



Session V-ADS

Advantage Database Server in depth

Joachim Dürr

Abstrakt

Die ersten Schritte sind getan: (DBF-)Daten wurden in ADS integriert, die Applikation wurde umgestellt und arbeitet im Client/Server Modus. Doch Advantage kann noch mehr, als nur DBF in einer sicheren und stabilen Umgebung anzubieten.

Nun gilt es, in die Zukunft zu schauen.

Lernen Sie in dieser Session, wie Sie Ihre Datenbank mit Stored Procedures, Triggern, Replikation und ähnlichen Features erweitern können, um auch für kommende Anforderungen gewappnet zu sein.

ADS als reiner Verwalter der Daten

Es war einmal, vor langer, langer Zeit ... auch so könnte die Geschichte des Advantage Database Servers beginnen. Die Wurzeln reichen zurück bis in die frühen Neunziger, als Clipper das (PC-Datenbank-) Maß aller Dinge war und nächtliche Re-Organisationen (Pack und Reindex) auf der Prioritäten-Liste der Administratoren ganz oben stand. Mit ADS stand anfänglich eine Bibliothek zur Verfügung, welche Server-seitig diese Re-Organisation durchführen konnte und damit eine Menge Zeit einsparte. Nach und nach entwickelte sich der ADS erst zu einer Client/Server Erweiterung für Clipper, später dann zu einem voll funktionsfähigen relationalen Client/Server Datenbankmanagementsystem mit der Möglichkeit, die Daten entweder per ISAM (index sequential accessing method) oder aber per SQL abzufragen und zu ändern.

Von Anfang wurde ein Ziel konsequent verfolgt: Hohe Performance, hohe Verfügbarkeit und hohe Stabilität der Daten ohne die Notwendigkeit der ständigen Administration. Selbst bei der Erst-Inbetriebnahme ist dieses Konzept spürbar: Der ADS muss nur auf den vorhandenen Dateiserver installiert werden (Windows, Linux oder Novell Netware), eine spezielle Anpassung oder Einbindung der Datenbanken ist nicht notwendig. Wie bei Visual FoxPro gibt die Applikation den Pfad zur Datenbank vor, der Server dient nur als reiner Verwalter und kann dabei die Dateien vor unberechtigten Zugriffen schützen.

Unterstützte Tabellenformate

Das erste unterstützte Tabellenformat des ADS war Clipper (ADS_NTX), also DBF-Tabellen mit NTX Indizes und DBT Memo-Dateien. Mit einer Index-Ordnung pro Datei und maximal 15 Indexdateien ist dieses Format sehr beschränkt. Mehr Flexibilität bietet hier das so genannte FoxPro oder dBase Format (dBase III+, ADS_CDX) mit DBF-Tabellen, CDX Indizes und FPT Memos. Hier können bis zu 50 Index-Tags in einer Index-Datei enthalten sein, und das weiterhin bei der Grenze von maximal 15 Index-Dateien pro Tabelle. Abgeleitet davon wurde das eigene Tabellen-Format ADT (ADS_ADT) mit ADI-Indizes und ADM-Memos. ADT hat gegenüber DBF einige Vorteile:

- NULL-Unterstützung
- Automatische Wiederverwendung gelöschter Datensätze
- erweiterte Datentypen
- Längere Feld- und Indexnamen
- Längere Index-Schlüssel
- Echte Unique Indizes
- Binär Indizes

Seit Version 9 des ADS werden zusätzlich DBF-Tabellen vom Typ Visual FoxPro 9 direkt unterstützt (ADS_VFP). Erweiterungen, die dieses Format mit sich bringt, wurden auch in ADT eingepflegt

Waren DBF-Tabellen in der Vergangenheit auf 4GB (mit VFP 2GB) begrenzt, so können sie mit ADS bis zu 16 Trillionen Bytes anwachsen, die maximale Größe ist jedoch von der Datensatz-Breite abhängig (2GB * Record Länge).

Advantage Data Dictionaries - ein anderer DBC

Eine Sammlung von Einzel-Dateien reicht für das reine Speichern von Daten aus. DBMS der heutigen Zeit können jedoch noch viel mehr leisten, als nur die reine Datenspeicherung. So sollte das DBMS auch dafür Sorge tragen, nur valide Daten aufzunehmen und die Integrität der Datensätze untereinander sicherzustellen. Diese zusätzlichen Features sind bei Visual FoxPro über Database Container (DBC) implementiert. Es handelt sich im Grunde genommen um Meta-Tabellen, welche selbst nur die Struktur der Datenbank beinhalten, die Daten jedoch weiterhin in zusätzlichen DBF-Dateien speichert.

