Session D-REPA

Reparatur von VFP Datenbanken

Uwe Habermann
Uwe@hceood.eu

Einführung

VFP Datenbanken sind stabil und zuverlässig. Trotzdem kommt es in seltenen Fällen doch einmal vor, dass Datenbanken beschädigt werden. In diesem Vortrag werden mögliche Ursachen erläutert und Lösungen zur Reparatur gezeigt. Dabei kann es um beschädigte Tabelleninhalte. Tabellenköpfe, Indexdateien, Memodateien oder auch Datenbank-Container gehen. Es wird ein Weg zur automatisierten Reparatur beim Kunden vorgestellt. Es wird auch erörtert wie defekte Datenbanken, wie sie zum Beispiel durch SMB 2 und neuer entstehen können, vermieden werden können.

Die Idee

Hinweis: Ungeachtet der hier vorgestellten Reparaturmöglichkeiten sollten natürlich unbedingt häufig Datensicherungen gemacht werden.

Im Allgemeinen sind VFP Datenbanken stabil und zuverlässig. Trotzdem kann es unter ungünstigen Bedingungen passieren, dass Datenbanken beschädigt werden. Es wäre schön ein Werkzeug zu haben, das eine beschädigte VFP Datenbank automatisch reparieren kann. VFP bietet alle notwendigen Hilfsmittel, um so ein Werkzeug zu konstruieren.

Mit VFP wird das Werkzeug GenDBC.prg geliefert, das im Ordner Tools\Gendbc unterhalb der VFP Installation zu finden ist. GenDBC öffnet eine gültige VFP Datenbank, analysiert die Struktur und generiert eine Prg-Datei. Diese Prg-Datei enthält Code, mit dem eine neue Datenbank generiert werden kann. Falls die Datenbank gespeicherte Prozeduren enthält, dies ist insbesondere dann der Fall, wenn Trigger verwendet werden, generiert GenDBC außerdem eine Datei mit der Namenserweiterung .krt. Diese Krt-Datei enthält den Quellcode der gespeicherten Prozeduren sowie den Objektcode. Bei der Ausführung der generierten Prg-Datei wird der Inhalt der Krt-Datei ausgelesen und die gespeicherten Prozeduren und der Objektcode werden in die neue, leere Datenbank eingefügt.

Auf diese Weise lässt sich eine gültige Datenbank mit der gewünschten Struktur erstellen. Damit haben wir die wichtige Grundlage für die Reparatur einer Datenbank.

Wie kann man Fehler erkennen?

In den meisten Fällen erkennen Anwender eine beschädigte Datenbank daran, dass zur Laufzeit ihrer Anwendung eine Fehlermeldung erscheint. Die folgenden VFP Fehlermeldungen deuten auf einen Fehler in der Datenbank hin.

In selteneren Fällen beschweren sich Anwender, weil sie die gesuchten Daten nicht finden können. Dies deutet in den meisten Fällen auf eine beschädigte Indexdatei hin.

Noch seltener bemerken Anwender, dass ungültige Daten angezeigt werden. So können Zeichen in einem numerischen Feld oder in einem Datumsfeld stehen oder unerwartete Zeichen in einem Zeichenfeld stehen. In so einem Fall ist der Inhalt von Datensätzen beschäftigt. Solche Fehler werden meistens durch Fehler in Netzwerken verursacht, wenn ungültige Datenpakete transportiert werden.

5 && Datensatz außerhalb des Bereichs.
15 && Keine Tabelle.
19 && Indexdatei stimmt nicht mit Tabelle überein.
20 && Datensatz nicht im Index.
41 && Die Memodatei fehlt oder ist ungültig.
114 && Index stimmt nicht mit Tabelle überein. Löschen Sie die Indexdatei und erstellen Sie den Index erneut.
1550 && Interner .DBC-Übereinstimmungsfehler.
1551 && Datei ist eine ungültige Datenbank.
1552 && Datei ist keine Datenbank.
1553 && Datei ist eine Datenbank.
1561 && Datenbank ist ungültig. Bitte Gültigkeitsprüfung durchführen.
1562 && Objekt konnte nicht in der Datenbank gefunden werden.
1563 && Ansicht konnte nicht in der aktuellen Datenbank gefunden werden.
1567 && Die Primärschlüssel-Eigenschaft ist ungültig; bitte überprüfen Sie die Gültigkeit der Datenbank.
1707 && Strukturierte .CDX-Datei wurde nicht gefunden.
1884 && Die Eindeutigkeit des Indexes wird verletzt.
1886 && Index akzeptiert keine Nullwerte.

