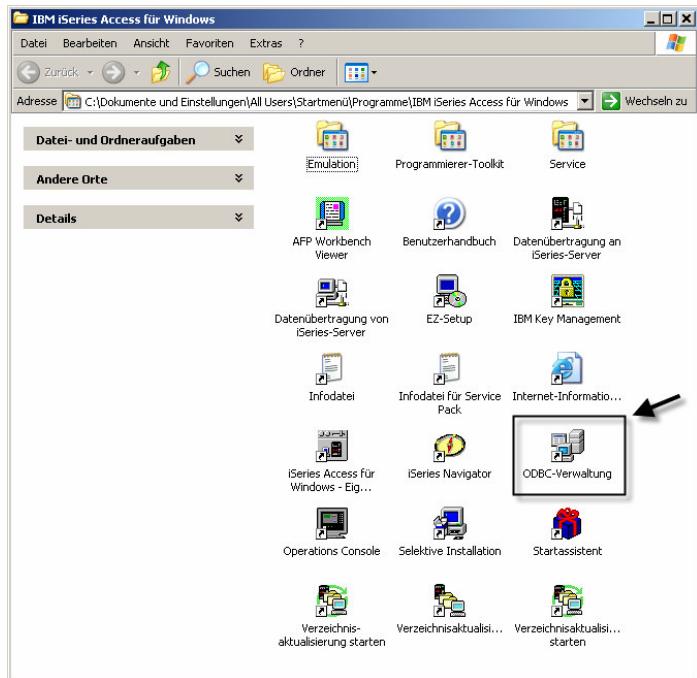
Hier der entsprechende Hinzufügen-Dialog:



1= Hinzufügen mit Alias

Nachdem Sie diese vorbereitenden Arbeiten erledigt haben, wenden Sie sich der Konfiguration auf der Seite des Windows PCs zu.

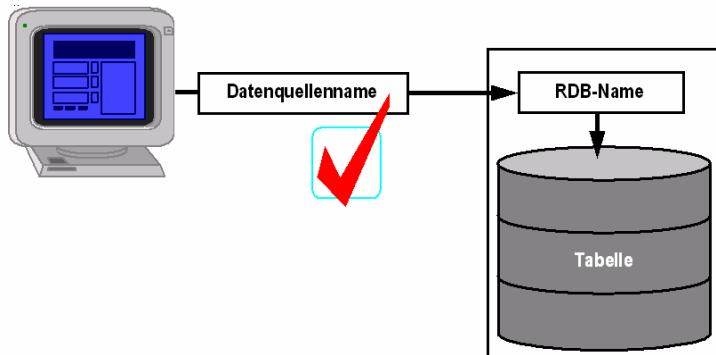
ODBC – Konfiguration



ODBC-Verwaltung

Die Konfiguration erfolgt aus der Access für Windows-Programmgruppe heraus. Klicken Sie hier doppelt auf das Symbol ODBC-Verwaltung.

Ziel der Konfiguration ist die Erstellung einer Datenquelle – abgekürzt DSN (Data Source Name). Die Datenquelle ist ein logischer Verbindungsname zu einer oder mehreren Bibliotheken.

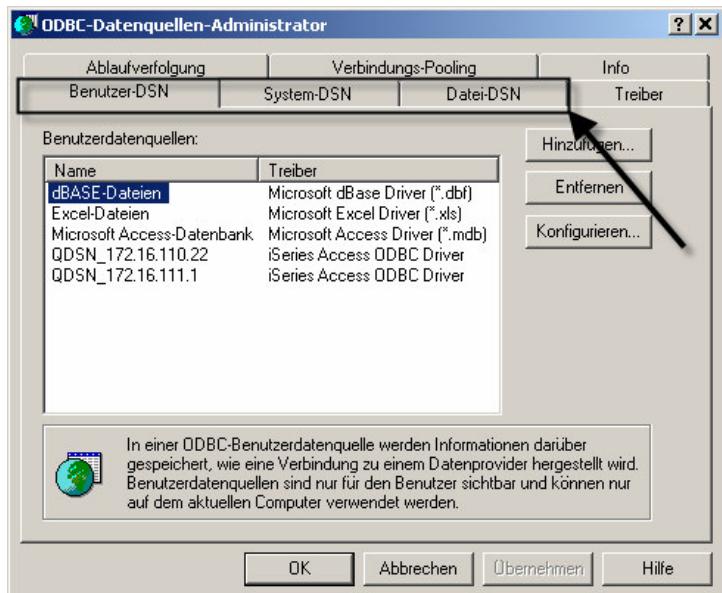


Der Datenquellename

Die Datenquelle stellt die Verbindung zur Datenbank her, die die Verwaltungseinheit für alle Bibliotheken darstellt.

