

Upsizing zum SQL Server

Microsoft Corp.

Der Upsizing Wizard, der mit Visual FoxPro ausgeliefert wird, erfordert eine signifikante Aktualisierung, um einige Defizite zu beseitigen und um wichtige neue Features zu unterstützen.

Hintergrund, Anforderungen und Anpassung

Der Upsizing Wizard enthält einige Defizite. Dazu gehören:

- Leere Felder vom Typ Datum und DatumZeit werden nicht korrekt behandelt, da es im SQL Server wie in anderen Datenbanken kein Konzept eines leeren DatumZeit-Werts gibt.
- Unter bestimmten Umständen kann das Laden von Daten in die upgesizeten Tabellen sehr langsam vor sich gehen.
- Felder, die mit einem im SQL Server reservierten Wort benannt sind, werden während des Upsizings umbenannt (als Beispiel wird ein Feld namens „Order“ in „Order_“ umbenannt). Dies kann zu Problemen mit bestehenden Programmen, Berichten, Formularen usw. führen.

Es sind verschiedene neue Features geplant:

- Der Wizard kann optional als Engine ohne Benutzeroberfläche ausgeführt werden. Dadurch ist es Entwicklern möglich, die Aufgabe des Upsizings programmgesteuert zu automatisieren, ohne sich manuell durch die verschiedenen Schritte der Oberfläche des Assistenten klicken zu müssen.
- Die neue Engine kann vom Data Explorer aus aufgerufen werden. Im Idealfall sollte der Anwender in der Lage sein, mit der rechten Maustaste auf eine VFP-Datenbank zu klicken und einige minimale Fragen zu beantworten (beispielsweise den Namen des Servers und der Datenbank); anschließend erstellt die Upsizing Engine eine SQL Server- oder andere Datenbank mit den am häufigsten verwendeten Optionen.
- Es werden besonders SQL Server 2005 und SQL Server 2005 Express unterstützt.
- Der Wizard wird Hooks bereitstellen, so dass er durch Entwickler erweitert werden kann, die sein Verhalten an verschiedenen Stellen des Upsizing-Prozesses an ihre Anforderungen anpassen wollen.
- Auf verschiedenen Stufen des Upsizing-Prozesses werden Ereignisse ausgelöst, so dass eine Komponente mit einer Benutzeroberfläche BINDEVENT() verwenden könnte, um eine Fortschrittsanzeige anzuzeigen.

Ziele

Die Ziele für einen aktualisierten Upsizing Wizard in Sedna lassen sich wie folgt beschreiben:

- Wiederverwenden möglichst vieler Teile des aktuellen Upsizing Wizards, wobei der Rest der Ziele erreicht wird.
- Entfernen aller Abhängigkeiten der Engine von visuellen Oberflächenelementen, auch von Formular-Steuerelementen und Nachrichtenanzeigen.
- Sicherstellen, dass der Wizard von vielen unterschiedlichen Stellen aus aufgerufen werden kann, auch vom Data Explorer aus.
- Definieren von Feldern der Datentypen Datum und DatumZeit in der Zieltabelle, die Nullwerte akzeptieren, wenn die die Felder des Typs Datum und DatumZeit in der Quelltablelle teilweise leer sind, auch wenn die Quellfelder keine Nullwerte akzeptieren, und Laden des Werts NULL statt leerer Werte in die Quelltablelle.
- Wenn möglich Verbesserung der Performance beim Laden der Daten.
- Behandeln der Felder, die mit reservierten Worten des SQL Server benannt sind, mit Trennzeichen (beispielsweise “[Order]“).
- Wenn erforderlich Unterstützung aller neuen Features von SQL Server 2005 und SQL Server 2005 Express.
- Anlegen eines Erweiterungsobjekts, ähnlich der Vorschauanwendung von VFP 9. Dieses Objekt ermöglicht es einem Entwickler, das Verhalten der Engine an verschiedenen Stellen des Upsizing-Prozesses zu erweitern oder zu verändern.
- Hinzufügen der Unterstützung auftretender Ereignisse an verschiedenen Stellen des Upsizing-Prozesses.

Abhängigkeiten

Die Unterstützung des Aufrufs aus dem Data Explorer heraus erfordert, dass die Data Explorer-Anwendung aktualisiert wird, damit sie die Upsizing-Engine aufruft, wenn der Anwender dies wünscht.

Szenarien

Eine VFP-Datenbank vom Data Explorer aus mit dem Wizard umwandeln

Bob möchte eine bestehende VFP-Datenbank in eine SQL Server 2005-Datenbank umwandeln. Er ruft den Task Pane Manager auf und wählt dort den Data Explorer. Er klickt auf die Schaltfläche Add Connection, wählt im Dialog Add Connection FoxPro Database und wählt seine VFP-Datenbank. Nachdem die Datenbank im Data Explorer angezeigt wird, klickt er mit der rechten Maustaste darauf und wählt im Kontextmenü den Eintrag Upsizing Wizard. Alternativ dazu kann er die VFP-Datenbank auf einen SQL Server 2005 ziehen, der im Data Explorer angezeigt wird.

