

Einleitung

In dieser Dokumentation wird speziell auf die Programmierung mit XProfan und dem Microsoft SQL-Server eingegangen. XProfan unterstützt eine Vielzahl von Funktionen zur Programmierung für Datenbankabfragen. In dem Thema SQL-Server wird ausschließlich auf die Programmierung über die ODBC-Schnittstelle eingegangen. Andere Funktionen wie DBase, Firebird etc. können Sie über die Hilfe von XProfan oder im Internet nachschlagen und ist hier kein Thema.

Warum SQL-Server?

Abgespeicherte Verfahrensweisen

Der einzige wesentliche Vorteil bei der Verwendung von SQL Servern ist der Gebrauch von abgespeicherten Verfahrensweisen. Abgespeicherte Verfahrensweisen sind Code-Zeilen, die von der Anwendung aufgerufen werden. Sie sind auf dem Server abgelegt und bereits für schnellere Reaktionszeiten zusammengestellt. Abgespeicherte Verfahrensweisen erfordern Kenntnis von der SQL Server-Syntax, genannt T-SQL (SQL-Transaktion). Die Verwendung von abgespeicherten Verfahrensweisen zentralisiert auch den Code, so dass das Problem schlechter Datenbank-Abfragen von einem Datenbank-Administrator beobachtet und gelöst werden kann.

Skalierbarkeit

Der Begriff „Skalierbarkeit“ wird verwendet, um die Wachstumsfähigkeit eines Unternehmens, das größer wird, zu beschreiben. Wenn Unternehmen schnell wachsen, kann eine kleine Datenbank-Anwendung wie Access zu einem Engpass für eine Webseite oder eine Desktop-Software werden. Ein Microsoft SQL Server ist für große und kleine Unternehmen schnell, so dass im Falle eines wachsenden Unternehmens der SQL Server die neuen Volumen der Datenbank Anfragen verarbeiten kann. Ein SQL Server kann Millionen von Datensätzen und Transaktionen verarbeiten.

Sicherheit

Sicherheit ist ein wichtiges Thema für jede Webseite. Bei einem SQL Server kann der Administrator den Zugang für die Nutzer gewähren oder verweigern. Der SQL Server hat eine bestimmte Sektion innerhalb der Anwendung, unter der Benutzerberechtigungen hinzugefügt werden können. Ein SQL Server ermöglicht es Administratoren, festzulegen, auf welche Tabellen und abgespeicherten Verfahrensweisen die Benutzer zugreifen können und welche diese abfragen können. Dadurch wird limitiert, welche Unterlagen und Benutzerinformationen abgefragt werden können, was die Kundendaten des Unternehmens schützt.

Transaktionsprotokolle

Transaktionsprotokolle sind Objekte auf dem SQL Server, die Abrufe aufzeichnen, Datensätze aktualisieren und löschen. Es gibt zwei Gründe, Transaktionsprotokolle aufzubewahren. Der erste ist für das „Rollback-Verfahren“. Dieser Prozess wird für versehentliche Änderungen oder Löschungen verwendet. Der Administrator kann Datensätze mit Hilfe von Transaktionsprotokollen auf die ursprünglichen Daten zurücksetzen. Zweitens können die Transaktionsprotokolle für Zwecke der Gefahrenabwehr verwendet werden. Wenn der Administrator eine Verletzung der Sicherheit vermutet, kann er die Transaktionsprotokolle für alle Arten der Datenbeschaffung beobachten und die Schwere des Schadens identifizieren.

Automatisches Backup

Ein SQL Server verfügt über eine automatische Backup-Option. Der SQL Server speichert automatisch eine Kopie der Datenbank und der Transaktionsprotokolle auf einer anderen Festplatte oder einem anderen Medium, wie einer CD-ROM oder DVD. Kleine Anwendungen wie Access verfügen nicht über diese Option, und solche Backups sind ein integraler Bestandteil bei der Zurückgewinnung während eines Desasters. Ein SQL Server hat auch Verfahren, welche es dem Administrator erlauben, eine Datenbank schnell wieder herzustellen, wenn Daten verloren gingen oder beschädigt wurden, oder der Server einen Crash der Festplatte hatte. Auch können Daten in Clouds gespeichert werden, deren zurück Gewinnung allerdings Zeit in Anspruch nimmt. Das „Umschalten“ auf eine zweite Platte, mit der gesicherten Datenbank dauert hingegen nur Sekunden.