Der Datenquellen-Administrator

Nach Öffnen des Symbols ODBC-Verwaltung sehen Sie folgenden Dialog.



Der Datenquellen-Administrator

Die Datenquellen-Arten

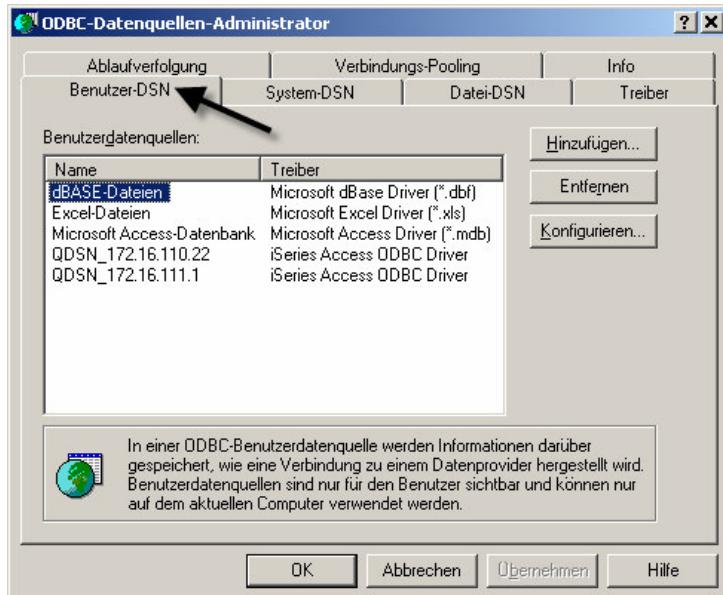
Anhand der Registerreiter können Sie erkennen, dass es drei verschiedene Arten von Datenquellen (DSN) gibt:

- Benutzer-DSN
- System-DSN
- Datei-DSN

Der Nutzen der verschiedenen Datenquellen erschließt sich, wenn man mit einem Windows-Betriebssystem arbeitet, das Objektzugriffsberechtigungen unterscheidet, wie Windows NT, Windows 2000 oder Windows XP.

Die Betriebssysteme Windows 95, 98 und ME unterscheiden keine Objektzugriffsberechtigungen. Unterhalb dieser Betriebssysteme ist die Wirkung von Benutzer- und System-DSN identisch.

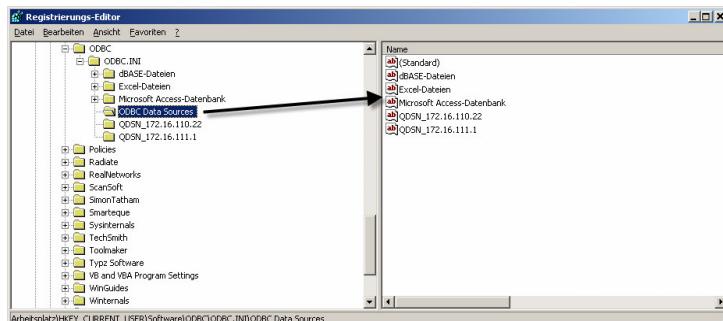
Benutzer-Datenquellen



Die Benutzer DSN

Die Benutzerdatenquelle steht nur für den Anwender zur Verfügung, der sie erstellt hat. Legt also der Benutzer Luttkus auf seinem Windows-PC eine Benutzer-DSN an, so kann ein Benutzer Hartmann, der sich am PC von Luttkus mit seinem eigenen Konto anmeldet nicht mit der Benutzer-DSN von Luttkus arbeiten.

Technisch wird diese Restriktion dadurch bewirkt, dass alle Konfigurationsdaten der Benutzer-DSN im Zweig `Hkey_Current_User → Software → ODBC` der Registry des Windows-PCs gespeichert werden.



