

16 Konzeption des Data Warehouse

Für das Verständnis dieses Kapitels wird die Lektüre der Kapitel 3 und 4 vorausgesetzt. Kenntnisse der Kapitel 5 ff. benötigen Sie dagegen nur insoweit, als gezielt darauf verwiesen wird.

Voraussetzungen

16.1 Konzeptionelle Überlegungen zur SQL-Lösung

Bei der SQL-Lösung müssen genau dieselben Probleme gelöst werden wie bei der Lösung mit Access, die in den vorangehenden Kapiteln beschrieben wurde: Aus denselben Quelldaten, die aus fünf Inseln stammen, soll ein konsistentes Data Warehouse erstellt werden, auf dessen Daten mit OLAP-Tools zugegriffen werden kann. Prinzipiell wäre es möglich, die dort gefundenen Wege, insbesondere die Organisation und Durchführung der Transformationsprozesse, auf die Situation des SQL Servers praktisch 1:1 zu übertragen: Es gibt keine Abfrage, die sich mit dem SQL Server nicht ebenso gut realisieren ließe wie mit der JetEngine von Access. VBA-Prozeduren allerdings könnten mit dem SQL Server nicht umstandslos realisiert werden, weil dieser keine Programmiermöglichkeit zur Verfügung stellt. In aller Regel jedoch dürfte in der Arbeitsumgebung eines SQL Servers eine Möglichkeit zur Realisierung von VB- oder VBA-Prozeduren vorhanden sein, so dass dies kein ernstliches Gegenargument wäre. Wenn dann das Datenzugriffsobjekt ADO (statt DAO, wie bei unserer Access-Lösung) verwendet wird, kann sehr leicht auf die Objekte des SQL Servers zugegriffen werden, so dass Datenmanipulationen einfach möglich sind.

Es ist aber zu berücksichtigen, dass uns mit dem SQL Server in Form des Transact-SQL viel mehr Möglichkeiten der SQL-Sprache zur Verfügung stehen als mit der SQL-Sprachvariante bei Access. Diese können so angewandt werden, dass ein Rückgriff auf VB-Prozeduren nicht erforderlich ist. Nicht zuletzt aus didaktischen Gründen ist es mein Ziel zu demonstrieren, dass die gesamte SQL-Lösung allein mit den Möglichkeiten des Microsoft SQL Servers und der mit ihm ausgelieferten Komponenten zu realisieren ist. Dies bedeutet im Einzelnen:

► DTS

Alle Transformationsprozesse sollen mit Hilfe von DTS organisiert und ausgeführt werden. Dies hat nicht nur den Vorteil eines einheitlichen Vorgehens, sondern es schafft auch ein großes Maß an Transparenz: Allein schon die Visualisierung im Fenster des DTS-Designers macht den komplexen Transformationsprozess durchsichtig und schafft Überblick,

wo dieser beispielsweise bei der Access-Lösung leicht verloren geht. DTS stellt eine hervorragende Möglichkeit dar, die einzelnen Transformationsprozesse als Module zu erstellen, sie aber gleichwohl als Teile einer Gesamtaufgabe zu konzipieren und darzustellen. Dies bringt nicht nur große Vorteile beim Debugging des Gesamtprozesses mit sich, sondern erleichtert auch die Pflege des Data Warehouse in hohem Maße.

► **Transact-SQL**

Die Konzentration auf Transact-SQL als der einzigen verwendeten »Sprache« weist nicht nur den Vorteil der Einheitlichkeit auf, sondern gewährleistet auch ein hohes Maß an Performance. Wenn darüber hinaus die Möglichkeiten von gespeicherten Prozeduren, Variablen und Funktionen genutzt werden, steht Transact-SQL der Verwendung von Visual Basic für die Manipulation von Daten in nichts nach.

► **Verzicht auf Visual Basic-Prozeduren**

Bis auf eine Rahmenprozedur, die das DTS-Paket startet und ein paar äußerliche Bereinigungen an den ASCII-Daten vornimmt, sollen für die SQL-Lösung keine Visual Basic-Prozeduren eingesetzt werden.

► **Analysis Services**

Für den Zugriff von multidimensionalen Analysewerkzeugen wie beispielsweise PivotTables, aber auch solchen von Drittanbietern, werden die mit dem SQL Server ausgelieferten Analysis Services genutzt. Dies bedeutet praktisch, dass Cubes auf dem OLAP Server gebildet werden, auf die über OLE DB von allen diesbezüglichen Frontends zugegriffen werden kann. In der Version 7.0 des SQL Servers wurden die Analysis Services der Version 8.0 als OLAP Services bezeichnet. Der Unterschied zwischen beiden besteht im Wesentlichen in der Benennung, denn in beiden Versionen steht der OLAP Server im Mittelpunkt des Features. Selbstverständlich ist dieser in der Version 8.0 weiterentwickelt worden. Gleichwohl kann die Lösung, die hier entwickelt wird, mit beiden Versionen realisiert werden.

16.2 Modellierung des Data Warehouse

Bei der Bildung des Data Warehouse geht es darum, die Daten aus den Dateninseln in eine relationale Datenbank zu überführen. Hierbei handelt es sich um eine strukturelle Aufgabe, die nach den allgemeinen Regeln für das Design relationaler Datenbanken unabhängig vom speziellen RDBMS gelöst werden muss. Daraus ergibt sich, dass das Datenbank-Design im Falle der Access-Lösung prinzipiell mit dem der SQL-Lösung übereinstim-

men muss. Entsprechend wird hier auch verfahren: Das Data Warehouse der Access-Lösung ist in der Access-Datenbank *LocAm1.mdb* bzw. *LocAm2.mdb* realisiert worden. Die SQL Server-Datenbank *LocAm1SQL* bzw. *LocAm2SQL* weist dieselbe Tabellenstruktur auf wie die entsprechende Access-Datenbank. Aus diesem Grunde können die Überlegungen zur Modellierung des Data Warehouse der Access-Lösung auf die Modellierung der entsprechenden SQL Server-Datenbank übertragen werden. Es würde eine nicht vertretbare Wiederholung bedeuten, wenn sie an dieser Stelle für die SQL-Lösung noch einmal ausführlich wiedergegeben würden. Daher sei insoweit auf Kapitel 5, Abschnitt 5.2, verwiesen. Wo sich Abweichungen von den dort vorgegebenen Wegen ergeben, wird an den betreffenden Stellen darauf hingewiesen. In jedem Falle aber wird in den folgenden Ausführungen die Realisierung jedes einzelnen Objekts der Datenbank *LocAm2SQL*, in der das Data Warehouse der SQL-Lösung ja realisiert wird, behandelt.

16.3 Einbettung der Aktualisierung in eine Visual Basic-Rahmenprozedur

Wie bereits gesagt, sollen sämtliche Transformations- und damit Aktualisierungsprozesse als DTS-Tasks realisiert werden, so dass die Aktualisierung des Data Warehouse dann durch Ausführen des entsprechenden DTS-Pakets vorgenommen wird. Aus zwei Gründen habe ich den Start des DTS-Pakets in eine VB-Prozedur eingebettet:

- ▶ Wenn das Ausführen des DTS-Pakets nicht zeitgesteuert automatisch, sondern manuell erfolgen muss, wie es im vorliegenden Fall geboten ist, erscheint der Start des Pakets durch eine VB-Prozedur die sicherste und bequemste Startart, besonders dann, wenn das entsprechende Programm in kompilierter Form als .exe-Datei vorliegt.
- ▶ Die Art und Weise, in der die Transformation der ASCII-Daten im vorliegenden Falle geregelt ist, setzt eine VB-gesteuerte Operation an den aktuellen ASCII-Dateien voraus, die nicht (bzw. nur unzureichend) vom DTS-Paket selbst gestartet werden kann.

Aus diesen beiden Gründen ist es zweckmäßig, die VB-Prozedur, welche die Operation an den ASCII-Daten vornimmt, gleich mit einer anderen Prozedur, die das DTS-Paket startet, zusammen in einem VB-Programm zu vereinen. Dies ist im Visual Basic-Projekt *DW_SQL_Update.vbp* geschehen, das Sie auf der Begleit-CD im Pfad `\DW\SQL` finden. Das Startformular – es ist zugleich das einzige Formular – dieses Programms ist in Abbildung 16.1 wiedergegeben: Nachdem Jahr und Monat für die Aktua-

lisierung markiert sind, wird durch Klicken auf die Schaltfläche *Aktualisieren* eine Prozedur gestartet, welche die Datenbereinigung der aktuellen ASCII-Dateien vornimmt. Wenn diese Operation erledigt ist, wird mit einem Dialogfeld gefragt, ob das DTS-Paket gestartet werden soll. Falls diese Frage bejaht wird, wird das Paket ausgeführt. Der Code zum Starten eines DTS-Pakets aus einer VB-Prozedur heraus wurde bereits oben wiedergegeben und kurz erklärt, vgl. Abschnitt 15.6.4, Unterpunkt *Ausführen von Paketen aus einer Visual Basic-Prozedur heraus*. Die VB-Prozedur, welche die ASCII-Daten bereinigt, wird weiter unten im Zusammenhang mit der Transformation der ASCII-Daten wiedergegeben und erklärt, vgl. Abschnitt 17.1, Unterpunkt *Bereinigung der ASCII-Quelldatensätze von Flatterzeilen*.



Abbildung 16.1 Dialogfeld Aktualisierung Data Warehouse LocAmSQL

16.4 Keine Transformation der Anschriften

Im Rahmen der Access-Lösung wurden auch Anschriften aus drei Datenquellen (Fibudaten, Personaldaten und ad hoc angefügte Anschriften) in das Data Warehouse einbezogen. Darauf wird im Rahmen der SQL-Lösung verzichtet.

17 Transformation der ASCII-Daten in relationale Form

Für das Verständnis dieses Kapitels wird die Lektüre der Kapitel 3 und 4 vorausgesetzt. Kenntnisse der Kapitel 5 ff. benötigen Sie dagegen nur insoweit, als gezielt darauf verwiesen wird.

Voraussetzungen

17.1 Konzeption der Transformation in relationale Form

Wie bereits bei der Access-Lösung muss auch hier für die Transformation der Personaldaten im ASCII-Format, die als Berichte vom Mainframe kommen, ein verhältnismäßig großer Aufwand betrieben werden. Der Weg von den .asc-Dateien in die Tabellen der SQL Server-Datenbank *LocAm2SQL* soll zunächst beispielhaft an den Personalstammdaten problematisiert werden. Diese kommen aus einer ASCII-Datei *S01MMJJ.asc* und werden in die Tabelle *PersonalStamm* der Datenbank *LocAm2SQL* transformiert. Entsprechend den oben in Abschnitt 3.2.1, dargestellten Anforderungen ist es erforderlich, nicht nur einzelne Felder vom Textformat in Zahlen- und Datumsformate zu transformieren, sondern es muss auch für jeden Datensatz geprüft werden, ob der betreffende Mitarbeiter bereits in der Tabelle *PersonalStamm* geführt wird oder nicht, und entsprechend muss ein INSERT oder UPDATE ausgeführt werden.