Auch der ADS bietet solch ein Konzept - Advantage Data Dictionaries (ADD). Ein ADD erweitert die Datenbank um Meta-Informationen und Objekte, welche bei freien Tabellen (Verzeichnis=Datenbank) nicht möglich wären. Die Tabellen verbleiben trotzdem noch als Einzel-Dateien im Dateisystem, können aber im Fall von ADT nicht mehr ohne das ADD selbst geöffnet werden (eine Veränderung des Headers trägt dafür Sorge). Sie brauchen jedoch keine Angst zu haben, dass Ihre DBF durch das Data Dictionary plötzlich unbrauchbar werden: DBF ist ein offen gelegter Standard und wird vom ADS nicht missbraucht. DBF-Tabellen bleiben trotz Aufnahme in ein Data Dictionary weiterhin auch von außerhalb ansprechbar, selbst von nicht-ADS-basierten Anwendungen.

Welche Erweiterungen ein Data Dictionary bringt, werden wir im folgenden sehen.

Erweiterte Tabellen- und Feldeigenschaften

Tabellen, die Teil eines Data Dictionaries sind, bieten in den Feldeigenschaften zusätzliche Optionen zur Spezifikation der Feld-Inhalte. Diese *Field Level Constraints* umfassen Default-, Minimal- und Maximalwert, welche als Definition nicht nur statische Werte bekommen können, sondern auch Funktionen aus der Advantage Expression Engine, welche an die Xbase-Syntax angelehnt ist. Daneben lässt sich noch auswählen, ob NULL ein zulässiger Wert ist. Werden beim Speichern diese Regeln verletzt, so kann eine eigene, angepasste Fehlermeldung als Exception geworfen werden.

Auch Tabellen bieten zusätzliche Eigenschaften: Da die Struktur der Tabellen und der Indizes bekannt ist, brauchen die zugehörigen Dateien bei leerem Inhalt nicht zwangsweise ausgeliefert werden. Ein Anhängen der Eigenschaft AutoCreate bewirkt, dass nicht vorhandene Dateien vom ADS beim Öffnen automatisch angelegt

werden. Sollen die Datei-Inhalte vor unberechtigten Blicken geschützt werden, so lässt sich die Tabelle mitsamt ihrer Indizes komplett verschlüsseln. Eine spezielle Entschlüsselung in der Applikation ist dazu nicht notwendig: Das übernimmt ADS durch die Benutzeranmeldung.

Gültigkeits-Regeln lassen sich nicht nur auf einzelne Felder, sondern auch auf komplette Datensätze anlegen. Diese nennen sich dann Record Level Constraints und verstecken sich in den Tabellen-Eigenschaften. Wie bei den Regeln auf Feld-Ebene mit eigener Fehlermeldung, falls dieser Constraint verletzt wird.

Valide Daten

Constraints sind auf einzelne Felder bzw. Tabellen beschränkt. In relationalen Datenbanken ist es jedoch auch wichtig, die einzelnen Tabellen untereinander korrekt zu halten. Dafür zuständig sind Referenz-Integritäts-Regeln (RI), welche auch über eine grafische Oberfläche per Drag'n'Drop erstellt werden können. RI-Regeln wirken auf Updates und Deletes von Master-Datensätzen und legen fest, wie bei vorhandenen Detail-Datensätzen vorzugehen ist. Für beide Aktionen gibt es vier Mögliche Varianten:

- Restrict: Es ist nicht erlaubt, den Primärschlüssel zu ändern bzw. den Datensatz zu löschen, solange noch abhängige Datensätze vorhanden sind
- Cascade: Die Änderung bzw. Löschung wird an die Detail-Datensätze durchgereicht, um die Integrität zu erhalten
- Set NULL: Bei Update des Primärschlüssels bzw. beim Löschen eines Master-Datensatzes wird das Fremdschlüsselfeld der abhängigen Detail-Datensätze auf NULL gesetzt. Die Integrität wird dadurch gebrochen, weil nicht zugeordnete Detail-Datensätze vorhanden sind.
- Set Default: Dasselbe wie Set NULL, nur wird der Fremdschlüsselwert auf den eingestellten Default gesetzt.

Views - Meine Sicht der Daten

Views bieten eine Möglichkeit, eine bestimmte Sicht auf die Daten zu gewähren. Dabei wird diese Sicht durch ein SQL-Select-Statement repräsentiert. Mögliche Anwendungsfälle sind das Ausblenden von bestimmten Spalten für unberechtigte Benutzer oder aber die Zusammenfassung von häufig benötigten Abfragen. Je nach zugrunde liegendem Statement sind diese Views beschreibbar oder nicht. Read-Only Views können jedoch durch das Anlegen von bestimmten (INSTEAD OF) Triggern beschreibbar gemacht werden - der Entwickler ist in diesem Falle dafür verantwortlich, die Datenänderungen an die Basis-Tabellen durchzureichen.

Benutzerverwaltung

Mit der Verwendung von Data Dictionaries bekommt der Entwickler eine komfortable Möglichkeit an die Hand, Berechtigungen auf alle Datenbank-Objekte, bei Tabellen bis auf Feld-Ebene, zu vergeben. Um diese Berechtigungen nicht für jeden Benutzer einzeln vergeben zu müssen, lassen sich Gruppen anlegen. Ist ein Benutzer dabei Mitglied mehrerer Gruppen, so akkumulieren sich deren Berechtigungen.