Warum XProfan?

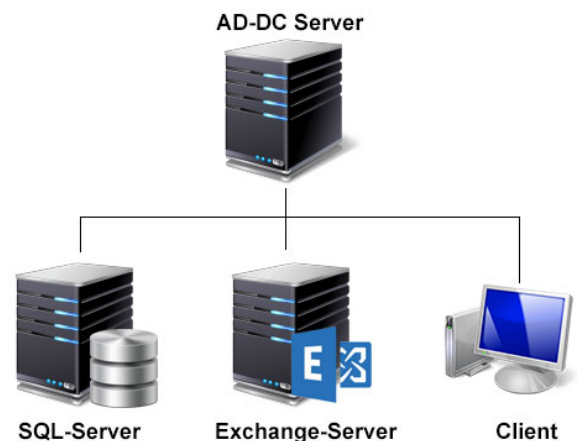
Ich klaue mal einfach eine Aussage aus einer Werbung: Weil einfach, einfach einfach ist...

Wenn ich an frühere Zeiten denke, mußte man schon eine Menge Knowhow mitbringen, um mit Programmen überhaupt mal einen Datensatz rauszukitzeln. Um z.B. mit Interbase-Datenbanken zu kommunizieren, war die erste Wahl Delphi. Da es von demselben Hersteller (Borland) kam, konnte man mit Hilfe von ADO direkt auf Datenbanken zugreifen etc... Allerdings mußte man eine Menge an Quelltexten erstellen, und brachte auch Probleme bei wachsenden Datenbanken mit.

Bei XProfan hingegen benötigt man hingegen nur wenige Zeilen Code um eine Datenbank via ODBC abzufragen. Es ist einfach und übersichtlich. Auch kann man direkt aus XProfan die Transaktionskontrolle übernehmen, was wir beim SQL-Server aber nicht benötigen, da man in der Eigenschaften der Datenbank genau einstellen kann, wie die Daten zu handhaben sind. Die Sicherheitsschichten ggf. die Transaktion kann automatisch anpasst werden.

Aufbau der Umgebung für die Beispiele in dieser Dokumentation

Für die Beispiele in dieser Dokumentation benutze ich einen Windows Server 2012 R2, der als Primärer Domänencontroller (AD DC) eingerichtet ist. Zusätzlich ist ein zweiter, eigenständiger SQL-Server (mit SQL-Server 2014) im Netzwerk. Für die Programmierumgebung nutze ich einen einfachen Client. (Im Moment mit Windows 7). Zusätzlich ist ein eigenständiger Exchange-Server ins Netzwerk integriert, um im späteren Verlauf näher einzugehen, wie man auch einen Exchange an eine SQL-Datenbank integriert. Für den Anfang reichen aber erst mal ein SQL-Server und der Client. Diese können auch auf einen einzelnen Rechner installiert werden, um den Beispielen zu folgen.



Jetzt ist die oben genannte Hard- und Software etwas für eine gehobene Infrastruktur. Für kleinere Firmen reicht natürlich auch Windows Server 2012 R2 Essentials sowie ein SQL-Server Express. (max. bis 10GB Daten pro Datenbank) Dieses reicht auch für unsere Beispiele völlig aus. Für den produktiven Einsatz empfehle ich dringend den Domänen Controller und den SQL-Server zu trennen.

Virtualisieren

Bei einem Server 2012 R2 Standard oder Datacenter kann man den Server auch mittels Hyper-V virtualisieren. Um alle Beispiele nur zum Testen, kann man sich auch von der Microsoft Internetseite einen Windows 2012 R2 Server herunterladen und z.B. in Oracle VM VirtualBox installieren. Ich trage mit Absicht keine Links hier in die Dokumentation ein, da es sich sowieso permanent ändert. Es gibt nur einen Standardlink, um alle Programme zu laden: <http://www.google.de>

Windows Server 2012 R2

Bei Microsoft kann man die Evaluierungsversionen für 180 Tage kostenfrei herunterladen. Diese 180 Tage kann man bis zu 240 Tage ganz legal verlängern. Einfach die Konsole als Administrator öffnen und "`slmgr.vbs -rearm`" eingeben und bestätigen. Danach ein Neustart und gut ist. Diesen kann man zum Testen einfach in VirtualBox installieren. Je nach Hardware ist das allerdings wie Zähne ziehen, da dieser nicht wirklich schnell ist. Wer aber noch wenig oder gar keine Erfahrung mit Windows Server 2012 R2 gemacht hat, dem Empfehle ich dieses zu tun, da man einfach mit „rumspielen“ kann. Auch hat man die Möglichkeit, das geänderte einfach wieder „zurückzusetzen“ sollte man wirklich Mist gebaut haben.