Liste der Fehlermeldungen

Die Anwendung zur Datenbankreparatur

Zunächst muss der Entwickler die gültige Datenbank dem Projekt hinzufügen. Auf dem Entwicklungsrechner ist die Datenbank selbstverständlich gültig vorhanden. Diese gültige Datenbank ist die Grundlage für die Reparatur.

Beim Erstellen einer Exe-Datei aus diesem Projekt wird mithilfe eines Project Hook eine angepasste Version von GenDBC.prg aufgerufen, die sich im Projekt befindet und den Namen GenDbcFunc.prg hat. Hiermit wird eine Prg-Datei generiert. Falls sich in der Datenbank gespeicherte Prozeduren befinden, wird zusätzlich eine Datei mit der Namenserweiterung .krt generiert. Die generierten Dateien werden mit dem Project Hook vor der Erstellung der Exe-Datei in das Projekt eingeschlossen und anschließend wieder entfernt. Damit kann mit der Exe-Datei eine gültige Datenbank erstellt werden.

Wenn die Exe-Datei ausgeführt wird, erscheint zunächst ein Ordnerauswahldialog, mit dem der Ordner der beschädigten Datenbank ausgewählt werden kann. Die Datenbank im ausgewählten Ordner wird repariert. Es ist nicht sinnvoll den Anwender hier die zu reparierende Datenbank auswählen zu lassen, weil der Datenbank-Container fehlen könnte und dies der Grund für die benötigte Reparatur ist.

Wahlweise kann der Pfad, in dem sich die zu reparierende Datenbank befindet, der Exe-Datei als Parameter übergeben werden.

Wie funktioniert die Reparatur?

Eine Datenbank kann auf vielfältige Weise beschädigt sein kann. Bei der Reparatur gehen wir vom schlechtesten Fall aus. Das heißt, eine oder mehrere Dateien des Datenbank-Containers sind beschädigt oder fehlen, Tabellen oder dazugehörigen Memo- oder Indexdateien sind beschädigt oder fehlen und/oder Tabellenköpfe oder Tabelleninhalte sind beschädigt.

Schritte der Datenbankreparatur

1. Es wird ein weiterer temporärer Ordner angelegt, im weiteren Verlauf *NEWDATA* genannt. In diesem Ordner wird die Funktion CreateEmptyDbc() ausgeführt. Mit dieser Funktion wird der von GenDBC generierte Code ausgeführt und es wird eine neue, leere Datenbank angelegt.

2. In ersten Schritt wird die Datenbank aus dem beim Start ausgewählten Ordner, im weiteren Verlauf *ORIGINAL* genannt, in einen temporären Ordner kopiert. Der Name dieses temporären Ordners wird mit der Funktion SYS(2015) generiert. Die Reparatur wird im temporären Ordner durchgeführt. Die Dateien im Ordner *ORIGINAL* bleiben unberührt bis die Reparatur abgeschlossen ist.

3. Im temporären Ordner wird die Datenbank exklusiv geöffnet.

4. Wenn die Datenbank bei Schritt 2 nicht geöffnet werden konnte, werden die drei Dateien des Datenbank-Containers (.dbc, .dct, .dcx) aus dem Ordner *NEWDATA* in den temporären Ordner kopiert.

5. Im Ordner *NEWDATA* befindet sich eine gültige Datenbank mit allen benötigten Dateien. Falls im temporären Ordner Dateien fehlen, werden diese aus dem Ordner *NEWDATA* in den temporären Ordner kopiert. Hierbei kann es sich um .dbf, .cdx oder .fpt Dateien handeln.

6. Die drei Dateien des Datenbank-Containers (.dbc, .dct und .dcx) werden in einen weiteren neuen Ordner kopiert, den wir *DBCORG* nennen.