Der Upsizing Wizard wird aufgerufen. Bob arbeitet sich durch die verschiedenen Schritte des Assistenten und wählt die Optionen aus, die er verwenden will (einige Optionen, beispielsweise die umzuwandelnde Datenbank, sind aufgrund des Kontexts des Data Explorers bereits ausgefüllt).

Er wählt Finish, um die Datenbank umzuwandeln. Nachdem der Prozess beendet ist, wird der Data Explorer aktualisiert, um die umgewandelte Datenbank anzuzeigen.

Dieses Szenario ist lediglich eine Abkürzung, die dem Aufruf des Upsizing Wizard aus dem Menü Extras | Assistenten entspricht.

Eine VFP-Datenbank vom Data Explorer aus mit der Engine upsizen

Jane möchte eine bestehende VFP-Datenbank in eine SQL Server 2005-Datenbank upsizen. Sie klickt auf die Schaltfläche Add Connection, wählt im Dialog Add Connection FoxPro Database und wählt ihre VFP-Datenbank. Nachdem die Datenbank im Data Explorer angezeigt wird, klickt sie mit der rechten Maustaste darauf und wählt im Kontextmenü den Eintrag Upsizing Wizard.

Es wird ein Dialog aufgerufen, die Sie auffordert, die Datenquelle auszuwählen, die umgewandelt werden soll. Sie kann eine Connection in der Datenbank oder eine DSN auswählen oder einen Verbindungsstring angeben. Anschließend gibt sie an, ob sie eine neue Datenbank erstellen oder die bestehende Datenbank umwandeln möchte. Im ersten Fall gibt sie den Namen der Datenbank an. Im zweiten Fall wählt sie aus einer Liste bestehender Datenbanken eine aus. Sie klickt auf die Schaltfläche Upsize, um den Prozess zu starten. Eine Fortschrittsanzeige im Dialog zeigt Ihnen den Fortschritt der Umwandlung an. Ist der Prozess beendet, wird der Dialog geschlossen und der Data Explorer wird aktualisiert, um die umgewandelte Datenbank anzuzeigen.

Eine VFP-Datenbank programmgesteuert upsizen

Mary verfügt über mehrere bestehende VFP-Datenbanken, die Sie auf SQL Server 2005 upsizen möchte. Da die Datenbanken recht groß sind, möchte sie diese Aufgabe nicht interaktiv erledigen, sondern der Prozess soll über Nacht durchgeführt werden.

Sie schreibt eine Routine, die eine Datenbank öffnet, die Upsizing Engine instantiiert, die Eigenschaften der Engine entsprechend der gewünschten Optionen einstellt (Verbindungsinformation, neuer oder bestehender Datenbankname, Feldmappings usw.) und die Datenbank upsizet. Sie testet diese Routine mit einer kleinen Datenbank, um sicherzustellen, dass sie ohne Fehler und ohne Benutzeroberfläche funktioniert. Wenn sie mit der Arbeitsweise des Prozesses zufrieden ist, schreibt sie ein „Treiber“-Programm, das die Routine für jede Datenbank, die upgesizet werden soll, einmal aufruft und führt dieses Programm vor dem Feierabend aus.

Funktionale Beschreibung

Upsizing Engine ohne Benutzeroberfläche

Die bestehende Upsizing Engine ist eine Klasse namens UpsizeEngine in WizUsz.PRG. Die Klasse weist viele Abhängigkeiten von der Benutzeroberfläche auf. Eine Abhängigkeit ist das Vorhandensein des Formulars des Upsizing Wizards. So enthält die Methode AnalCleanUp den folgenden Code:

```
*Turn off the recordsource of grid and columns to prevent errors when source table  
is closed
```

```
OWizard.Form1.PageFrame1.Page1.PageFrame1.Page7.grdTypeMap.RECORDSOURCE=""  
OWizard.Form1.PageFrame1.Page1.PageFrame1.Page7.grdTypeMap.RECORDSOURCE=""
```

Eine andere Art der Benutzeroberfläche besteht darin, dass einige Methoden die Funktion MESSAGEBOX() aufrufen, um dem Anwender Fehlermeldungen anzuzeigen. Eine dritte Art ist der Einsatz von WAIT WINDOW und einer Fortschrittsanzeige.

Code wie dieser wird in einer reinen Engine-Umgebung oder wenn er von einer anderen Quelle, beispielsweise dem Data Explorer, aufgerufen wird, nicht funktionieren. Diese Probleme können wie folgt beseitigt werden:

- Abhängigkeiten im Upsizing Wizard können in eine von UpsizeEngine abgeleitete Klasse verschoben werden. Diese Klasse wird durch den Upsizing Wizard statt UpsizeEngine verwendet.
- Fehlermeldungen werden im Array `aErrorMessages` gespeichert, nicht mit `MESSAGEBOX()` angezeigt; eine Benutzeroberfläche wie der Upsizing Wizard kann anschließend die Inhalte dieses Arrays anzeigen, um dem Anwender die Probleme anzuzeigen, die während des Laufs des Assistenten aufgetreten sind.
- Statt ein `WAIT WINDOW` oder etwas Ähnliches zu verwenden, um den Fortschritt anzuzeigen, wird der Klasse eine leere Methode namens `UpdateStatus` hinzugefügt und es wird `RAISEEVENT('UpdateStatus', UpdateMessage)` verwendet. Jede Benutzeroberfläche, auch der Upsizing Wizard, die über den Fortschritt des Upsizings informiert werden möchte, wird mit `BINDEVENT()` an `UpsizeEngine.UpdateStatus` gebunden.