Es scheint sich anzubieten, eine derartige Aufgabe mit dem DTS-Task *Datengesteuerte Abfrage* zu lösen, sieht diese doch explizit das Ausführen einer Insert- oder Update-Abfrage in Abhängigkeit von Bedingungen in den Daten vor. Tatsächlich lässt sich das Problem nicht mit diesem Task bearbeiten, weil die datengesteuerte Abfrage nur die Daten der Datenquelle, nicht jedoch die des Datenziels prüfen kann. Letzteres aber ist hier erforderlich, weil in der Zieltabelle *PersonalStamm* geprüft werden muss, ob eine bestimmte Personalnummer bereits vorhanden ist oder nicht. In der Access-Lösung für dieses Problem haben wir eine VBA-Prozedur verwendet, in der mit der *Seek*-Methode des *Recordset*-Objekts nach bestimmten Datensätzen in der Zieltabelle gesucht wird. Da wir uns im Rahmen der SQL-Lösung entschieden haben, keine VB-Prozeduren für die Transformationsprozesse zu verwenden, bleibt als Lösungsweg nur der über *Transact-SQL*, ein Weg, der allerdings ebenso gut zu gehen ist.

Daher muss eine Möglichkeit gefunden werden, mit SQL-Anweisungen zu prüfen, ob für eine Personalnummer, die aus den Quelldaten eingelesen wurde, in der Zieltabelle *PersonalStamm* bereits ein entsprechender Da-

Temporäre
Tabelle

tensatz vorhanden ist. Dies setzt voraus, dass nicht nur das Datenziel als Tabelle der Datenbank *LocAm2SQL* vorhanden ist, wie dies mit Personalstamm zutrifft, sondern es müssen auch die Quelldaten in relationaler Form vorliegen. Aus diesem Grunde ist es erforderlich, zwischen die Quelldaten im ASCII-Format *S01MMJJ.asc* und die Zieltabelle Personalstamm in der Datenbank *LocAm2SQL* eine *temporäre Tabelle* einzufügen, deren Aufgabe nur darin besteht, alle Datensätze der aktuellen *.asc*-Datei zum Zugriff in relationaler Form anzubieten. Diese Tabelle bekommt den Namen *Actual_S01*. Für die beiden anderen Typen von ASCII-Dateien der Personaldaten, *P01MMJJ.asc* und *P02MMJJ.asc*, soll aus Gründen der einheitlichen Organisation der Transformationsprozesse entsprechend verfahren werden, so dass auch die beiden weiteren Tabellen *Actual_P01* und *Actual_P02* eingerichtet werden müssen.

**Bereinigung der
ASCII-Quelldaten-
sätze von
Flutterzeilen**

Die Datenzeilen der ASCII-Dateien haben trotz des Formats fester Spaltenbreiten für die einzelnen Felder unterschiedliche Längen. Dies liegt letztlich daran, dass sie als Reports vom Mainframe erzeugt werden, wobei sich Wiederholungsfelder ergeben. Wenn diese am Ende einer Datenzeile angeordnet sind, werden nur so viele Felder ausgegeben, wie diese mit Inhalt gefüllt sind. Daher können die Datenzeilen flattern. Die Transformation der ASCII-Dateien in die temporäre relationale Form soll mit dem DTS-Task *Daten transformieren* ausgeführt werden. Bei der Entwicklungsarbeit zeigte sich, dass dieser Task feste Zeilenlängen voraussetzt, wenn, wie bei uns erforderlich, als Datenquelle eine Textdatei mit festen Spaltenbreiten für die Felder angegeben wird. Bei flatternden Datenzeilen wird der Task *Daten transformieren* dagegen im Allgemeinen mit einem Fehler beendet. Dieses Problem habe ich auf die folgende Weise gelöst:

Entsprechend den drei temporären Tabellen *Actual_S01*, *Actual_P01* und *Actual_P02* in der SQL Server-Datenbank *LocAm2SQL* werden die drei jeweils aktuellen ASCII-Dateien *S01MMJJ.asc*, *P01MMJJ.asc* und *P02MMJJ.asc* auf Betriebssystemebene in die drei temporären ASCII-Dateien *Actual_S01.asc*, *Actual_P01.asc* und *Actual_P02.asc* transformiert. Diese Transformation ist im Prinzip nichts anderes als eine reine Kopie der Datenzeilen mit folgender Bereinigung: Durch Anfügen von Leerzeichen wird jede Datenzeile auf dieselbe Länge gebracht, so dass die Zeilenlänge in den drei ASCII-Dateien *Actual_S01.asc*, *Actual_P01.asc* bzw. *Actual_P02.asc* dann einheitlich 160, 1100 bzw. 760 beträgt.

Da eine Operation an den ASCII-Daten aus den vorgenannten Gründen ohnehin erforderlich war, wurde noch eine weitere Manipulation vorgenommen, um den Mangel zu beseitigen, dass den Inhalten der ASCII-Da-

teien nicht zu entnehmen ist, auf welches Aktualisierungsdatum sie sich beziehen. (Dies ist ohne die folgende Ergänzung nur am Namen der originalen ASCII-Dateien erkennbar.) Der Beginn jeder Datenzeile wurde um den Monatsersten des Aktualisierungsmonats ergänzt, und zwar im Format JJJJMMTT, so dass beispielsweise jede Datenzeile für den Aktualisierungsmonat Februar 2000 mit der Zeichenfolge 20000201 beginnt. In der VB-Prozedur wird dies durch die Anweisung

```
Print #2, Jahr & Monat & "01" & strText
```

erreicht. Die explizite Übernahme des Monatsersten in die ASCII-Dateien ermöglicht es, diese Information auch in die entsprechenden Tabellen Actual_xxx der SQL Server-Datenbank *LocAm2SQL* weiterzugeben. Da die weiteren Transformationsprozesse völlig ohne VB-Prozeduren organisiert sind, sondern allein mit den Mitteln von DTS und Transact-SQL, erweist sich die Möglichkeit, in einer jeweils aktuellen Tabelle den aktuellen Monatsersten zu ermitteln, als großer Vorteil.

Das Listing der Prozedur *ASCIIDateienBereinigen* ist im Folgenden wiedergegeben.

```
Private Sub ASCIIDateienBereinigen( _  
    DateiNamensanfang, Jahr, Monat, Zeilenlänge)  
    Dim strInputFile, strOutputFile, strText, Länge  
    strInputFile = "V:\Input\" & DateiNamensanfang _  
        & Monat & Right(Jahr, 2) & ".ASC"  
    strOutputFile = "V:\Input\Actual_" _  
        & DateiNamensanfang & ".ASC"  
  
    Open strInputFile For Input As #1  
    Open strOutputFile For Output As #2  
    Do While Not EOF(1)  
        Line Input #1, strText  
        Länge = Len(strText)  
        If Länge < Zeilenlänge Then  
            strText = strText _  
                & Space(Zeilenlänge - Länge)  
        End If  
        Print #2, Jahr & Monat & "01" & strText  
    Loop
```

```
Close #1
Close #2
End Sub
```

Listing 17.1 Code der VB-Prozedur ASCIIDateienBereinigen

Die bereinigende Kopie wird mit der VB-Prozedur ASCIIDateienBereinigen durchgeführt, deren Code im Listing 17.1 wiedergegeben ist. Sie finden diese Prozedur auf der Begleit-CD im Visual Basic-Projekt \DW:\SQL\DW_SQL_Update.vbp im Klassenmodul des Formulars *Form_Update*. Die Prozedur ASCIIDateienBereinigen wird aus dem Dialogfeld *Aktualisierung Data Warehouse LocAmSQL* heraus, das oben in Abbildung 16.1 wiedergegeben ist, ausgeführt: Durch Klicken auf die Schaltfläche *Aktualisieren* wird die Ereignisprozedur *Aktualisieren_Click* gestartet. Ihr Code ist in Listing 17.2 wiedergegeben. Sie finden diese Prozedur ebenfalls auf der Begleit-CD im Visual Basic-Projekt \DW:\SQL\DW_SQL_Update.vbp im Klassenmodul des Formulars *Form_Update*.

```
Private Sub Aktualisieren_Click()
Dim Pfad, Dateiname, Präfix(1 To 3) As String, i
On Error GoTo Aktualisieren_Click_1
If Me.Jahr = "" Or Me.Monat = "" Then
    MsgBox "Sie müssen ein Jahr " _
        & "und einen Monat markieren!"
    Exit Sub
End If

Pfad = "V:\Input\"
Präfix(1) = "S01"
Präfix(2) = "P01"
Präfix(3) = "P02"

For i = 1 To 3
    Dateiname = Präfix(i) & Me.Monat _
        & Right(Me.Jahr, 2) & ".ASC"
    If Dir(Pfad & Dateiname) = "" Then
        MsgBox "Die Datei '" & Pfad & Dateiname _
            & "' wurde nicht gefunden."
        Exit Sub
    End If
Next i
```

```

ASCIIDateienBereinigen Präfix(1), Me.Jahr, _
    Me.Monat, 160
ASCIIDateienBereinigen Präfix(2), Me.Jahr, _
    Me.Monat, 1100
ASCIIDateienBereinigen Präfix(3), Me.Jahr, _
    Me.Monat, 760

If MsgBox("DTS-Paket jetzt starten?", vbYesNo) = _
    vbYes Then
    Screen.MousePointer = vbHourglass
    'OLAPCubesAufbereiten
    DTSPaketAusführen
    Screen.MousePointer = 1
End If

Exit Sub

Aktualisieren_Click_1:
Screen.MousePointer = 1

End Sub

```

Listing 17.2 Code der Ereignisprozedur Aktualisieren_Click

Die Prozedur `Aktualisieren_Click` liest die im Dialogfeld markierten Zeitangaben (Jahr und Monat) aus und ruft dann die Prozedur `ASCIIDateienBereinigen` für jede der drei zu transformierenden ASCII-Dateien auf, wobei die jeweils zutreffenden Argumentwerte übergeben werden. Nach Beendigung der (sehr schnellen) drei Transformationen wird mit einem `MsgBox`-Dialogfeld gefragt, ob das DTS-Paket gestartet werden soll. Dies kann bejaht oder verneint werden.

Die Gesamtkonzeption der Überführung der ASCII-Daten in relationale Form ist in Abbildung 17.1 grafisch dargestellt: In einer ersten Transformation werden die drei originären ASCII-Dateien mittels VB-Prozedur in drei temporäre ASCII-Dateien bereinigt kopiert. Die zweite Transformation – die Überführung der ASCII-Daten in drei temporäre Datenbanktabellen – wird mit DTS-Tasks vom Typ *Daten transformieren* erledigt. Die dritte Transformation schließlich, welche die Personaldaten in die fünf normalisierten Personaltabellen überführt, wird mit DTS-Tasks vom Typ *SQL ausführen*, in denen jeweils Transact-SQL-Skripte angegeben sind, durchgeführt.