Ausschnitt aus der Windows Registry

Vergleichen Sie einmal dieses Bild mit dem vorherigen. Sie sehen die offensichtlichen Übereinstimmungen

Jeder Benutzer eines Windows-PCs mit einem eigenen Benutzerkonto besitzt einen eigenen Zweig in der Registry, mit dem er über private Datenquellen verfügen kann.

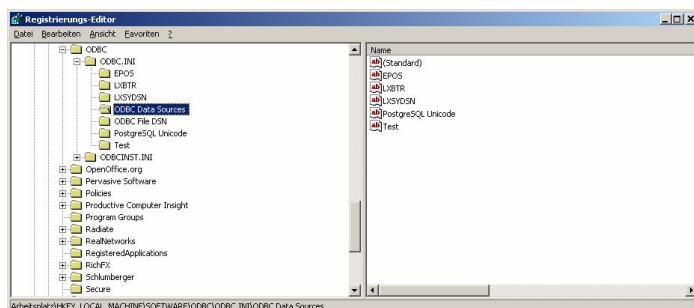
System-Datenquellen

Die Systemdatenquelle ist eine von jedem Benutzer eines Windows-PCs erreichbare Datenquelle. Hier wieder ein Beispiel: Der Benutzer Luttkus erzeugt eine Systemdatenquelle. Meldet sich jetzt der Benutzer Hartmann am Windows-PC von Luttkus an, so kann er ebenfalls mit der von Luttkus angelegten Systemdatenquelle arbeiten.



Die System-DSN

Die Konfigurationsdaten der System-DSN werden im Registry-Zweig Hkey_Local_Maschine → Software → ODBC abgespeichert.



Hkey_Local_Machine

Auch hier erkennen Sie die Übereinstimmungen zwischen diesem und dem vorherigen Bild.

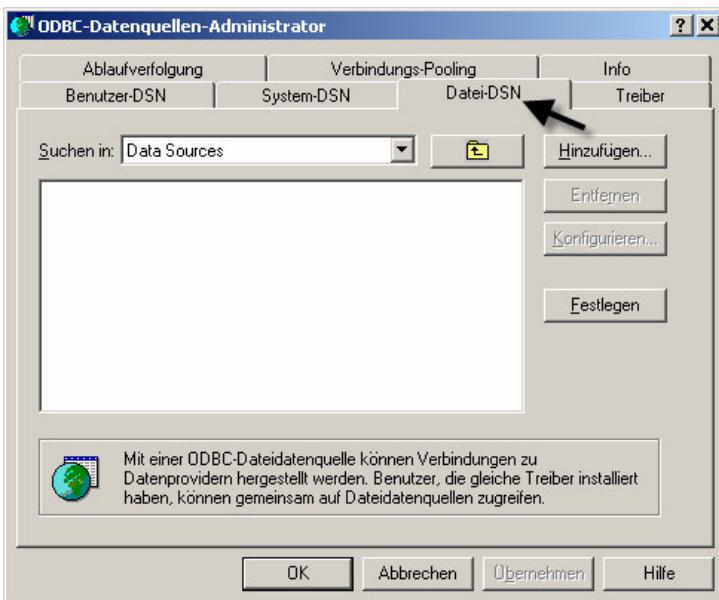
In diesem Zweig der Windows Registry werden für das Windows globale Konfigurationsdaten abgelegt.

Beide Datenquellenarten, Benutzer- und System-DSN, haben ein Merkmal gemeinsam. Sie werden in der Windows Registry gespeichert und sind damit an das lokale System gebunden. Sie können also nicht ohne tiefere

Systemeingriffe auf anderen Systemen genutzt werden. Aus diesem Grund werden sie auch Computer-Datenquellen genannt. Dieser Begriff ist deshalb wichtig, weil beim Ansprechen der Datenquellen der Dialog „Externe Daten von Microsoft-Anwendungen“ nach Benutzerdatenquellen einerseits und Computerdatenquellen andererseits fragt.

Datei-Datenquellen

Die Konfigurationsdaten einer Datei-DSN werden in Form einer Datei abgespeichert.