7. Im temporären Ordner wird die Datenbank exklusiv geöffnet. Dies sollte nach Schritt 5 gelingen. Falls dies nicht gelingt, wird die Datenbankreparatur abgebrochen und alle temporär angelegten Dateien und Ordner werden gelöscht.

8. Im temporären Ordner wird jetzt jede Tabelle repariert.

Es ist praktisch kaum möglich festzustellen, ob der Kopf einer Tabelle fehlerfrei ist. Eine Prüfung ist aber auch nicht erforderlich. Im Ordner *NEWDATA* sind alle Tabellen mit gültigen Köpfen vorhanden. Die Größe eines Tabellenkopfes kann mit der Funktion HEADER() ermittelt werden. Mit den Dateifunktionen FOPEN() und FREAD() kann der Tabellenkopf in eine Variable gelesen werden und mit FWRITE() in die vermeintlich beschädigte Tabelle geschrieben werden. Vor dem Zurückschreiben des Tabellenkopfes muss anhand der Dateigröße und der Größen von Tabellenkopf und Datensatz die Anzahl der Datensätze berechnet werden und die Bytes 4 bis 7 des Tabellenkopfes eingetragen werden.

Anschließend wird die Tabelle exklusiv geöffnet. Falls dabei der Fehler 114 (Index stimmt nicht mit Tabelle überein. Löschen Sie die Indexdatei und erstellen Sie den Index erneut.) auftritt, wird die korrekte Cdx-Datei aus dem Ordner *NEWDATA* kopiert. Damit haben wir eine gültige Indexdatei, die jedoch keine Referenzen auf die Datensätze enthält. Dies lässt sich mit dem Befehl REINDEX beheben, den wir zu einem späteren Zeitpunkt ausführen.

Falls der Fehler 41 (Die Memodatei fehlt oder ist ungültig.) auftritt, können wir mit den hier vorgestellten Mitteln nichts machen. Mit dem Werkzeug Memoscan, das mit FoxFix geliefert wird, können Memodateien repariert werden. Memoscan wird weiter unten erläutert. Beschädigte

Memodateien können wir leider nur entfernen, was mit dem Verlust der beschädigten Daten verbunden ist. Die fehlerfreie, aber leere FPT-Datei wird aus dem Ordner *NEWDATA* kopiert. Danach kann die Tabelle fehlerfrei geöffnet werden und die Felder der Typen Memo, Objekt und Blob werden aus der Tabellenstruktur entfernt.

Danach werden die Trigger aus der Tabelle entfernt.

Jetzt werden korrupte Datensätze als gelöscht markiert. Wenn hierbei Datensätze gelöscht werden, wird anschließend der Befehl PACK ausgeführt, damit Datensätze mit möglicherweise ungültigen Primärschlüsseln entfernt werden. Bei der Ausführung des Befehls PACK wird die Tabelle automatisch auch neu indiziert. Wenn keine Datensätze gelöscht wurden, wird der Befehl REINDEX ausgeführt, um die Neuindizierung vorzunehmen.

Im Ordner *NEWDATA* wird ein ggf. vorhandenes Feld vom Typ Integer (Autoinc) in den Typ Integer konvertiert. Dadurch wird das Feld beschreibbar.

Jetzt sind alle Vorbereitungen getroffen und die Daten aus der Tabelle im temporären Ordner werden an die Tabelle im Ordner *NEWDATA* angefügt. Da wir im Ordner *NEWDATA* Tabellen mit der vollständigen und gültigen Struktur haben, sind auch die zuvor möglicherweise entfernten Felder vom Typ Memo, Objekt oder Blob vorhanden.

9. Alle Datenbanken werden geschlossen.
10. Die drei Dateien des Datenbank-Containers (DBC, DCT und DCX) werden aus dem Ordner DBCORG in den Ordner NEWDATA kopiert. Damit sind die Trigger wieder in der Datenbank und auch die Felder vom Typ Integer (Autoinc) sind wieder richtig.
11. Im Ordner *NEWDATA* wird für jede Tabelle der nächste Wert für Felder vom Typ Integer (Autoinc) gesetzt.
12. Alle Datenbanken werden geschlossen.
13. Es werden alle Dateien aus dem Ordner *NEWDATA* in den Ordner *ORIGINAL* kopiert.
14. Alle temporär angelegten Dateien und Ordner werden gelöscht.