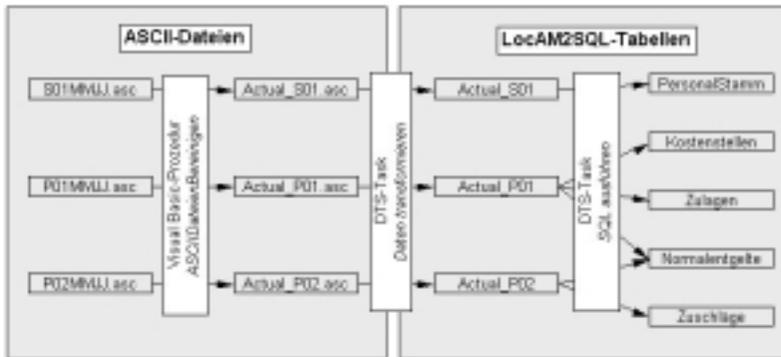


Abbildung 17.1 Transformation der ASCII-Daten in die Tabellen des SQL Servers

17.2 Realisierung der VB-Prozeduren ASCIIDateienBereinigen und Aktualisieren_Click

Übungstabellen
in LocAm2SQL
löschen

Falls Sie es nicht bereits am Ende von Kapitel 15, erledigt haben, sollten Sie spätestens jetzt die im Verlauf von Kapitel 15 in der SQL Server-Datenbank *LocAm2SQL* angelegten Übungstabellen löschen. Es handelt sich um die folgenden Tabellen:

- ▶ Buchungen
- ▶ Normalentgelte
- ▶ PersonalStamm
- ▶ Products

Sie löschen diese wie andere Datenbankobjekte, indem Sie aus dem Kontextmenü den Befehl *Löschen* wählen.

Visual Basic-
Projekt DW_SQL_
Update.vbp

Der Code der beiden Prozeduren ist im vorangehenden Abschnitt in Listing 17.1 und Listing 17.2 wiedergegeben. Das Visual Basic-Projekt, das diese beiden Prozeduren enthält, finden Sie auf der Begleit-CD im Pfad `\DW:\SQL\DW_SQL_Update.vbp`, die beiden Prozeduren im Klassenmodul des Formulars *Form_Update*.



- ▶ Kopieren Sie die drei zum VB-Projekt gehörenden Dateien

`\DW:\SQL\DW_SQL_Update.vbp`

`\DW:\SQL\DW_SQL_Update.frm`

`\DW:\SQL\DW_SQL_Update.frx`

von der Begleit-CD in einen beliebigen Ordner Ihrer Festplatte, am besten in den Ordner `v:\`. Sie können dann das VB-Projekt *DW_SQL_*

Update.vbp mit Visual Basic öffnen und starten. Dann wird das Startformular angezeigt, das oben in Abbildung 16.1 wiedergegeben ist. Den Code der Prozeduren finden Sie im Klassenmodul dieses Formulars.

- ▶ Das Visual Basic-Projekt kann auch als ausführbare .exe-Datei kompiliert und gespeichert werden. Falls Sie die SQL-Lösung des Data Warehouse später als Gesamtlösung weitergeben wollen, empfiehlt sich dieser Weg vermutlich. Sie sollten aber erst dann eine .exe-Datei erstellen, wenn auch das DTS-Paket für die Aktualisierung des Data Warehouse fertig ist und fehlerfrei vom Visual Basic-Projekt aufgerufen und ausgeführt werden kann.

Für Leser, die nicht über Visual Basic 6.0 verfügen, habe ich dasselbe Projekt auch als Access-Datenbank realisiert, die Sie auf der Begleit-CD im Pfad `\DW:\SQL\DW_SQL_Update.mdb` finden. Daher besteht die Alternative:

Access-Datenbank
`DW_SQL_Update.mdb`

- ▶ Kopieren Sie die Access-Datenbank `\DW:\SQL\DW_SQL_Update.mdb` von der Begleit-CD in einen beliebigen Ordner Ihrer Festplatte, am besten in den Ordner `v:\`. Sie finden dort das Formular *Aktualisierung* und in dessen Klassenmodul den Code der Prozeduren. Die Prozeduren sind gegenüber den oben angegebenen Listings der VB-Prozeduren an wenigen Stellen leicht abgeändert, um den besonderen Erfordernissen von Access-VBA und des Access-Steuerelements *Listenfeld* zu entsprechen.



Die folgende, zunächst einmalige Ausführung der Aktualisierungsprozedur hat nicht nur Testcharakter, sondern wird auch für die weitere Entwicklung des Data Warehouse auf dem SQL Server vorausgesetzt, denn diese ist darauf angewiesen, dass die drei Dateien *Actual_S01.asc*, *Actual_PO1.asc* und *Actual_PO2.asc* mit Datensätzen gefüllt sind. Im weiteren Text wird davon ausgegangen, dass es sich dabei um Datensätze für Februar 2000 handelt.

Ausführen für
Februar 2000

- ▶ Starten Sie das Visual Basic-Projekt *DW_SQL_Update*, oder öffnen Sie die Access-Datenbank *DW_SQL_Update.mdb* und darin das Formular *Aktualisierung*.
- ▶ Markieren Sie in den Listenfeldern das Jahr 2000 und den Monat 02, und klicken Sie dann auf die *Schaltfläche Aktualisieren*. Beantworten Sie die anschließende Frage, ob das DTS-Paket gestartet werden soll, mit *Nein*.
- ▶ Prüfen Sie, ob sich nunmehr im Pfad `v:\Input` die drei ASCII-Dateien *Actual_S01.asc*, *Actual_PO1.asc* und *Actual_PO2.asc* mit Größen von ca. 24 KB, 109 KB und 76 KB befinden.



17.3 Realisierung der Transformation von Actual_S01.asc

Um diese Transformation zu bewerkstelligen, sind drei DTS-Tasks zu erstellen:

► **SQL ausführen: Actual_S01 löschen**

In einem ersten Schritt werden die gegebenenfalls vorhandenen Datensätze in der SQL Server-Tabelle Actual_S01 mit dem DTS-Task *SQL ausführen* gelöscht.

► **Daten transformieren: Transformation nach Actual_S01**

In einem zweiten Schritt werden die Datenzeilen der ASCII-Datei Actual_S01.asc mit dem DTS-Task *Daten transformieren* entsprechend dem Spaltenplan in die temporäre SQL Server-Tabelle Actual_S01 transformiert. Dabei wird jede Datenzeile übernommen.

► **SQL ausführen: Actual_S01 verarbeiten**

In einem dritten Schritt wird mit den Datensätzen der temporären Tabelle Actual_S01 mit dem DTS-Task *SQL ausführen* in der Tabelle PersonalStamm ein INSERT oder UPDATE ausgeführt.

SQL Server-Tabelle Actual_S01 erstellen

Die Tabelle Actual_S01 muss zunächst einmal in der Datenbank *LocAm2SQL* erstellt werden. Sie weist die in Abbildung 17.2 wiedergegebenen Felddefinitionen auf.

	Spaltenname	Datentyp	Größe	NULL zulassen
PK	Persnr	int	4	
	Nachname	nvarchar	30	✓
	Vorname	nvarchar	30	✓
	Geburtsdatum	smalldatetime	4	✓
	Eintrittsdatum	smalldatetime	4	✓
	Strasse	nvarchar	25	✓
	PLZ	nvarchar	5	✓
	Ort	nvarchar	34	✓
	Sex	nvarchar	2	✓
	DW_Eintrittsdatum	smalldatetime	4	✓
	Austrittsdatum	smalldatetime	4	✓

Abbildung 17.2 Felddefinitionen der Tabelle Actual_S01



► Erstellen Sie die Tabelle Actual_S01 in der Datenbank *LocAm2SQL* manuell entsprechend den Definitionsmerkmalen aus Abbildung 17.2.

► Alternativ: Kopieren Sie die Tabellenstruktur aus der Datenbank *LocAm1SQL* mit dem DTS-Task *SQL Server-Objekte kopieren*, der oben in Abschnitt 15.4.3, erklärt ist.

Im Laufe der weiteren Arbeit am Data Warehouse für die SQL-Lösung werden Sie vermutlich oft einzelne Objekte aus der Datenbank *LocAm1SQL* in die Datenbank *LocAm2SQL* kopieren, statt diese manuell zu erstellen. Statt jedes Mal einen neuen DTS-Task *SQL Server-Objekte kopieren* zu erstellen, empfehle ich Ihnen das folgende Vorgehen: Erstellen Sie einmal einen DTS-Task *SQL Server-Objekte kopieren*, z. B. für den im letzten Aufzählungspunkt genannten Zweck. Speichern Sie dann dieses DTS-Paket mit nur dem einen Task unter einem beliebigen Namen. Wenn Sie wiederum ein SQL Server-Objekt kopieren wollen, öffnen Sie dieses DTS-Paket, öffnen Sie dann das Eigenschaftfenster für den Task, und bearbeiten Sie die Einstellungen auf der Registerkarte *Kopieren* entsprechend dem zu kopierenden Objekt. Führen Sie den Task anschließend aus.

Tipp

Mit dem folgenden Unterpunkt beginnen Sie, das DTS-Paket für die Transformationen des Data Warehouse zu erstellen. Dabei sind einige recht komplexe Tasks einzufügen und zu konfigurieren, so dass sich leicht Fehler im Erstellungsprozess ergeben können. Auf der Begleit-CD finden Sie das DTS-Paket *\DW\SQL\LocAm2SQL.dts*, das die fertige DTS-Lösung für die Datenbank *LocAm2SQL* in Form einer strukturierten Speicherdatei enthält. Wenn Sie dieses Paket gleichzeitig mit dem zu erstellenden Paket *LocAm2SQL* öffnen, können Sie während des Arbeitens am letzteren Paket die einzelnen DTS-Tasks und -Verbindungen vergleichen und/oder testen.

**DTS-Paket
LocAm2SQL.dts
als Referenz**

Zum Testen ist es allerdings erforderlich, die Verbindung zu Ihrem SQL Server einzurichten, denn im abgespeicherten Zustand sind die Verbindungen zu meinem SQL Server mit dem Namen *BROSIUSPDC\SQL8_1* definiert. Gehen Sie zu diesem Zweck wie folgt vor:

- ▶ Öffnen Sie das DTS-Paket auf die folgende Weise: Klicken Sie im Enterprise Manager mit der rechten Maustaste auf den Knoten *Data Transformation Services*, wählen Sie den Befehl *Paket öffnen...*, wählen Sie im Dialogfeld *Datei auswählen* von der Begleit-CD die Datei *\DW\SQL\LocAm2SQL.dts*, und bestätigen Sie die Dialogfelder. Dann wird das DTS-Paket im Fenster des DTS-Designers geöffnet.
- ▶ Doppelklicken Sie auf die erste Verbindung zum SQL Server mit der Bezeichnung *LocAm2SQL*, das ist die Zielverbindung des ersten Tasks *Daten transformieren* mit der Bezeichnung *Transformation nach Actual_S01*, um das Eigenschaftfenster der Verbindung zu öffnen.
- ▶ Wählen Sie als Server Ihren SQL Server und dazu die Datenbank *LocAm2SQL*.

- ▶ Bestätigen Sie das Eigenschaftenfenster mit OK. Dann wird das Dialogfeld *Taskverweise* angezeigt, weil durch die geänderte Verbindung möglicherweise Transformationen in Tasks, die auf die geänderte Verbindung verweisen, fehlerhaft werden könnten. Das trifft im vorliegenden Fall jedoch nicht zu. Bestätigen Sie daher das Dialogfeld, ohne eine Kontrollkästchen aktiviert zu haben.
- ▶ Speichern Sie das geänderte DTS-Paket. Dies können Sie mit dem Befehl *Datei, Speichern* oder mit *Datei, Speichern unter...* bewerkstelligen. Beim ersten Weg wird die strukturierte Speicherdatei *LocAm2SQL.dts* aktualisiert (sie dürfte dann allerdings nicht direkt von der Begleit-CD geöffnet worden sein), beim zweiten Weg können Sie das Paket auf dem SQL Server oder als strukturierte Speicherdatei auf Ihrer Festplatte speichern. Wenn Sie das Paket auf dem SQL Server speichern, sollten Sie nicht den Namen *LocAm2SQL* verwenden, denn dieser wird in den folgenden Schritten als Name für das hier zu entwickelnde DTS-Paket verwendet.