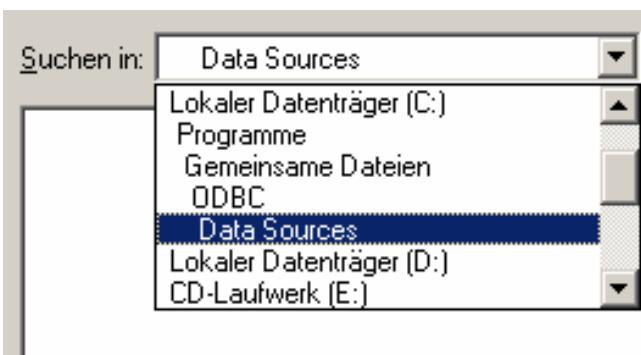


Die Datei-DSN

Deshalb können Datei-DSN auch im Netzwerk verteilt werden und mithilfe von Objektrechten vor unberechtigten Zugriffen geschützt werden.

Wichtig: Kopieren Sie eine Dateidatenquelle auf einen fernen Arbeitsplatz, so muss auf diesem bereits eine ODBC-Verwaltung existieren. Es reicht nicht, nur die Datei-DSN zu übertragen.

Sie sehen im Datei-DSN-Dialog den Knopf „Festlegen“. Dieser Knopf ist für Anwender von Microsoft Office-Produkten sehr wichtig.

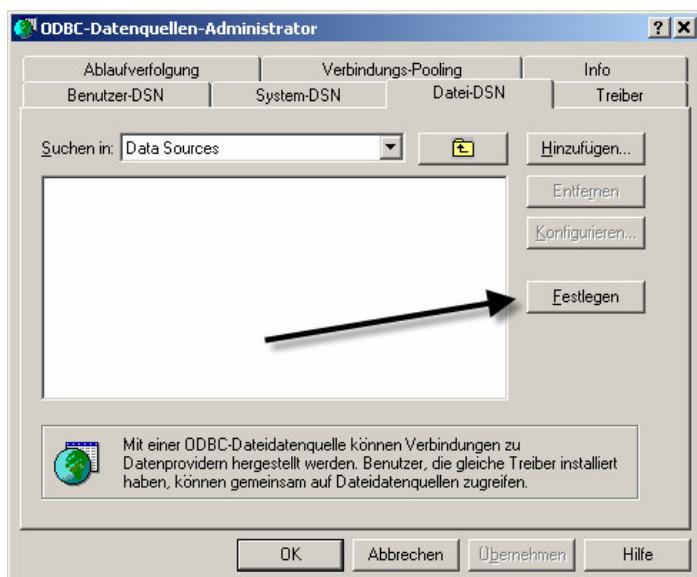


Der Datenquellen-Zugriffspfad

Der Standardpfad, in dem die ODBC-Verwaltung Datei-Datenquellen ablegt, ist

LW:\Programme\Gemeinsame Dateien\ODBC\Data Sources

LW bezeichnet das Windows-Systeminstallationslaufwerk.

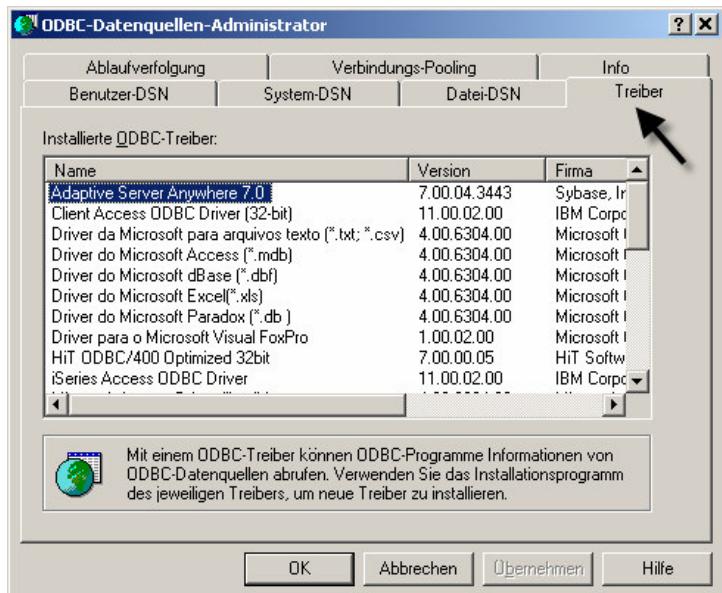


Der „Festlegen“-Knopf

Drücken Sie auf den Knopf „Festlegen“, so wird den Microsoft Office Produkten der Zugriffspfad zu den Datei-DSN mitgeteilt.