Damit ist die Reparatur abgeschlossen. Im Ordner *ORIGINAL* befindet sich eine gültige Datenbank. Alle Tabellen haben eine gültige Struktur und lassen sich öffnen. Alle gültigen Datensätze sind erhalten geblieben.

Beispiele für Fehler, die automatisch repariert werden können

- Fehlende Dateien des Datenbank-Containers
- Fehlende .dbf, .cdx oder .fpt-Dateien
- Ungültiger Dateikopf in einer der Dateien
- Ungültige Primärschlüssel
- Zerstörter Datensätze
- Zerstörte Indexdatei

Was kann diese Anwendung nicht reparieren?

Leider lassen sich nicht alle Probleme automatisch lösen. Dies betrifft insbesondere beschädigte Memodateien.

Ein professionelles Werkzeug zur Reparatur von VFP Datenbanken bietet die Firma Xitech an. Es ist das Werkzeug FoxFix. FoxFix kommt mit einem speziellen Reparaturwerkzeug für Memodateien: MemoScan.exe.

MemoScan analysiert Memodateien, zeigt ungültige Zeiger an und kann diese auch reparieren.

Weitere Informationen sind auf der Seite des Herstellers zu finden:

http://www.xitech-europe.co.uk/foxfix.php

SMB

Seit dem Erscheinen von VFP 9 wurden von Microsoft mehrere Generationen von Windows Server veröffentlicht. Damit wurden auch neue Versionen von SMB, dem Server Message Block-Protokoll herausgegeben. Hiermit wurden neue Technologien zum Datentransfer im Netzwerk eingeführt. Vom ursprünglichen Blockmodus ist man zum Stream-Modus übergegangen. Der Blockmodus wird von VFP aber für Datensatzsperren und Aktualisierungen von Indexdateien benötigt. Mit den neuen Versionen von SMB kann es zu Dateninkonsistenzen und zu zerstörten Indexdateien kommen.

In den letzten Jahren gab es verschiedene Tipps zu diesem Thema und Registry-Patches, mit denen Server auf das ursprüngliche SMB Protokoll zurückgesetzt werden sollten. Mit der Abschaltung der neueren SMB Versionen waren Netzwerk-Administratoren oft nicht einverstanden.

Es hat sich aber gezeigt, dass einfachere Änderungen zum gewünschten Ergebnis führen. Tatsächlich kann SMB 2 oder SMB 3 auf dem Server aktiviert bleiben. Überhaupt muss auf den Servern keinerlei Änderung vorgenommen werden und es kann trotzdem stabil mit VFP Datenbanken gearbeitet werden.

Es ist völlig ausreichend, wenn an den Arbeitsplatz-Rechnern die Caching-Einstellungen deaktiviert werden, die in Zusammenhang mit neueren SMB Versionen eingeführt wurden.

Durch diese Patches werden auch andere Programme wieder stabil lauffähig, die ebenfalls mit neueren Versionen von SMB Probleme haben, wie zum Beispiel Access.

Hier die Registry-Einträge, mit denen das SMB Caching ausgeschaltet wird:

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\LanmanWorkst
ation\Parameters]
"FileInfoCacheLifetime"=dword:00000000
"FileNotFoundCacheLifetime"=dword:00000000
 "DirectoryCacheLifetime"=dword:00000000
```

Da das SMB 2 Protokoll beim Erscheinen zahlreiche Fehler hatte, wurden viele Hotfixes veröffentlicht, um die Probleme zu beseitigen.

Zusätzlich zur Anwendung der Registrierungsschlüssel von oben ist daher die Anwendung der Hotfixes unbedingt empfehlenswert.

Jürgen Wondzinski alias wOOdy hat sich intensiv mit den neueren Versionen des SMB Protokolls beschäftigt und die sich daraus ergebenden Probleme für VFP Anwendungen analysiert. wOOdy hat auch die Hotfixes zusammengestellt und in einer Excel-Tabelle den Ablauf bei der Installation der Hotfixes beschrieben.

Das Archiv mit den Hotfixes und der Installationsbeschreibung ist hier zu finden:

http://www.hceood.eu/download/CeBIT2015-SMB-Fixes.zip