17.3.1 Task SQL ausführen »Actual_S01 löschen«

DTS-Verbindung
LocAm2SQL
erstellen

Der erste Task *SQL ausführen* setzt voraus, dass eine *DTS-Verbindung* zur Datenbank *LocAm2SQL* definiert ist. Gehen Sie folgendermaßen vor, um diese zu erstellen:



- ▶ Öffnen Sie ein neues DTS-Paket: Klicken Sie im Enterprise Manager mit der rechten Maustaste auf den Knoten *Data Transformation Services*, und wählen Sie im Kontextmenü den Befehl *Neues Paket*.
- ▶ Fügen Sie im Fenster des DTS-Designers eine neue Verbindung *Microsoft OLE DB Provider for SQL Server* ein, und konfigurieren Sie diese so, dass sie auf die SQL Server-Datenbank *LocAm2SQL* verweist. Geben Sie ihr die Bezeichnung »LocAm2SQL«, indem Sie diese Zeichenkette in das Eingabefeld rechts von der Option *Neue Verbindung* eingeben.

Task SQL aus-
führen erstellen

Dieser Task soll alle Datensätze in der Tabelle *Actual_S01* löschen, damit mit dem folgenden Task *Daten transformieren* die aktuellen Datensätze aus der ASCII-Datei *Actual_S01.asc* in die leere Tabelle eingefügt werden können. Gehen Sie folgendermaßen vor, um den Task einzurichten:



- ▶ Fügen Sie im Fenster des DTS-Designers einen Task *SQL ausführen* ein.
- ▶ Geben Sie im Eigenschaftenfenster dieses Tasks als Beschreibung den Text »Actual_S01 löschen« an, belassen Sie es bei der angezeigten vorhandenen Verbindung *LocAm2SQL*, und geben Sie als SQL-Anweisung den folgenden Code ein:

```
DELETE FROM Actual_S01
```

- ▶ Bestätigen Sie mit OK, und testen Sie den Task: Klicken Sie mit der rechten Maustaste auf den Task, und wählen Sie im Kontextmenü den Befehl *Schritt ausführen*.

17.3.2 Task Daten transformieren »Transformation nach Actual_S01«

DTS-Verbindung zu Actual_S01.asc erstellen



Bevor der Task *Daten transformieren* eingerichtet werden kann, muss eine *DTS-Verbindung* zur Datenquelle *Actual_S01.asc* erstellt werden (eine Verbindung zum Datenziel ist bereits erstellt worden). Gehen Sie folgendermaßen vor:

- ▶ Fügen Sie in das Fenster des DTS-Designers eine neue Verbindung vom Typ *Text File (Source)* ein.
- ▶ Geben Sie im Eigenschaftenfenster dieser Verbindung als Bezeichnung »Actual_S01.asc« an, und wählen Sie als Dateiname *v:\Input\Actual_S01.asc*.
- ▶ Klicken Sie auf die Schaltfläche *Eigenschaften...* Die voreingestellten Angaben zum Dateiformat im Dialogfeld *Textdateieigenschaften* sind in Ordnung, klicken Sie daher auf die Schaltfläche *Weiter >*. Dann bietet das Dialogfeld *Textdateieigenschaften* die Möglichkeit zur Festlegung von *Spaltenpositionen bei Feldern fester Breite*, vgl. Abbildung 17.3.

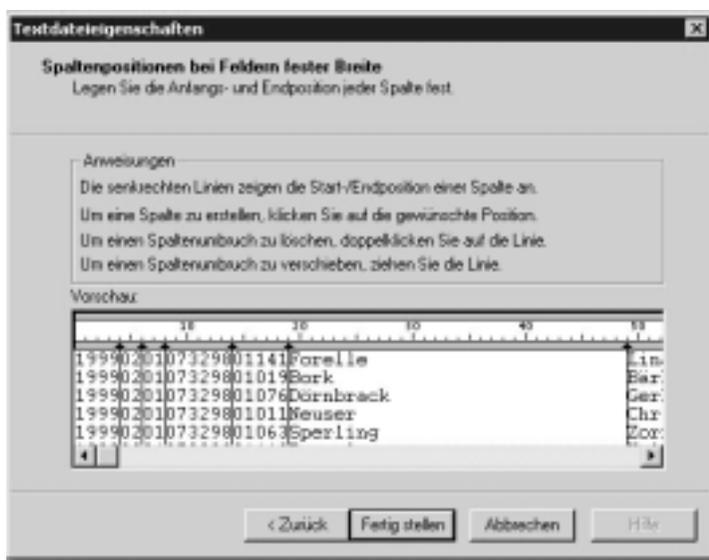


Abbildung 17.3 Dialogfeld *Textdateieigenschaften* zum Festlegen der Spaltenpositionen

- Legen Sie die Spaltenpositionen nach folgendem Plan¹ fest:

FeldNr	Endposition
1	4
2	6
3	8
4	14
5	19
6	49
7	79
8	83
9	85
10	87
11	91
12	93
13	95
14	120
15	125
16	159
17	161

Tabelle 17.1 Spaltenpositionen für Textdatei Actual_S01.asc

- Klicken Sie auf die Schaltfläche *Fertig stellen* und anschließend auf OK.



Task Daten transformieren »Transformation nach Actual_S01« erstellen

Nachdem je eine DTS-Verbindung für Datenquelle und Datenziel definiert worden ist, kann nunmehr ein Task *Daten transformieren* eingefügt werden, der die Datenzeilen aus der ASCII-Datei *Actual_S01.asc* in die SQL Server-Tabelle Actual_S01 transformieren soll. Gehen Sie wie folgt vor:

1. Die Spaltenpositionen in Tabelle 17.1 entsprechen denen im Spaltenplan für die Datei *S01MMJJ.asc*, wie er oben in Tabelle 3.1 angegeben ist. Die Positionen hier sind lediglich jeweils um die Zahl 8 erhöht, weil am Anfang jeder Datenzeile der Monatserster des Aktualisierungsmonats im Format JJJJMMTT eingefügt wurde.

- ▶ Ordnen Sie die beiden DTS-Verbindungen im Fenster des DTS-Designers so an, dass das Datenziel rechts von der Datenquelle positioniert ist.
- ▶ Markieren Sie zuerst die Datenquelle und dann, mit gedrückter Strg-Taste, das Datenziel, und klicken Sie dann in der Symbolleiste *Task* auf den Task *Daten transformieren*. Daraufhin wird der Task als Pfeillinie zwischen den beiden DTS-Verbindungen eingefügt.
- ▶ Doppelklicken Sie auf die Pfeillinie, um das Eigenschaftenfenster für den Task zu öffnen. Geben Sie auf der Registerkarte *Quelle* als Bezeichnung »Transformation nach Actual_S01« ein.
- ▶ Wählen Sie auf der Registerkarte *Ziel* den Tabellennamen Actual_S01.
- ▶ Aktivieren Sie die Registerkarte *Transformationen*. Wählen Sie dann zunächst im Dialogfeld *Transformationen überprüfen* die Option *Ungültige Transformationen entfernen*, und bestätigen Sie das Dialogfeld mit OK. Mit diesem Vorgang hat es die folgende Bewandtnis: Der DTS-Designer legt beim Einfügen des Tasks *Daten transformieren* automatisch die erste verfügbare Tabelle in der Zieldatenbank als Zieltabelle fest und erstellt auf dieser Basis automatisch Transformationen zwischen den Feldern der Quelldatei und der Zieltabelle. Wenn die Zieltabelle geändert wird, werden auch die Transformationen hinfällig. Daher wird in diesem Fall das Dialogfeld *Transformationen überprüfen* angezeigt.
- ▶ Klicken Sie auf der Registerkarte *Transformationen* auf *Alle auswählen* und dann auf *Alle löschen*, um alle automatisch vorgeschlagenen Transformationen zu löschen, denn diese entsprechen nicht unseren Absichten.

Transformationen definieren

Für jedes einzelne der zehn Zielfelder in der Tabelle Actual_S01 muss auf der Registerkarte *Transformationen* eine eigene Transformation definiert werden. Dabei gelten die Zuordnungen, wie sie in Tabelle 17.2 angegeben sind.

Quellfeld	Zielfeld	Transformationstyp
Col001	DW_Eintrittsdatum	<i>ActiveX Script</i>
Col002		
Col003		

Tabelle 17.2 Transformationen für den Task »Transformation nach Actual_S01«

Quellfeld	Zielfeld	Transformationstyp
Col004	ohne Zielfeld	
Col005	PersNr	<i>Copy Column</i>
Col006	Nachname	<i>Trim String</i>
Col007	Vorname	<i>Trim String</i>
Col008	Geburtsdatum	<i>ActiveX Script</i>
Col009		
Col010		
Col011	Eintrittsdatum	<i>ActiveX Script</i>
Col012		
Col013		
Col014	Strasse	<i>Trim String</i>
Col015	PLZ	<i>Copy Column</i>
Col016	Ort	<i>Trim String</i>
Col017	Sex	<i>ActiveX Script</i>
Col018	Ohne Zielfeld	

Tabelle 17.2 Transformationen für den Task »Transformation nach Actual_S01« (Forts.)

Transformation DW_Eintrittsdatum

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col001, Col002 und Col003 und das Zielfeld DW_Eintrittsdatum (Eintrittsdatum in das Data Warehouse), und klicken Sie auf die Schaltfläche *Neu*.
- ▶ Wählen Sie im Dialogfeld *Neue Transformation erstellen ActiveX Script*, und bestätigen Sie mit OK.
- ▶ Klicken Sie im Dialogfeld *Transformationsoptionen auf Eigenschaften...* Dann wird der Editor für ActiveX-Skript angezeigt. Ersetzen Sie die erste Codezeile innerhalb der Function Main so, wie dies im folgenden Listing der geänderten Prozedur wiedergegeben ist. Zeilenumbrüche sind im Editor von ActiveX-Skript in der selben Weise möglich wie bei Visual Basic bzw. VBA.

```
Function Main()
    DTSDestination("DW_Eintrittsdatum") = _
        DateSerial(DTSSource("Col001"), _
```

```

DTSSource("Col002"),DTSSource("Col003"))
Main = DTSTransformStat_OK
End Function

```

Mit dieser Zuweisung werden die drei textlichen Informationen für Jahr, Monat und Tag in einen Datumswert transformiert.

- ▶ Klicken Sie zunächst auf die Schaltfläche *Analysieren*, um den geänderten Code syntaktisch überprüfen zu lassen.
- ▶ Klicken Sie nach erfolgreicher Syntaxüberprüfung auf die Schaltfläche *Testen*. Dann wird die bearbeitete Transformation mit den originalen Daten der Quelldatei ausgeführt, und das Ergebnis wird in eine temporäre Zieltabelle geschrieben, vgl. das Dialogfeld *Transformation testen*, das in Abbildung 17.4 wiedergegeben ist.