Das Register Treiber

Im Register Treiber sehen Sie die auf diesem System zur Verfügung stehenden ODBC-Treiber.

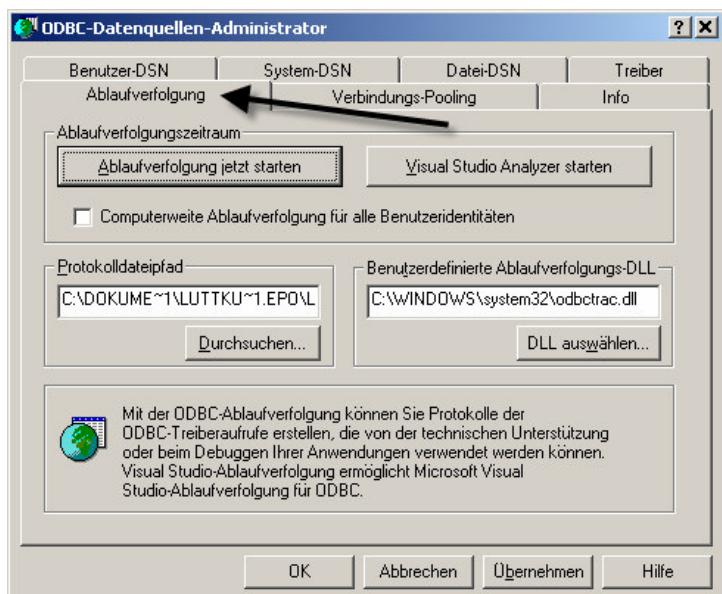


Das Register Treiber

Die in diesem Bild gezeigten ODBC-Treiber wurden durch die Komplettinstallation des Microsoft Office, durch die Installation des HiT ODBC-Treibers und durch Access für Windows auf das System gebracht. Es ist durchaus wahrscheinlich, dass diese Liste auf Ihrem Windows-PC anders aussieht.

Das Register Ablaufverfolgung

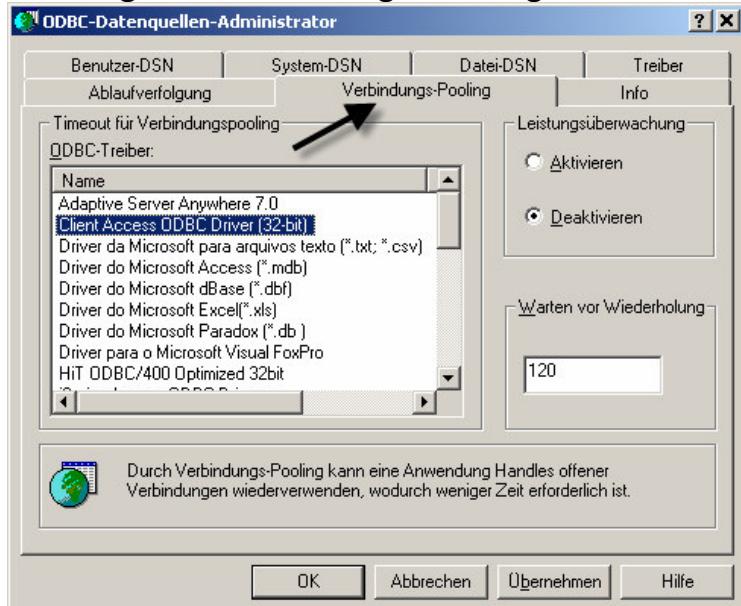
Ergeben sich bei dem Betrieb von ODBC Fehler, so können Sie Protokolle für den Zugriff erstellen, indem Sie im Register Ablaufverfolgung die Ablaufverfolgung starten.



Das Register Ablaufverfolgung

Das Ablaufverfolgungsprotokoll wird unter dem Namen SQL.LOG gespeichert. Sie können hier aber auch einen eigenen Dateinamen vergeben.