Installation von SQL-Server 2014

Die Installation prüft vorab welche Komponenten noch benötigt werden. Auf den Datenbankserver können Sie aber vorab bei der Ersteinrichtung die Rolle .NET Framework 3.5-Funktionen installieren. Generell sollten auch alle Updates für den Server schon installiert sein.

Features

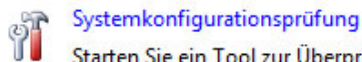
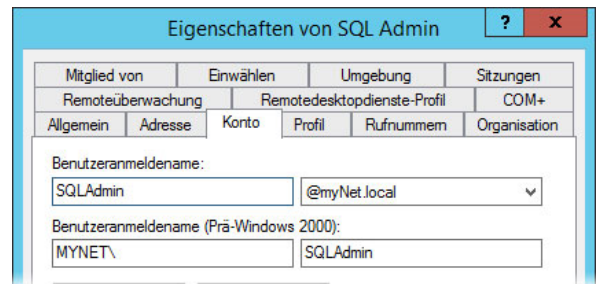


Sollte beim Hinzufügen von .NET Framework 3.5 der Fehler kommen:

! Müssen Sie einen alternativen Quellpfad angeben? Für mindestens eine

So klickt man unten in dem Fenster auf: „Alternativen Quellpfad angeben“. Legen Sie die Windows Server 2012 R2 Installations-DVD in das Laufwerk und geben Sie den Pfad: <DVD>\sources\sxs Beispiel: E:\sources\sxs ein. Nun wird das Framework installiert.

Als erstes erstellt man am Domänencontroller einfach vorläufig einen neues Benutzerkonto für die Installation (Hier im Beispiel einfach den Benutzer: SQLAdmin) und gibt ihm Administratorrechte. Vorläufig konfiguriere ich die Gruppenrichtlinien noch gar nicht. Darum kümmern wir uns später. Auf dem SQL-Server melden wir uns auch als dieser Benutzer an. Jetzt können wir mit der eigentlichen Installation beginnen. Nach dem einlegen der Installations-DVD erscheint das SQL-Server Installations Center. Dort prüfen wir erst einmal, ob auch alle Bedingungen erfüllt sind, um SQL-Server zu installieren. Hierzu klicken wir einfach unter dem Hauptpunkt Planen auf die Systemkonfigurationsprüfung.



Systemkonfigurationsprüfung

Starten Sie ein Tool zur Überprüfung des Systems auf Bedingungen, die eine erfolgreiche Installation von SQL Server verhindern könnten.

Sollten hier Warnungen oder gar Fehler angezeigt werden, sollten Sie die einzelnen Schritte abarbeiten. Sollte alles in Ordnung sein, können wir mit der Installation beginnen. Links im Menü auf Installation klicken und aus dem Menü den Eintrag Neue eigenständige SQL Server-Installation.... wählen.



Neue eigenständige SQL Server-Installation oder Hinzufügen von Funktionen zu einer vorhandenen Installation

Starten Sie einen Assistenten, mit dem Sie SQL Server 2014 in einer Umgebung ohne Cluster installieren oder einer vorhandenen SQL Server 2014-Instanz Funktionen hinzufügen können.

Jetzt kann man wählen, ob man die Testversion (Evaluation) installieren möchte, oder die SQL Server Express. Hat man hingegen einen Product-Key so kann man ihn eintragen und mit Weiter kommen wir auch schon zu den Lizenzbestimmungen, die man gleich ausdrucken kann, und gleich in die Rechtsabteilung der Unternehmens per E-Mail schickt. Nachdem das Okay aus der Rechtsabteilung kommt, klicken wir natürlich auf:

Ich akzeptiere die Lizenzbedingungen.