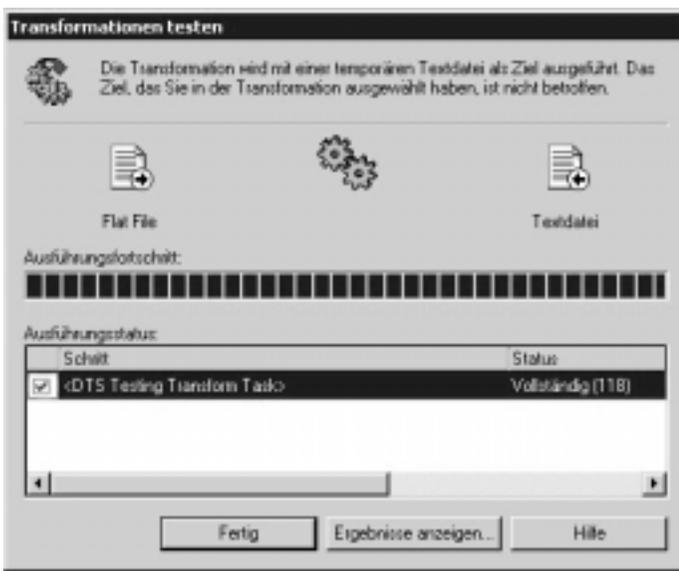


Abbildung 17.4 Dialogfeld Transformationen testen, wiedergegeben für den Test von 118 Datensätzen der Tabelle Actual_S01.asc

- ▶ Klicken Sie im Dialogfeld *Transformation testen* auf *Ergebnisse anzeigen*. Dann werden die Werte des Feldes, in das in der Zieltabelle transformiert wird, für alle Datensätze angezeigt. Im vorliegenden Fall müssen es lauter gleiche Datumswerte sein, die den Monatsersten der aktuellen Datei *Actual_S01* wiedergeben.

- ▶ Bestätigen Sie schließlich nach erfolgreichem Test das Dialogfeld *Transformation testen*, anschließend den Editor für ActiveX-Skript und dann das Dialogfeld *Transformationsoptionen*. Im Eigenschaftfenster für den Task wird jetzt die Transformation von den Feldern Col001, Col002 und Col003 zum Feld DW_Eintrittsdatum durch Verbindungslinien angezeigt und als Typ *ActiveX Script* ausgewiesen.

Transformation Geburtsdatum

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col008, Col009 und Col010 und das Zielfeld Geburtsdatum, und klicken Sie auf die Schaltfläche *Neu*.
- ▶ Verfahren Sie in den folgenden Schritten analog zum Erstellen der Transformation für das Zielfeld DW_Eintrittsdatum, vgl. oben in diesem Gliederungspunkt den Unterpunkt *Transformation DW_Eintrittsdatum*.

Transformation Eintrittsdatum

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col011, Col012 und Col013 und das Zielfeld Eintrittsdatum, und klicken Sie auf die Schaltfläche *Neu*.
- ▶ Verfahren Sie in den folgenden Schritten analog zum Erstellen der Transformation für das Zielfeld DW_Eintrittsdatum, vgl. oben in diesem Gliederungspunkt den Unterpunkt *Transformation DW_Eintrittsdatum*.

Transformation Sex

- ▶ Markieren Sie auf der Registerkarte *Transformationen* das Quellfeld Col017 und das Zielfeld Sex, und klicken Sie auf die Schaltfläche *Neu*.
- ▶ Verfahren Sie in den folgenden Schritten analog zum Erstellen der Transformation für das Zielfeld DW_Eintrittsdatum, vgl. oben in diesem Gliederungspunkt den Unterpunkt *Transformation DW_Eintrittsdatum*. Ersetzen Sie jedoch die erste Codezeile der Function *Main* durch die folgenden Codezeilen:

```
If DTSSource("Col017") = "01" Then
    DTSDestination("Sex") = "m"
ElseIf DTSSource("Col017") = "02" Then
    DTSDestination("Sex") = "w"
End If
```

Transformation PersNr

- ▶ Markieren Sie auf der Registerkarte *Transformationen* das Quellfeld Col005 und das Zielfeld PersNr, und klicken Sie auf die Schaltfläche *Neu*.
- ▶ Wählen Sie im Dialogfeld *Neue Transformation erstellen* den Transformationstyp *Copy Column*, und bestätigen Sie zweimal mit OK.

Transformation PLZ

Verfahren Sie analog zum Zielfeld PersNr, nur ordnen Sie Col015 dem Zielfeld PLZ zu.

Transformation Nachname, Vorname, Strasse und Ort

Verfahren Sie analog zum Zielfeld PersNr, nur ordnen Sie Quell- und Zielfelder entsprechend den Zuordnungen in Tabelle 17.2 zu, und wählen Sie den Transformationstyp *Trim String*.

Gesamten Task testen

Wenn Sie alle Transformationen erfolgreich definiert haben, sollten Sie den Task als einzelnen Schritt ausführen: Klicken Sie mit der rechten Maustaste auf den Task, und wählen Sie im Kontextmenü den Befehl *Schritt ausführen*.

Workflow für die beiden ersten Tasks einfügen

Verbinden Sie den ersten Task *Actual_S01 löschen* mit dem zweiten Task *Transformation nach Actual_S01* durch den Workflow *Bei Erfolg*. Gehen Sie dazu folgendermaßen vor:



- ▶ Ordnen Sie die Tasks nebeneinander so an, dass der Task *Actual_S01 löschen* in der oberen linken Fensterecke des DTS-Designers positioniert ist und der Task *Transformation nach Actual_S01* mit seinen beiden DTS-Verbindungen rechts daneben.
- ▶ Markieren Sie zunächst den Task *Actual_S01 löschen* und anschließend mit gedrückter Strg-Taste die DTS-Quellverbindung zum Task *Transformation nach Actual_S01*, und wählen Sie dann den Befehl *Workflow · Bei Erfolg*. Daraufhin wird eine grün-weiße Pfeillinie zwischen den beiden Tasks eingefügt.

17.3.3 Task SQL ausführen »Actual_S01 verarbeiten«



Fünf Tabellen für die Personaldaten erstellen

Der Task *SQL ausführen* mit dem Namen *Actual_S01 verarbeiten* setzt voraus, dass in der Datenbank *LocAm2SQL* die Tabelle *PersonalStamm* existiert, wie auch die weiteren Tasks für die Transformation der ASCII-Personaldaten das Vorhandensein der entsprechenden Tabellen in der Datenbank *LocAm2SQL* voraussetzen. Die Definitionsmerkmale für die fünf Personaltabellen *PersonalStamm*, *Normalentgelte*, *Kostenstellen*, *Zulagen* und *Zuschläge* sind oben für die Access-Datenbank *LocAm2.mdb* angegeben worden, vgl. Abschnitt 6.3.1, und dort Tabelle 6.1 bis Tabelle 6.5. Diese Felddefinitionen müssen entsprechend für die fünf Personaltabellen in der SQL Server-Datenbank *LocAm2SQL* übernommen werden, wobei zu beachten ist, dass die Datentypen in Access- und SQL Server-Tabellen nicht identisch sind. Obwohl wir für den unmittelbar folgenden Task eigentlich nur die Tabelle *PersonalStamm* benötigen, sollen aus Vereinfachungsgründen bereits an dieser Stelle alle fünf Personaltabellen in der Datenbank *LocAm2SQL* realisiert werden. Gehen Sie folgendermaßen vor:

- ▶ Erstellen Sie die Tabellen *PersonalStamm*, *Normalentgelte*, *Kostenstellen*, *Zulagen* und *Zuschläge* manuell unter Beachtung der angegebenen Definitionsmerkmale.
- ▶ Alternativ: Kopieren Sie die Tabellen *PersonalStamm*, *Normalentgelte*, *Kostenstellen*, *Zulagen* und *Zuschläge* ohne Daten aus der Datenbank *LocAm1SQL*, indem Sie einen DTS-Task *SQL Server-Objekte kopieren* ausführen.² Aktivieren Sie zu diesem Zweck im Eigenschaftenfenster des Tasks auf der Registerkarte *Kopieren* die Optionen *Alle abhängigen Objekte einschließen* und *Erweiterte Eigenschaften einbeziehen*, deaktivieren Sie die Optionen *Daten kopieren* und *Alle Objekte einbeziehen*, klicken Sie auf die *Schaltfläche Objekte auswählen*, und aktivieren Sie im Dialogfeld *Objekte auswählen* die Kontrollkästchen der fünf Tabellen. Mit diesen Optionen werden mit den fünf Tabellen zugleich die zwischen der Mastertabelle *PersonalStamm* und den anderen vier Tabellen definierten 1:n-Beziehungen kopiert.
- ▶ Bestätigen Sie alle Dialogfelder des Tasks, und führen Sie diesen als einzelnen Schritt aus.

2. Dieser Task wurde oben ausführlich behandelt, vgl. Abschnitt 15.4.3. Beachten Sie auch den Tipp, der oben für das Kopieren von SQL Server-Objekten mit dem entsprechenden DTS-Task gegeben wurde, vgl. Abschnitt 17.3, Unterpunkt *Tipp*.

DTS-Task SQL ausführen »Actual_S01 verarbeiten« erstellen

Mit diesem Task soll die Aufgabe gelöst werden, die Datensätze in der Tabelle PersonalStamm mit den Datensätzen in Actual_S01 zu aktualisieren oder mit INSERT zu ergänzen. Für diesen wie auch viele nachfolgende DTS-Tasks vom Typ *SQL ausführen* ist es stets erforderlich, mehr oder minder umfangreiche SQL-Skripte im Eigenschaftfenster des Tasks einzufügen. Bis auf dieses erste Mal werde ich im Weiteren darauf verzichten, das jeweilige SQL-Skript vollständig im Text dieses Buches wiederzugeben, und zwar aus folgenden Gründen: Zum einen würde die Wiedergabe unverhältnismäßig viel Raum beanspruchen. Zum anderen gehe ich davon aus, dass Sie für die Realisierung der entsprechenden Tasks nicht den gesamten SQL-Code abschreiben möchten, da es die Alternative gibt, ihn aus .sql-Dateien von der Begleit-CD in das Eigenschaftfenster des Tasks einzufügen. Da diese Dateien reinen Text enthalten, können Sie sie darüber hinaus mit jedem Texteditor öffnen und auf diese Weise den Code analysieren, gegebenenfalls auch drucken.



Der SQL-Code für die vorliegende Aufgabe lautet folgendermaßen:

```
INSERT INTO PersonalStamm
    SELECT * FROM Actual_S01
    WHERE NOT EXISTS
        (SELECT * FROM PersonalStamm
         WHERE Actual_S01.PersNr = PersonalStamm.PersNr)
GO
UPDATE PersonalStamm
    SET
        PersNr = a.PersNr,
        Nachname = a.Nachname,
        Vorname = a.Vorname,
        Geburtsdatum = a.Geburtsdatum,
        Eintrittsdatum = a.Eintrittsdatum,
        Strasse = a.Strasse,
        PLZ = a.PLZ,
        Ort = a.Ort,
        Sex = a.Sex
    FROM Actual_S01 a
    WHERE a.PersNr = PersonalStamm.PersNr
GO
```

Listing 17.3 SQL-Skript für den SQL ausführen-Task »Actual_S01 verarbeiten«

Mit diesen Anweisungen werden zunächst in die Tabelle PersonalStamm neue Datensätze aus Tabelle Actual_S01 eingefügt, soweit Datensätze mit den entsprechenden Personalnummern in Tabelle PersonalStamm nicht existieren. Dann wird mit allen Datensätzen aus Tabelle Actual_S01 ein UPDATE an den Datensätzen in Tabelle PersonalStamm ausgeführt.