Für Standard-Aufgaben sind seit Version 9 des ADS Default-Gruppen angelegt. DB:Admin für Administratoren, DB:Backup für Benutzer, welche ein Hot Backup durchführen dürfen (also lesend auf alle Daten zugreifen können) und DB:Debug für Benutzer, denen es gestattet ist, angelegte SQL Skripten zu debuggen. Für allgemeinen Zugriff ist die Gruppe DB:Public zuständig. Dieser Gruppe gehören automatisch alle Benutzer an, um die allgemeine Freigabe von Tabellen und anderen Objekten zu vereinfachen.

Serverseitige Erweiterungen

Daten-intensive Operationen, welche nur ein kleines Resultset zurückliefern, gehören ebenso wie Kapselungen von Geschäftslogik als Stored Procedure in der Datenbank hinterlegt. ADS bietet die Möglichkeit, solche Erweiterungen entweder als ANSI-standardisierte SQL Skripten oder aber als Funktionen in externen Bibliotheken zu hinterlegen. Neben Win32-DLLs (bzw Shared Objects für Linux) können diese auch COM-

Objekte oder .NET Assemblies sein. Sie können also direkt in Ihrer geliebten Entwicklungsumgebung Funktionen erstellen und dem Server unterschieben.

Aber nicht nur Stored Procedures bieten diese Flexibilität: Dasselbe gilt auch für Trigger, welche als Reaktion auf Daten-Operationen automatisch ausgeführt werden. Als Typen stehen BEFORE, AFTER und INSTEAD OF Trigger auf alle Events (INSERT, UPDATE, DELETE) zur Verfügung. Speziell für die Replikation gibt es zwei weitere Trigger, um eventuell auftretende Konflikte aufzulösen.

Reicht der Befehls-Umfang der SQL Engine nicht aus, kann diese über User Defined Functions erweitert werden. UDF sind in der Datenbank hinterlegte SQL Skripten, welche beliebig viele Aufruf-Parameter und genau einen Rückgabe-Wert beinhalten können. Zur einfacheren Organisation und zur Vermeidung von Namenskonflikten können UDFs in Packages verpackt werden.

Fehlern in den serverseitigen Erweiterungen kommt man entweder über Debugger der Entwicklungsumgebung oder, im Fall von SQL Skripten, mit dem integrierten SQL Debugger auf die Schliche.

Hot Backup - eine heiße Sache

Ein großes Thema bei Datenbanken ist die Sicherung derselben. Zwar lassen sich DBF-Datenbanken (auch wenn sie von ADS im Zugriff sind) jederzeit über das Dateisystem kopieren, bei vielen großen Dateien kann es jedoch zu logischen Korruptionen führen, wenn die Datenbank momentan in Benutzung ist. Seit Version 8 des ADS gibt es die Möglichkeit, einen echten Schnappschuss der Datenbank anzufertigen. Advantage synchronisiert dazu die internen Worker-Threads bevor mit der Sicherung begonnen wird. Während dem Backup auftretende Datenänderungen merkt sich der Server und sichert statt dem aktuellen Datensatz den alten Stand, so dass die logische Integrität der Daten erhalten bleibt. Für Batch-Vorgänge (CRON, MS Scheduler) steht ein Kommandozeilen-Programm zur Verfügung.

Replikation

Müssen verteilte Systeme synchron gehalten werden, so kommt die ADS Replikation zum Zuge. Direkt in den Server eingebaut ist eine asynchrone 1:n Push Replikation. Datenänderungen werden erkannt und in eine Log-Tabelle, der Replication Queue, geschrieben. Ein Hintergrund-Prozess arbeitet diese Änderungen ab und schickt sie an angemeldete Datenbanken (Subscriptions). Steht eines dieser Ziele nicht zur Verfügung, so wird die Queue Table gefüllt bis die Datenbank wieder erreichbar ist. Damit lässt sich auf einfachste Weise ein Briefcase-Modell realisieren.

Replikation kann nur im Zusammenhang mit dem ADS eingesetzt werden, der Local Server (ALS) wird nicht unterstützt. Eine separate Lizenzierung ist dafür notwendig.

Zusammenfassung

Wie Sie sehen konnten ist ADS mehr als nur ein reiner Datenspeicher. Selbst mit DBF-Dateien lässt sich eine Datenbank realisieren, welche einem reinen SQL Server in nichts nachsteht.

Visual FoxPro Entwickler profitieren nicht nur von wesentlich größeren Tabellen, sondern auch von der Möglichkeit, den Server mit eigenen (COM-) Bibliotheken zu erweitern.

Solange die DBF-Tabellen nicht die 2GB-Grenze überschreiten, können sie sogar parallel von nicht-ADS-basierten Anwendungen genutzt werden, so dass einer sanften Migration gewachsener Applikationen nach Client/Server nichts mehr im Wege steht.