Ob Sie jetzt beim Programm zu Verbesserung teilnehmen möchten, und der Firma Microsoft Daten schicken wollen, überlasse ich jedem selbst. Mit Weiter werden jetzt die Globalen Regeln nochmals geprüft, was wir ja schon am Anfang der Installation gemacht haben. Wenn das Ergebnis keine Fehlermeldungen und Warnungen ausgibt, springt das Programm gleich zur Frage, ob Windows-Update auch für SQL-Server Updates bereitstellen soll. Hier empfehle ich ganz klar, daß Häkchen zu setzen. Bei Weiter werden die Setup-Installationsregeln geprüft. Hier kommt Standardgemäß die Warnung ob die Ports für SQL-Server geöffnet sind. Diese Warnung kann erst einmal ignoriert werden. Entweder vergibt man die Ports einzeln (z.B. 1433 für das Datenbankmodul) oder gibt nach der Installation das Programm allgemein frei, um die Ports dynamisch zu verwalten. Darauf kommen wir nochmals zurück, sobald die Installation beendet ist.

Funktionsauswahl

Für unsere Beispiele benötigen wir eigentlich nur die SQL-Database Engine. Wer nur die minimalste Installation haben möchte, den empfehle ich die Komponenten:

Instanzfunktionen:

- Database Engine Services

Freigegebene Funktionen:

- Konnektivität der Clienttools
- Dokumentationskomponenten
- Verwaltungstools Vollständig

Damit hat man das minimalste an Diensten die beim Serverstart geladen werden. In diesem Beispiel hier, werde ich aber mal alle „Alle Funktionen mit Standardwerten“ installieren. Es gibt noch so schöne Sachen wie Reportig-Service die ich ganz gerne einsetze, sowie die Sicherung der Datenbanken u.v.m.

Instanzkonfiguration

Bei der Erstinstallation gibt es ja noch keinerlei Instanz einer Datenbank. Für das Beispiel, behalte ich einfach mal den Namen MSSQLSERVER. Hier kann man auch einen anderen Namen gewählt werden. Dieser sollte aber so einfach wie möglich gehalten werden, da auch Verzeichnisse mit den Namen angelegt werden. Mit weiter, werden die Dienste angezeigt, und wie diese beim Serverstart geladen werden. Je mehr Funktionen ausgewählt worden sind, desto mehr Dienste laufen natürlich auch. Wer aber einen leistungsstarken Server hat, braucht sich nicht wirklich Gedanken drüber machen.

Datenbankmodulkonfiguration

Unter dem Reiter Serverkonfiguration empfehle ich einen gemischten Modus, da zusätzlich zu den Benutzern von Windows auch der „Superuser“ SA angelegt wird. Dieser Benutzer ist quasi der root für den SQL-Server. Dieser benötigt natürlich auch ein eigenes Passwort. Jetzt kommt auch der Grund, warum wir uns mit SQLAdmin an den Server angemeldet haben. Mit der Schaltfläche „Aktuellen Benutzer hinzufügen“ setzen wir fest, daß der User SQLAdmin der Admin vom SQL-Server wird. Sollte aus irgendeinem Grund der Benutzer SQLAdmin gesperrt, gelöscht oder getötet worden sein, so haben wir noch Zugriff über den Benutzer SA.

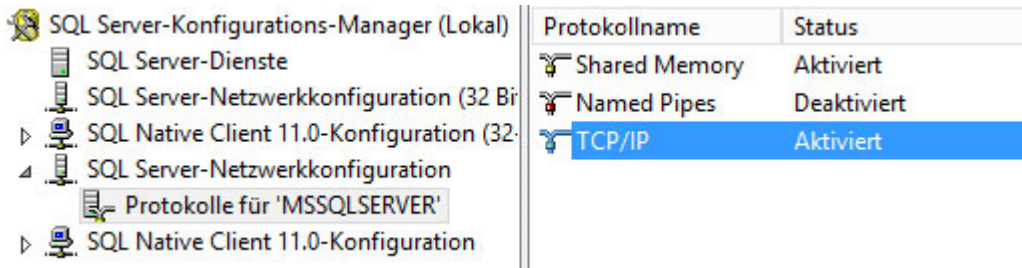
Unter dem Reiter Datenverzeichnisse kommt es jetzt natürlich darauf an, auf welcher Festplatte die Daten gespeichert werden. In vielen Konfigurationen wird auf einer Platte die Daten gespeichert, und die Sicherung auf einer anderen Platte. Hier kommt es auf die Strategie an, welches vom Systemadministrator festgelegt worden ist. Beispielsweise sind die Daten auf einer Festplatte, die im RAID-Verbund gespiegelt werden. Den Reiter FILESTREAM wird jetzt mal ganz gekonnt ignoriert, da ich darüber ein eigenes Buch schreiben könnte...