Gehen Sie folgendermaßen vor, um den Task einzufügen:

- ▶ Klicken Sie in der Symbolleiste *Task* auf das Symbol *SQL ausführen*.
- ▶ Geben Sie im Eigenschaftenfenster des Tasks als Beschreibung den Namen »Actual_S01 verarbeiten« ein.
- ▶ Klicken Sie im Eigenschaftenfenster auf die Schaltfläche *Durchsuchen*, wählen Sie im Dialogfeld *Datei auswählen* von der Begleit-CD die Datei `\DW\SQL\SQLSkripte\Actual_S01.sql`, bestätigen Sie mit *Öffnen*, und bestätigen Sie das Eigenschaftenfenster mit OK.
- ▶ Testen Sie, ob der Task fehlerfrei ausgeführt wird, indem Sie ihn als einzelnen Schritt ausführen (Kontextmenü). Die Tabelle PersonalStamm muss nach diesem Schritt 142 Datensätze enthalten. Prüfen Sie außerdem, ob die *ActiveX Script*-Transformationen zu korrekten Ergebnissen geführt haben, indem Sie beispielsweise die verschiedenen Datumswerte kontrollieren.

Workflow einfügen



Fügen Sie vom Task *Transformation nach Actual_S01* zum Task *Actual_S01 verarbeiten* einen Workflow *Bei Erfolg* ein: Markieren Sie zunächst die Zielverbindung des ersten Tasks und dann, mit gedrückter Strg-Taste, den zweiten Task, und wählen Sie dann den Befehl *Workflow • Bei Erfolg*.

DTS-Paket testen

Testen Sie das DTS-Paket in seinem bisherigen Entwicklungsstand, der im Fenster des DTS-Designers so aussieht, wie in Abbildung 17.5 wiedergegeben.

Klicken Sie in der Standardsymbolleiste des DTS-Designers auf *Ausführen*. Dann wird der Ausführungsfortschritt der einzelnen Schritte in einem Dialogfeld simultan mit der Ausführung des DTS-Pakets angezeigt.

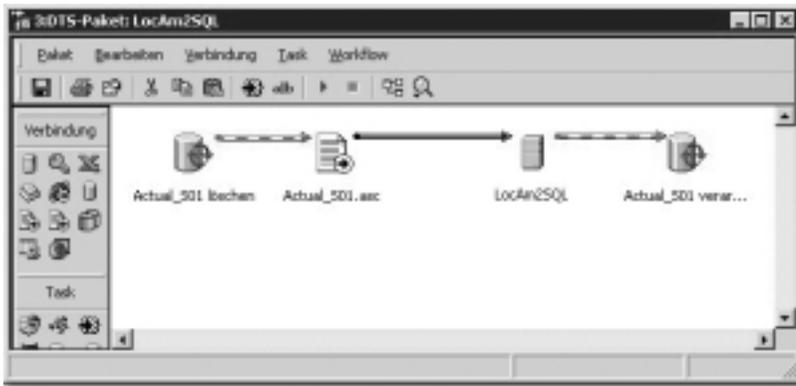


Abbildung 17.5 Fenster des DTS-Designers nach Fertigstellung der Transformationen von Actual_S01.asc

DTS-Paket als LocAm2SQL speichern

Speichern Sie das DTS-Paket unter dem Namen *LocAm2SQL* auf dem SQL Server. Beachten Sie, dass jede weitere Speicherung als weitere Version des Pakets abgelegt wird. Dies kann im Laufe der Zeit vorteilhaft sein, weil Sie den Entwicklungsfortschritt auch nachträglich verfolgen und gegebenenfalls auf frühere (weil z. B. fehlerfreie) Versionen zurückgreifen können. Auf der anderen Seite bläht sich das DTS-Paket dadurch immer weiter auf. Aus diesem Grunde sollten Sie vielleicht gelegentlich alte Versionen aus dem Paket löschen, vgl. dazu oben Abschnitt 15.6.3.



17.4 Realisierung der Transformation von Actual_P01.asc

Um diese Transformation zu bewerkstelligen, sind dieselben drei DTS-Tasks zu erstellen wie im vorangehenden Abschnitt für die Transformation von *Actual_S01.asc*, wenngleich mit anderen Inhalten:

► SQL ausführen: Actual_P01 löschen

Im ersten Schritt werden die gegebenenfalls vorhandenen Datensätze in der SQL Server-Tabelle *Actual_P01* mit dem DTS-Task *SQL ausführen* gelöscht.

► Daten transformieren: Transformation nach Actual_P01

Im zweiten Schritt werden die Datenzeilen der ASCII-Datei *Actual_P01.asc* mit dem DTS-Task *Daten transformieren* entsprechend dem Spaltenplan in die temporäre SQL Server-Tabelle *Actual_P01* transformiert. Dabei wird jede Datenzeile übernommen.

► **SQL ausführen: Actual_P01 verarbeiten**

Im dritten Schritt werden die Datensätze der temporären Tabelle Actual_P01 mit dem DTS-Task *SQL ausführen* in die Tabellen Kostenstellen, Normalentgelte und Zulagen transformiert.

SQL Server-Tabelle Actual_P01 erstellen

Die Tabelle Actual_P01 muss zunächst einmal in der Datenbank *LocAm2SQL* erstellt werden. Sie weist die in Abbildung 17.6 wiedergegebenen Felddefinitionen auf. Beachten Sie, dass dabei die Felder *strKostenstelle*, *strKostenstellengewicht*, *ZulagenBetrag* und *ZulagenSchlüssel* Wiederholungsfelder enthalten und daher jeweils sehr große Feldlängen aufweisen.

	Spaltenname	Datentyp	Größe	NULL zulassen
	Monatsnr	smallint	4	✓
	PersNr	int	4	✓
	strKostenstelle	nchar	360	✓
	strKostenstellengewicht	nchar	360	✓
	ArbeitgeberBrutto	money	8	✓
	Teilzeitfaktor	decimal	9	✓
	Sonderentgelt	money	8	✓
	DienstortSchlüssel	int	4	✓
	ZulagenBetrag	char	200	✓
	ZulagenSchlüssel	char	140	✓

Abbildung 17.6 Felddefinitionen der Tabelle Actual_P01



- Erstellen Sie die Tabelle Actual_P01 in der Datenbank *LocAm2SQL* manuell entsprechend den Definitionsmerkmalen aus Abbildung 17.6.
- Alternativ: Kopieren Sie die Tabellenstruktur aus der Datenbank *LocAm1SQL* mit dem DTS-Task *SQL Server-Objekte kopieren* (vgl. oben Abschnitt 15.4.3).

17.4.1 Task SQL ausführen »Actual_P01 löschen«



Dieser Task soll alle Datensätze in der Tabelle Actual_P01 löschen, damit mit dem folgenden Task *Daten transformieren* die aktuellen Datensätze aus der ASCII-Datei *Actual_P01.asc* in die leere Tabelle eingefügt werden können. Gehen Sie folgendermaßen vor, um den Task einzurichten:

- Fügen Sie im Fenster des DTS-Designers einen Task *SQL ausführen* ein.
- Geben Sie im Eigenschaftenfenster dieses Tasks als Beschreibung den Text »Actual_P01 löschen« an, wählen Sie als vorhandene Verbindung *LocAm2SQL*, und geben Sie als SQL-Anweisung den folgenden Code ein
`DELETE FROM Actual_P01`

- ▶ Bestätigen Sie mit OK, ziehen Sie den Task an den linken Rand des Fensters unterhalb des entsprechenden Tasks »Actual_SO1 löschen«, und führen Sie einen Test aus: Klicken Sie mit der rechten Maustaste auf den Task, und wählen Sie im Kontextmenü den Befehl *Schritt ausführen*.

17.4.2 Task Daten transformieren »Transformation nach Actual_PO1«

DTS-Verbindung zu Actual_PO1.asc erstellen

Bevor der Task *Daten transformieren* eingerichtet werden kann, muss eine DTS-Verbindung zur Datenquelle *Actual_PO1.asc* erstellt werden. Gehen Sie folgendermaßen vor:



- ▶ Fügen Sie in das Fenster des DTS-Designers eine neue Verbindung vom Typ *Text File (Source)* ein.
- ▶ Geben Sie im Eigenschaftenfenster dieser Verbindung als Bezeichnung »Actual_PO1.asc« an, und wählen Sie als Dateinamen *v:\Input\Actual_PO1.asc*.
- ▶ Klicken Sie auf die Schaltfläche *Eigenschaften...* Die voreingestellten Angaben zum Dateiformat im Dialogfeld *Textdateieigenschaften* sind in Ordnung, klicken Sie daher auf die Schaltfläche *Weiter >*.
- ▶ Legen Sie die Spaltenpositionen nach folgendem Plan fest. Achten Sie dabei sorgfältig darauf, die vielen für unseren Plan fehlerhaften Spaltenpositionen, die automatisch vorgeschlagen wurden, zu löschen.

FeldNr	Endposition
1	4
2	6
3	8
4	14
5	19
6	379
7	659
8	669
9	677

Tabelle 17.3 Spaltenpositionen für Textdatei Actual_PO1.asc

FeldNr	Endposition
10	687
11	689
12	767
13	967
14	1108

Tabelle 17.3 Spaltenpositionen für Textdatei Actual_P01.asc (Forts.)

- ▶ Klicken Sie auf die Schaltfläche *Fertig stellen* und anschließend auf OK.
- ▶ Positionieren Sie die neue Verbindung rechts neben dem Task *Actual_P01 löschen*.



Neue Verbindung zu LocAm2SQL einfügen

Für den Task *Daten transformieren* benötigen wir wiederum auch eine Verbindung zur SQL Server-Datenbank *LocAm2SQL*:

- ▶ Klicken Sie in der Symbolleiste *Verbindung* auf *Microsoft OLE DB Provider for SQL Server*.
- ▶ Aktivieren Sie im Dialogfeld *Verbindungseigenschaften* die Option *Vorhandene Verbindung*, wählen Sie dazu die Verbindung *LocAm2SQL*, und bestätigen Sie mit OK.
- ▶ Positionieren Sie die neue Verbindung rechts neben der Verbindung *Actual_P01.asc*.



Task Daten transformieren »Transformation nach Actual_P01« erstellen

Dieser Task soll die Datenzeilen aus der ASCII-Datei *Actual_P01.asc* in die SQL Server-Tabelle *Actual_P01* transformieren. Gehen Sie wie folgt vor:

- ▶ Markieren Sie zuerst die Datenquelle und dann, mit gedrückter Strg-Taste, das Datenziel, und klicken Sie dann in der Symbolleiste *Task* auf den Task *Daten transformieren*.
- ▶ Doppelklicken Sie auf die Pfeillinie, die den Task symbolisiert, um sein Eigenschaftenfenster zu öffnen. Geben Sie auf der Registerkarte *Quelle* als Bezeichnung »Transformation nach Actual_P01« ein.
- ▶ Wählen Sie auf der Registerkarte *Ziel* den Tabellennamen *Actual_P01*.