Die Möglichkeit, aus dem Data Explorer heraus aufgerufen zu werden

Die anderen Änderungen an der Upsizing Engine, die in diesem Abschnitt beschrieben werden, betreffen die Möglichkeit, die Upsizing Engine von anderen Stellen als dem Upsizing Wizard aufzurufen, beispielsweise aus dem Data Explorer heraus. Diese Möglichkeit wird speziell getestet, um sicherzustellen, dass mit dieser Vorgehensweise keine Probleme auftreten.

Felder vom Typ Datum und DatumZeit korrekt behandeln

UpsizeEngine enthält drei Routinen, die Änderungen erfordern, um Felder vom Typ Datum und DatumZeit korrekt zu behandeln. Die erste Routine ist `AnalyzeFields`, die für das Lesen der Tabellenstruktur verantwortlich ist, die upgesizet werden soll. Während der Prüfung der Struktur einer Tabelle muss diese Methode außerdem feststellen, ob Datensätze mit leeren Feldern vom Typ Datum oder DatumZeit vorhanden sind. Ist dies der Fall, müssen diese Felder markiert werden, damit sie NULL-Werte akzeptieren, auch wenn dies in der VFP-Version der Tabelle nicht der Fall ist.

Die anderen Routinen, die geändert werden müssen, sind `JimExport` und `ExecuteProc`, die für das Upsizing der Daten in einer speziellen Tabelle verwendet werden. Diese Methoden müssen in leere Felder vom Typ Datum und DatumZeit NULL-Werte einsetzen.

Verbesserung der Performance beim Laden von Daten

Der Befehl `BULK INSERT` des SQL Servers ist eine sehr schnelle Möglichkeit, Daten in eine Datenbank zu laden. Allerdings verwendet die Methode `SendData` in UpsizeEngine einen langsameren Mechanismus für das Laden von Daten, wenn Felder der Tabelle, die nicht vom Typ Zeichen sind, NULL-Werte akzeptieren. Der Grund dafür ist, dass bei der Verwendung von `COPY TO` für das Erstellen einer Textdatei die NULL-Werte verloren gehen und stattdessen Leerzeichen ausgegeben werden. Um die Performance des Ladens der Daten zu verbessern, wird die Textdatei, die durch die Methode `GenBulkInsertTextFile` erstellt wurde, so verarbeitet, dass leere Werte entfernt werden, und `SendData` wird geändert, um die Verwendung von `BULK INSERT` unter diesen Bedingungen zu ermöglichen.

Behandlung reservierter Objektnamen

Die Methode `RemotizeName` von `UpsizeEngine` sucht in einer Schlüsselworttabelle nach Objektnamen. Wird eine Übereinstimmung gefunden, wird das Objekt durch das Anhängen eines „_“ umbenannt. Statt das Objekt umzubenennen, werden Namens-Trennzeichen wie „[“ und „]“ hinzugefügt.

Die Behandlung von SQL Server 2005 und SQL Server 2005 Express

Ein Problem mit der Zusammenarbeit des Upsizing Wizards mit dem SQL Server 2005 besteht darin, dass in der Liste, in der der Anwender seine Datenquelle auswählt, keine Datenquellen angezeigt werden, die den SQL Native Client ODBC-Treiber verwenden, es sei denn, die Option `All` ist eingeschaltet, da speziell nach dem SQL Server-Treiber gesucht wird. Drei Methoden der Klasse `ChooseDataSource` (in `MyCtrls.VCX`) werden geändert, um den SQL Native Client-Treiber zu unterstützen: `GetServerType`, `InsDataSource` und `ServerType`.

Erweiterbarkeit bereitstellen

Die Vorschauanwendung von VFP 9 bietet die Erweiterbarkeit durch den Einsatz eines `Preview Extension`-Objekts. Ist ein solches Objekt vorhanden, rufen viele Methoden der Anwendung Methoden des Objekts auf. Dadurch hat ein Entwickler die Möglichkeit, das Verhalten der Berichtsvorschau zu erweitern. In ähnlicher Weise wird `UpsizeEngine` ein `Upsizing Extension`-Objekt unterstützen. Der Klasse wird die neue Eigenschaft `oExtension` hinzugefügt. Die meisten Methoden der Klasse werden prüfen, ob `oExtension` ein Objekt enthält und ob dieses Objekt eine Erweiterungsmethode für die aktuelle Methode enthält. Ist dies der Fall, wird die Methode des Objekts aufgerufen und ihr wird eine Referenz auf das `UpsizeEngine`-Objekt übergeben, wenn die Methode Eigenschaften referenziert oder Methoden der Upsizing Engine aufruft.