Jetzt kommen, je nach Funktionsauswahl, alle anderen Funktionsinstallationen, wo ich jedem Dienst, den Benutzer SQLAdmin hinzufüge, und die Verzeichnisse konfiguriere. Im Internet gibt es eine Menge Videos und Dokumentationen zu jedem einzelnen Punkt einer SQL-Server Installation. Das würde hier sonst den Rahmen sprengen, und ist auch nicht Thema dieser Dokumentation. Ganz am Ende, wird nochmals aufgelistet, was alles installiert wird, und dann kann man eigentlich die Installation auch schon starten. Nach der Installation empfehle ich den Server neu zu starten, und nach Updates zu suchen.

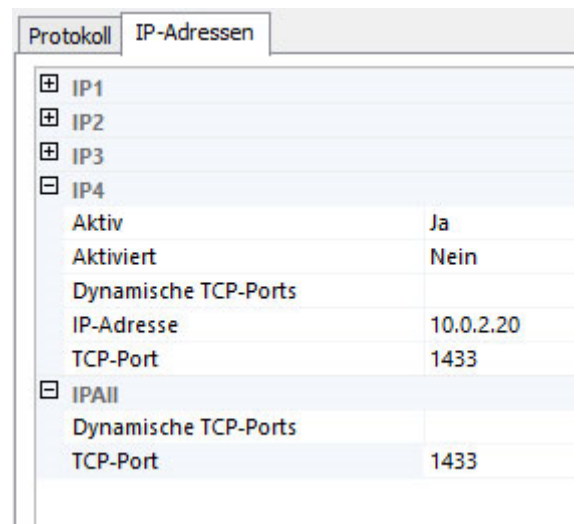
SQL-Server konfigurieren

Damit die Clients auf den SQL-Server zugreifen können, müssen nur zwei kleine Konfigurationen gemacht werden. Als erstes startet man den SQL SERVER KONFIGURATIONS MANAGER.

Wechseln Sie in die SQL-Server-Netzwerkkonfiguration / Protokolle für [SQLSERVERNAME]
Vergewissern Sie sich, das der Eintrag TCP/IP aktiviert ist, und doppelklicken Sie darauf.



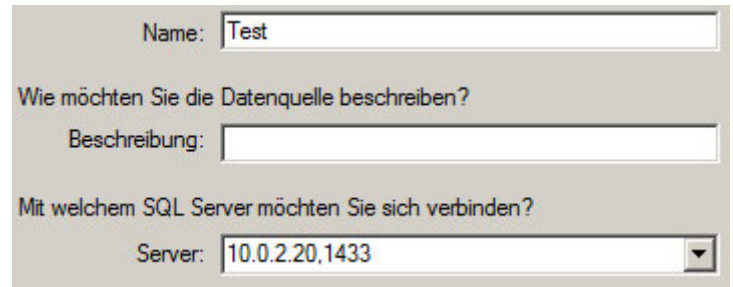
Geben Sie unter dem Menüpunkt IP4 die IP-Adresse des Servers ein. Dieses kann mit dem Konsolenbefehl `ipconfig` eingesehen werden. Generell haben alle Server bei mir in der Domäne eine feste IP-Adresse. Nur die Clients werden via DHCP versorgt. Unter TCP-Port wird Standardgemäß der Port 1433 verwendet. Diesen können Sie natürlich ändern. Vergewissern Sie sich, daß der Port und die IP-Adresse unter dem Reiter IP-Adresse überall gleich ist. Nach einer Änderung muß der SQL-Server neu gestartet werden. Wechseln Sie dazu auf SQL Server-Dienste, und klicken mit der rechten Maustaste auf den SQL-Server und wählen im PopUp-Menü die Option „Neu starten“ aus. Erst jetzt sind die Änderungen wirksam. Damit der SQL Server auch angesprochen werden kann, muß der vorher festgelegt Port auch in der Firewall freigegeben werden. In dem Beispiel zeige ich nur die Windows-Eigene-Firewall. Die Vorgehensweise kann aber auch auf andere Anbieter abstrahiert werden. Starten Sie die Windows-Firewall (am schnellsten mit Ausführen / `wf.msc`) Gehen Sie im linken Menü auf „Eingehende Regel“ und wählen im rechten Menü den Eintrag „Neue Regel...“ Wählen sie im Regeltyp die Option „Port“. Unter Protokolle und Ports geben Sie unter bestimmte lokale Ports den Port für den SQL-Server ein, den sie im Konfigurations-Manager festgelegt haben. (In unserem Beispiel 1433) Da es ein TCP-Port war, erübricht sich die Frage, ob wir TCP oder einen UDP Port benutzen wollen. Natürlich wollen wir, das die Clients einen Zugriff auf den SQL-Server bekommen, und wählen unter Aktion auch die Option „Verbindung zulassen“ aus. Da ich nur möchte, das Domänenbenutzer nur Zugriff auf die Datenbank haben, wähle ich unter Profil nur die Domäne aus. Als letztes gebe ich noch einen sinnvollen Namen ein, unter dem die Regel gespeichert wird. Und das war es auch schon! Mehr benötigt man nicht um den Server im Netzwerk frei zu geben.