- ▶ Aktivieren Sie die Registerkarte *Transformationen*, und klicken Sie auf *Alle auswählen* und dann auf *Alle löschen*, um alle automatisch vorgeschlagenen Transformationen zu löschen.

Transformationen definieren

Für jedes einzelne der zehn Zielfelder in der Tabelle Actual_P01 muss auf der Registerkarte *Transformationen* eine eigene Transformation definiert werden. Dabei gelten die Zuordnungen, wie sie in Tabelle 17.4 angegeben sind.

Quellfeld	Zielfeld	Transformationstyp
Col001	Monatserster	<i>ActiveX Script</i>
Col002		
Col003		
Col004	Ohne Zielfeld	
Col005	PersNr	<i>Copy Column</i>
Col006	strKostenstelle	<i>Copy Column</i>
Col007	strKostenstellengewicht	<i>Copy Column</i>
Col008	ArbeitgeberBrutto	<i>ActiveX Script</i>
Col009	Teilzeitzähler	<i>ActiveX Script</i>
Col010	Sonderentgelt	<i>ActiveX Script</i>
Col011	DienststartSchlüssel	<i>Copy Column</i>
Col012	Ohne Zielfeld	
Col013	ZulagenBetrag	<i>Copy Column</i>
Col014	ZulagenSchlüssel	<i>Copy Column</i>

Tabelle 17.4 Transformationen für den Task »Transformation nach Actual_P01«

Transformation Monatserster

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col001, Col002 und Col003 und das Zielfeld Monatserster, und klicken Sie auf die Schaltfläche *Neu*, wählen Sie den Transformationstyp *ActiveX Script*, und bearbeiten Sie das ActiveX-Skript folgendermaßen: Ersetzen Sie die erste Codezeile in der Function *Main* durch die folgende (hier umbrochene) Codezeile:

```
DTSDestination("Monatserster") = _
    DateSerial(DTSSource("Col001"),DTSSource( _
        "Col002"),DTSSource("Col003"))
```

- ▶ Testen Sie die Transformation, und bestätigen Sie dann zweimal mit OK.

Transformation PersNr, strKostenstelle, strKostenstellengewicht, DienststartSchlüssel, ZulagenBetrag, ZulagenSchlüssel

Die sechs Zielfelder PersNr, strKostenstelle, strKostenstellengewicht, DienststartSchlüssel, ZulagenBetrag und ZulagenSchlüssel sollen alle mit dem Transformationstyp *Copy Column* transformiert werden. Diese Transformationen können Sie in einem Vorgang zusammen definieren. Gehen Sie wie folgt vor:

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col005, Col006, Col007, Col011, Col013 und Col014 sowie die sechs angegebenen Zielfelder.
- ▶ Klicken Sie auf die *Schaltfläche Neu*, und wählen Sie den Transformationstyp *Copy Column*. Nach zweimaliger Bestätigung mit OK werden auf der Registerkarte *Transformationen* die sechs Transformationen durch eine gemeinsame Linie angezeigt.

Transformation ArbeitgeberBrutto, Teilzeitzähler, Sonderentgelt

- ▶ Die drei Zielfelder ArbeitgeberBrutto, Teilzeitzähler und Sonderentgelt sollen alle mit dem Transformationstyp *ActiveX Script* transformiert werden. Diese Transformationen können Sie ebenfalls in einem Vorgang zusammen definieren. Gehen Sie wie folgt vor:
- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col008, Col009 und Col010 und die Zielfelder ArbeitgeberBrutto, Teilzeitzähler und Sonderentgelt, klicken Sie auf die Schaltfläche *Neu*, wählen Sie den Transformationstyp *ActiveX Script*, und ändern Sie im *ActiveX-Skript* die ersten drei Codezeilen in der Funktion Main so, dass die Quellfelder jeweils mit der Funktion CSng in den Datentyp Single konvertiert werden. Die drei Codezeilen müssen dann folgendermaßen aussehen (hier umbrochen):

```
DTSDestination("ArbeitgeberBrutto") = _
    CSng(DTSSource("Col008"))
DTSDestination("Teilzeitzähler") = _
    CSng(DTSSource("Col009"))
```

```
DTSDestination("Sonderentgelt") = _  
CSng(DTSSource("Col010"))
```

Testen Sie die Transformationen, und bestätigen Sie dann alle Dialogfelder mit OK, denn damit ist der Task *Daten transformieren Transformation nach Actual_P01* vollständig definiert.

- ▶ Führen Sie den Task als einzelnen Schritt aus (Kontextmenü), um zu testen, ob er fehlerfrei arbeitet. Dies setzt voraus, dass die Tabelle Actual_P01 leer ist. Falls Sie insoweit unsicher sind, führen Sie sicherheitshalber zuvor den Task *Actual_P01* löschen aus.

17.4.3 Task SQL ausführen »Actual_P01 verarbeiten«

Dieser Task setzt voraus, dass in der Datenbank *LocAm2SQL* die drei Tabellen *Kostenstellen*, *Normalentgelte* und *Zulagen* definiert sind. Dies ist bereits zu Beginn des obigen Punktes 17.3.3, realisiert worden.

DTS-Task SQL ausführen »Actual_P01 verarbeiten« erstellen



Mit diesem Task soll die Aufgabe gelöst werden, die Datensätze in den Tabellen *Kostenstellen*, *Normalentgelte* und *Zulagen* mit den Datensätzen in *Actual_P01* zu aktualisieren bzw. mit INSERT zu ergänzen. Das SQL-Skript für den Task finden Sie auf der Begleit-CD im Pfad `\DW\SQL\SQL-Skripte\Actual_P01.sql`.

Gehen Sie folgendermaßen vor, um den Task einzufügen:

- ▶ Klicken Sie in der Symbolleiste *Task* auf das Symbol *SQL ausführen*.
- ▶ Geben Sie im Eigenschaftenfenster des Tasks als Beschreibung den Namen »Actual_P01 verarbeiten« ein, und wählen Sie die vorhandene Verbindung *LocAm2SQL*.
- ▶ Klicken Sie im Eigenschaftenfenster auf die Schaltfläche *Durchsuchen*, wählen Sie im Dialogfeld *Datei auswählen* von der Begleit-CD die Datei `\DW\SQL\SQLSkripte\Actual_P01.sql`, bestätigen Sie mit *Öffnen*, und bestätigen Sie das Eigenschaftenfenster mit OK.
- ▶ Testen Sie, ob der Task fehlerfrei ausgeführt wird, indem Sie ihn als einzelnen Schritt ausführen (Kontextmenü). Die Tabellen *Kostenstellen*, *Normalentgelte* und *Zulagen* müssen nach diesem Schritt 126 bzw. 100 bzw. 136 Datensätze für den Monatsersten 01.02.2000 enthalten.

Workflows einfügen



- ▶ Fügen Sie je einen Workflow *Bei Erfolg* zwischen den folgenden Tasks ein:
 - ▶ zwischen *Actual_S01 verarbeiten* und *Actual_PO1 löschen*
 - ▶ zwischen *Actual_PO1 löschen* und *Transformation nach Actual_PO1*
 - ▶ zwischen *Transformation nach Actual_PO1* und *Actual_PO1 verarbeiten*

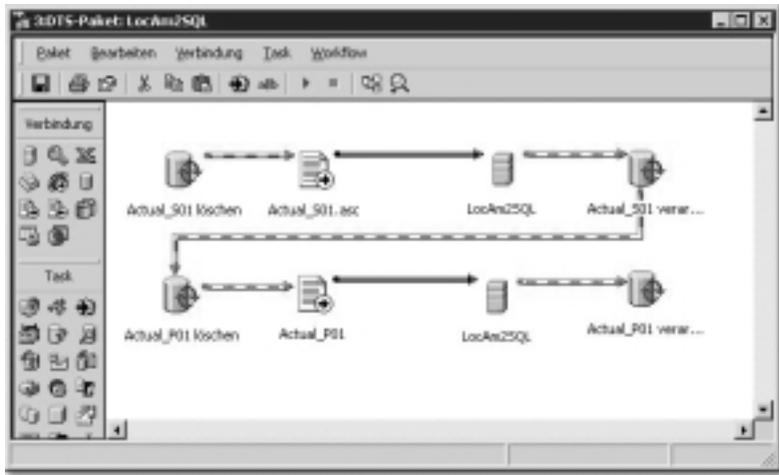


Abbildung 17.7 Fenster des DTS-Designers nach Fertigstellung der Transformationen von *Actual_PO1.asc*

- ▶ Ziehen Sie die Tasks und Workflows mit der Maus in Positionen, dass sich ungefähr das Aussehen von Abbildung 17.7 ergibt.
- ▶ Führen Sie das gesamte DTS-Paket in seinem bisherigen Entwicklungszustand aus, indem Sie auf das Symbol *Ausführen* klicken.
- ▶ Speichern Sie das DTS-Paket.

17.5 Realisierung der Transformation von *Actual_P02.asc*

Um diese Transformation zu bewerkstelligen, sind dieselben drei DTS-Tasks zu erstellen wie in den beiden vorangehenden Punkten für die Transformationen von *Actual_S01.asc* bzw. *Actual_PO1.asc*, wengleich mit anderen Inhalten:

► **SQL ausführen: Actual_P02 löschen**

Im ersten Schritt werden die gegebenenfalls vorhandenen Datensätze in der SQL Server-Tabelle Actual_P02 mit dem DTS-Task *SQL ausführen* gelöscht.

► **Daten transformieren: Transformation nach Actual_P02**

Im zweiten Schritt werden die Datenzeilen der ASCII-Datei *Actual_P02.asc* mit dem DTS-Task *Daten transformieren* entsprechend dem Spaltenplan in die temporäre SQL Server-Tabelle Actual_P02 transformiert. Dabei wird jede Datenzeile übernommen.

► **SQL ausführen: Actual_P021 verarbeiten**

Im dritten Schritt werden die Datensätze der temporären Tabelle Actual_P02 mit dem DTS-Task *SQL ausführen* in die Tabellen Normalentgelte und Zuschläge transformiert.

Die Tabelle Actual_P02 muss zunächst einmal in der Datenbank *LocAm2SQL* erstellt werden. Sie weist die in Abbildung 17.8 wiedergegebenen Felddefinitionen auf. Beachten Sie, dass dabei die Felder *strZuschlagsbetrag*, *strZuschlagSchlüssel* und *strÜberstunden* Wiederholungsfelder enthalten und jeweils Character-Datentypen haben mit Größen, die den jeweils in der ASCII-Datei belegten Spalten entsprechen.

SQL Server-Tabelle Actual_P02 erstellen

	Spaltenname	Datentyp	Größe	NULL zulassen
	Monatsster	smalldatetime	4	✓
	Periode	int	4	✓
	UnterbrechungsgrundSchlüssel	nvarchar	2	✓
	Unterbrechungsende	smalldatetime	4	✓
	Tarifgruppe	varchar	3	✓
	strZuschlagsbetrag	varchar	300	✓
	strZuschlagSchlüssel	varchar	210	✓
	strÜberstunden	varchar	210	✓

Abbildung 17.8 Felddefinitionen der Tabelle Actual_P021

► Erstellen Sie die Tabelle Actual_P02 in der Datenbank *LocAm2SQL* manuell entsprechend den Definitionsmerkmalen aus Abbildung 17.8.