Das Register Verbindungs-Pooling



Register Verbindungs-Pooling

Mit Hilfe des Registers Verbindungs-Pooling haben Sie einen direkten Einfluss auf die ODBC-Performance Ihres Arbeitsplatzes. Was verbirgt sich hinter diesem Begriff? Beginnt ein ODBC-Durchgriff auf eine Datenbank, so wird zuerst ein SQL-Connect-Befehl ausgeführt, alle weiteren Tätigkeiten entwickeln sich sodann auf der Basis dieser Verbindung. Wird die SQL-Kommunikation beendet, so wird ein SQL-Disconnect-Befehl ausgeführt und damit endet die gesamte SQL-bezogene Kommunikation. Soll innerhalb einer gegebenen Zeit die Kommunikation wieder aufgenommen werden, so muss der soeben beschriebene Vorgang komplett erneut initiiert werden. Dadurch entsteht ein nicht unerheblicher Zeitbedarf.

Durch einen Doppelklick auf den iSeries Access ODBC-Treiber Namen wird folgender Dialog geöffnet:



Treiber zum Pool hinzufügen

Soll dieser Zeitbedarf für wiederholte Kommunikations-Einleitungen minimiert werden, so aktivieren Sie den Parameter „Verbindungen zu diesem Treiber zusammenlegen“.

Jetzt sieht die SQL-Kommunikation wie folgt aus:

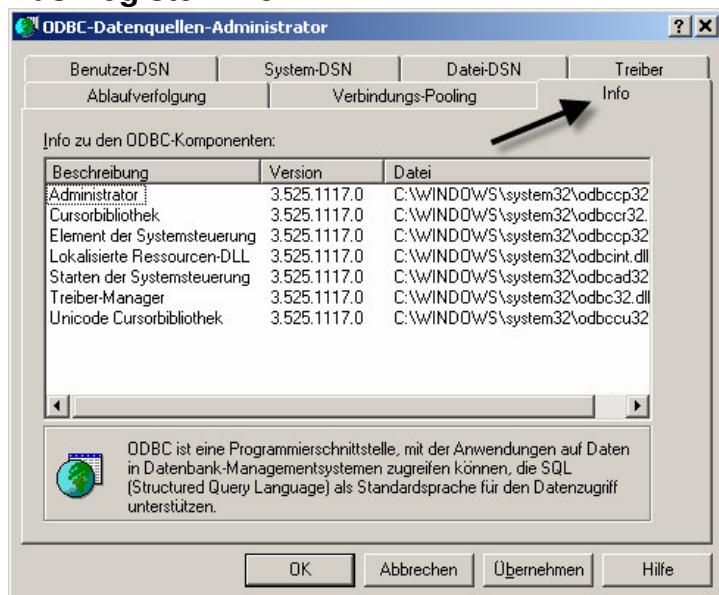
- 1.) Der SQL-Connect stellt Verbindung her.
- 2.) Die Anwendung arbeitet mit dieser Verbindung.
- 3.) Der SQL-Disconnect wird ausgeführt.

Jetzt wird die aufgegebene Verbindung in den Pool gestellt. Die Verbindung wird zwar von der Anwendung getrennt, jedoch bleibt eine „geparkte“ Verbindung zur angesprochenen DB bestehen, solange wie im Parameter Zeitspanne ... eingegeben. Versucht eine Anwendung, vor Ablauf der Wartezeit einen erneuten SQL-Connect abzusetzen, so wird zuerst im Verbindungspool nachgesehen, ob eine geparkte Verbindung besteht. Falls ja, wird die gepoolte Verbindung zur Datenkommunikation herangezogen.

Der Effekt: Eine gepoolte Verbindung wieder aufzunehmen, verursacht deutlich weniger Arbeitslast, verglichen mit dem Neuaufbau einer SQL-Kommunikation. Ergo: Verbindungen werden schneller aufgebaut.

Ist jedoch die Zeit des Parameters Zeitspanne ... abgelaufen, die dem Pool für einen Treiber zugeordnet wurde, so muss die SQL-Verbindung komplett neu aufgebaut werden.

Das Register Info



Register Info

Im Register Info werden Ihnen die Objekte des ODBC-Treibers, die Versionsstände und Speicherorte auf der Platte Ihres PCs angezeigt.

Lesen Sie weiter im nächsten TKL...