Verbindung vom Client via ODBC zum SQL-Server

Zu testen, ob das alles auch funktioniert hat was wir da fabriziert haben, starten wir auf einem Client-Computer mit dem SQLAdmin Konto und pingen erst mal unter der Konsole den SQL-Server an. (Beispiel: `ping 10.0.2.20`). Ist bei der Verbindung alles in Ordnung, starten wir den ODBC-Datenquellen-Administrator. Hier kommt schon die erste, kleine Hürde die uns Microsoft in den Weg stellt. Ist der Client ein 32-bit Betriebssystem, muß auch der 32-bit ODBC-Manager gestartet werden. Erst ab Windows 8.1 wird der ODBC-Manager in zwei Versionen unter Verwaltung angezeigt. Unter Windows 7 ist der Manager für 32-bit unter `C:\Windows\System32\odbcad32.exe` zu finden, und der für 64-bit unter `C:\Windows\SysWOW64\odbcad32.exe`. Dieser heißt auch `odbcad32.exe` und nicht `odbcad64.exe` (Ich nehme mal an, aus Kompatibilitätsgründen)

Nachdem jetzt der entsprechende ODBC-Manager gestartet worden ist, wählen wir den Reiter „System-DSN“ damit alle Benutzer des Computers Zugriff auf die Datenbank bekommen. Mit einem Klick auf die Schaltfläche Hinzufügen wählen wir aus der Liste der installierten ODBC-Treibern den SQL-Server (alternativ auch den SQL-Server Native Client) aus. Unter Name geben wir z.B. Test ein. Diesen Namen benötigen wir später in unserem XProfan Programm, um die Datenbank anzusprechen. Unter Server können wir jetzt nach einem Server suchen lassen, und ggf. wird auch der SQLDATA angezeigt. **Diesen verwenden wir allerdings nicht!** Stattdessen geben wir unter Server die IP-Adresse des SQL-Servers und mit einem Komma getrennt den Port an: Beispiel: 10 . 0 . 2 . 20 , 1433



The screenshot shows a dialog box for configuring a System DSN. It has three main sections: 1. 'Name:' with a text box containing 'Test'. 2. 'Wie möchten Sie die Datenquelle beschreiben?' with a 'Beschreibung:' text box that is currently empty. 3. 'Mit welchem SQL Server möchten Sie sich verbinden?' with a dropdown menu showing 'Server: 10.0.2.20,1433'.

Als nächstes wird gefragt, wie wir uns an den Server anmelden möchten. Da wir als SQLAdmin angemeldet sind, können wir ruhig die Windows-Authentifizierung benutzen. Wie man einer Datenbank bestimmten Benutzern oder Gruppen zuordnen kann, wird in einem späteren Zeitpunkt beschrieben. Jetzt wollen wir ja nur mal testen, ob der Zugriff funktioniert. Im nächsten Schritt ändern wir die Standarddatenbank mal auf master, da wir ja noch keine eigene Datenbank erstellt haben. Im nächsten Fenster kann man noch ein paar Ländereinstellungen etc... vornehmen, was man zum Testen aber nicht unbedingt benötigt. Am Ende kann man noch die Datenquelle testen. Das war es auch schon mit der Anbindung.

Mit einem Mini-Programm unter XProfan, können Sie auch feststellen, ob die Anbindung funktioniert:

```
cls
print db("SQLINIT", "DSN=Test")
waitinput
db("SQLDone")
```

Wird ein Wert ausgegeben >0 (das Handle der Datenbankverbindung) war die Verbindung erfolgreich.