► Alternativ: Kopieren Sie die Tabellenstruktur aus der Datenbank *LocAm1SQL* mit dem DTS-Task *SQL Server-Objekte kopieren* (vgl. oben Abschnitt 15.4.3).

17.5.1 Task SQL ausführen »Actual_P02 löschen«

Dieser Task soll alle Datensätze in der Tabelle Actual_P02 löschen, damit mit dem folgenden Task *Daten transformieren* die aktuellen Datensätze aus der ASCII-Datei *Actual_P02.asc* in die leere Tabelle eingefügt werden können. Gehen Sie folgendermaßen vor, um den Task einzurichten:



- ▶ Fügen Sie im Fenster des DTS-Designers einen Task *SQL ausführen* ein.
- ▶ Geben Sie im Eigenschaftenfenster dieses Tasks als Beschreibung den Text »Actual_PO2 löschen« an, wählen Sie als vorhandene Verbindung *LocAm2SQL*, und geben Sie als SQL-Anweisung den folgenden Code ein:

```
DELETE FROM Actual_PO2
```

- ▶ Bestätigen Sie mit OK, ziehen Sie den Task an den linken Rand des Fensters unterhalb des entsprechenden Tasks *Actual_PO1 löschen*, und führen Sie einen Test aus: Klicken Sie mit der rechten Maustaste auf den Task, und wählen Sie im Kontextmenü den Befehl *Schritt ausführen*.

17.5.2 Task Daten transformieren »Transformation nach Actual_PO2«



DTS-Verbindung zu Actual_PO2.asc erstellen

Bevor der Task *Daten transformieren* eingerichtet werden kann, muss eine *DTS-Verbindung* zur Datenquelle *Actual_PO2.asc* erstellt werden. Gehen Sie folgendermaßen vor:

- ▶ Fügen Sie in das Fenster des DTS-Designers eine neue Verbindung vom Typ *Text File (Source)* ein.
- ▶ Geben Sie im Eigenschaftenfenster dieser Verbindung als Bezeichnung »Actual_PO2.asc« an, und wählen Sie als Dateinamen *v:\Input\Actual_PO2.asc*.
- ▶ Klicken Sie auf die Schaltfläche *Eigenschaften...* Die voreingestellten Angaben zum Dateiformat im Dialogfeld *Textdateieigenschaften* sind in Ordnung, klicken Sie daher auf die Schaltfläche *Weiter >*.
- ▶ Legen Sie die Spaltenpositionen nach folgendem Plan fest. Achten Sie dabei sorgfältig darauf, die vielen für unseren Plan fehlerhaften Spaltenpositionen, die automatisch vorgeschlagen wurden, zu löschen.

FeldNr	Endposition
1	4
2	6
3	8
4	14

Tabelle 17.5 Spaltenpositionen für Textdatei Actual_PO2.asc

FeldNr	Endposition
5	19
6	21
7	25
8	27
9	29
10	32
11	332
12	542
13	752

Tabelle 17.5 Spaltenpositionen für Textdatei Actual_P02.asc (Forts.)

- ▶ Klicken Sie auf die Schaltfläche *Fertig stellen* und anschließend auf OK.
- ▶ Positionieren Sie die neue Verbindung rechts neben dem Task *Actual_P02 löschen*.

Neue Verbindung zu LocAm2SQL einfügen

Für den Task *Daten transformieren* benötigen wir wiederum auch eine Verbindung zur SQL Server-Datenbank *LocAm2SQL*:



- ▶ Klicken Sie in der Symbolleiste *Verbindung* auf *Microsoft OLE DB Provider for SQL Server*.
- ▶ Aktivieren Sie im Dialogfeld *Verbindungseigenschaften* die Option *Vorhandene Verbindung*, wählen Sie dazu die Verbindung *LocAm2SQL*, und bestätigen Sie mit OK.
- ▶ Positionieren Sie die neue Verbindung rechts neben der Verbindung *Actual_P02.asc*.

Dieser Task soll die Datenzeilen aus der ASCII-Datei *Actual_P02.asc* in die SQL Server-Tabelle *Actual_P02* transformieren. Gehen Sie wie folgt vor:

- ▶ Markieren Sie zuerst die Datenquelle und dann, mit gedrückter Strg-Taste, das Datenziel, und klicken Sie dann in der Symbolleiste *Task* auf den Task *Daten transformieren*.
- ▶ Doppelklicken Sie auf die Pfeillinie, die den Task symbolisiert, um sein Eigenschaftenfenster zu öffnen. Geben Sie auf der Registerkarte *Quelle* als Bezeichnung »Transformation nach Actual_P02« ein.

Task Daten transformieren »Transformation nach Actual_P02« erstellen

- ▶ Wählen Sie auf der Registerkarte *Ziel* den Tabellennamen Actual_P02.
- ▶ Aktivieren Sie die Registerkarte *Transformationen* und klicken Sie auf *Alle auswählen* und dann auf *Alle löschen*, um alle automatisch vorgeschlagenen Transformationen zu löschen.

Transformationen definieren

Für die Transformationen Zielfelder in der Tabelle Actual_P02 gelten die Zuordnungen, wie sie in Tabelle 17.6 angegeben sind.

Quellfeld	Zielfeld	Transformationstyp
Col001	Monatserster	<i>ActiveX Script</i>
Col002		
Col003		
Col004	ohne Zielfeld	
Col005	PersNr	<i>Copy Column</i>
Col006	UnterbrechungsgrundSchlüssel	<i>Copy Column</i>
Col007	Unterbrechungsende	<i>ActiveX Script</i>
Col008		
Col009		
Col010	Tarifgruppe	<i>Copy Column</i>
Col011	strZuschlagsbetrag	<i>Copy Column</i>
Col012	strZuschlagSchlüssel	<i>Copy Column</i>
Col013	strÜberstunden	<i>Copy Column</i>
Col014	ohne Zielfeld	

Tabelle 17.6 Transformationen für den Task »Transformation nach Actual_P02«

Transformation Monatserster

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col001, Col002 und Col003 und das Zielfeld Monatserster, und klicken Sie auf die Schaltfläche *Neu*, wählen Sie den Transformationstyp *ActiveX Script*, und bearbeiten Sie das ActiveX-Skript folgendermaßen: Ersetzen Sie die erste Codezeile in der *Function Main* durch die folgende (hier umbrochene) Codezeile:

```
DTSDestination("Monatserster") = _
    DateSerial(DTSSource("Col001"),DTSSource( _
        "Col002"),DTSSource("Col003"))
```

- ▶ Testen Sie die Transformation, und bestätigen Sie dann zweimal mit OK.

Transformation Unterbrechungsende

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col007, Col008 und Col009 und das Zielfeld Monatserster, und klicken Sie auf die *Schaltfläche Neu*, wählen Sie den Transformationstyp *ActiveX Script*, und bearbeiten Sie das ActiveX-Skript folgendermaßen: Ersetzen Sie die erste Codezeile in der Function *Main* durch die folgenden Codezeilen:

```
If Not DTSSource("Col007") = Space(4) Then
    DTSDestination("Unterbrechungsende") = _
        DateSerial(DTSSource("Col007"), _
            DTSSource("Col008"),DTSSource("Col009"))
Else
    DTSDestination("Unterbrechungsende") = Null
End If
```

Die Prüfung auf vier Leerzeichen für Col007 ist erforderlich, weil nicht jeder Datensatz einen Datumseintrag für dieses Feld aufweist, das dann mit Leerzeichen besetzt ist.

- ▶ Testen Sie die Transformation, und bestätigen Sie dann zweimal mit OK.

Transformation PersNr, UnterbrechungsgrundSchlüssel, Tarifgruppe, strZuschlagsbetrag, strZuschlagSchlüssel, strÜberstunden

Die sechs Zielfelder *PersNr*, *UnterbrechungsgrundSchlüssel*, *Tarifgruppe*, *strZuschlagsbetrag*, *strZuschlagSchlüssel* und *strÜberstunden* sollen alle mit dem Transformationstyp *Copy Column* transformiert werden. Diese Transformationen können Sie in einem Vorgang zusammen definieren. Gehen Sie wie folgt vor:

- ▶ Markieren Sie auf der Registerkarte *Transformationen* die Quellfelder Col005, Col006, Col010, Col011, Col012 und Col013 sowie die sechs angegebenen Zielfelder.

- ▶ Klicken Sie auf die Schaltfläche *Neu*, und wählen Sie den Transformationsstyp *Copy Column*. Nach zweimaliger Bestätigung mit OK werden auf der Registerkarte *Transformationen* die sechs Transformationen durch eine gemeinsame Linie angezeigt. Bestätigen Sie das Eigenschaftfenster mit OK, weil nunmehr alle Transformationen dieses Tasks definiert sind.
- ▶ Führen Sie den Task als einzelnen Schritt aus (Kontextmenü), um zu testen, ob er fehlerfrei arbeitet. Dies setzt voraus, dass die Tabelle *Actual_PO2* leer ist. Falls Sie insoweit unsicher sind, führen Sie sicherheitshalber zuvor den Task *Actual_PO2* löschen aus.

17.5.3 Task SQL ausführen »Actual_Po2 verarbeiten«

Dieser Task setzt voraus, dass in der Datenbank *LocAm2SQL* die beiden Tabellen *Normalentgelte* und *Zuschläge* definiert sind. Dies ist bereits zu Beginn des obigen Punktes 17.3.3, realisiert worden.



DTS-Task SQL ausführen »Actual_Po2 verarbeiten« erstellen

Mit diesem Task soll die Aufgabe gelöst werden, die Datensätze in den Tabellen *Normalentgelte* und *Zuschläge* mit den Datensätzen in *Actual_PO2* zu aktualisieren bzw. mit *INSERT* zu ergänzen. Das SQL-Skript für den Task finden Sie auf der Begleit-CD im Pfad `\DW\SQL\SQLSkripte\Actual_PO2.sql`.

Obwohl ich den SQL-Code im Allgemeinen nicht weiter erläutern möchte, sei an dieser Stelle, quasi stellvertretend für weitere entsprechende Datenzugriffe, auf einen Sachverhalt verwiesen, der sich deutlich von den Transformationsprozessen in der Access-Lösung unterscheidet: Im Rahmen des in Rede stehenden Tasks müssen u.a. in den drei Zieltabellen die Datensätze mit dem aktuellen Monatsersten gelöscht werden, bevor ein *INSERT* ausgeführt wird. Im Rahmen der Access-Lösung wurde der aktuelle Monatserste als Argumentwert an die VBA-Prozedur übergeben, die sie dann in die SQL-Anweisung zum Löschen einbauen konnte. Hier wird keine Prozedur aufgerufen, der von außen ein Argumentwert übergeben werden könnte. Statt dessen wird der aktuelle Monatserste dem Feld *Monatserster* in der Tabelle *Actual_PO1* entnommen und an die Variable *@Monatserster* übergeben, die dann in einer SQL-Anweisung direkt verwendet werden kann, wie beispielsweise in der folgenden Codezeile

```
DELETE FROM Kostenstellen WHERE
```


- ▶ Ziehen Sie die Tasks und Workflows mit der Maus in Positionen, dass sich ungefähr das Aussehen von Abbildung 17.9 ergibt.
- ▶ Führen Sie das gesamte DTS-Paket in seinem bisherigen Entwicklungszustand aus, indem Sie auf das Symbol *Ausführen* klicken.
- ▶ Speichern Sie das DTS-Paket.