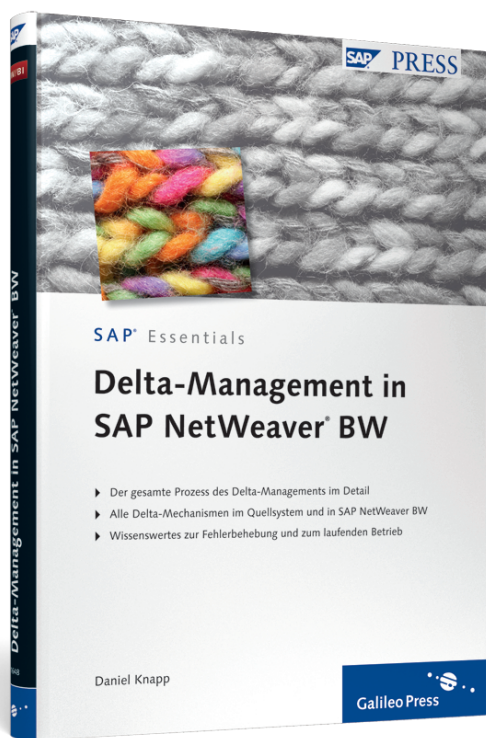


Serge Daniel Knapp

## Delta-Management in SAP NetWeaver® BW



  
Galileo Press

Bonn • Boston

# Inhalt

Vorwort .....	9
<b>1 Einleitung .....</b>	<b>11</b>
1.1 Inhalt dieses Buches .....	13
1.2 Zielgruppen des Buches .....	15
<b>2 Grundlagen .....</b>	<b>17</b>
2.1 Objekte in SAP NetWeaver BW .....	17
2.1.1 InfoProvider und Datenziele .....	17
2.1.2 DataStore-Objekt .....	18
2.1.3 DataSource .....	20
2.1.4 InfoPackage und Datentransferprozess .....	22
2.2 Gründe für den Einsatz von Delta-Verfahren .....	25
2.3 Möglichkeiten der Deltaermittlung .....	26
2.3.1 Echtes Delta .....	26
2.3.2 Pseudo-Delta .....	27
2.4 Delta-Verfahren .....	28
2.4.1 Delta-Typ .....	29
2.4.2 Satztyp .....	31
2.4.3 Serialisierung .....	34
2.5 Bestimmung der Delta-Fähigkeit von Extraktoren .....	36
2.5.1 Tabellen ROOSOURCE, RSDS und RSOLTPSOURCE ...	36
2.5.2 Die Tabelle RODELTAM .....	40
2.6 Auswirkungen von Delta-Verfahren auf SAP NetWeaver BW .....	41
2.6.1 Aggregationsverhalten von InfoProvidern .....	41
2.6.2 Auswirkung der Beleganlage .....	42
2.6.3 Auswirkung der Belegänderung .....	43
2.6.4 Auswirkung der Belegstornierung .....	44
2.6.5 Fazit .....	44
2.7 Zusammenfassung .....	46
<b>3 Delta-Mechanismen im Quellsystem .....</b>	<b>49</b>
3.1 Delta-Queue in der Service-API .....	49
3.1.1 Ablageort für neue und geänderte Datensätze .....	50

3.1.2	Speicherung des aktuellen und letzten Deltas .....	56
3.1.3	Versorgung mehrerer BW-Zielsysteme .....	57
3.2	Logistik-Extraktion .....	58
3.2.1	Logistik-Applikation .....	59
3.2.2	Delta-Initialisierung/Full-Update .....	61
3.2.3	Verbuchungsmethoden .....	63
3.3	Generisches Delta .....	69
3.3.1	Funktionsweise des generischen Deltas .....	70
3.3.2	Sicherheitsintervalle .....	73
3.4	Fortgeschrittene Techniken der Delta-Extraktion .....	76
3.4.1	Delta über Business Transaction Events .....	76
3.4.2	Delta über Funktionsbausteine .....	87
3.4.3	ALE-Fortschreibungszeiger .....	93
3.5	Realtime Data Acquisition .....	97
3.5.1	Kurze Beschreibung der Funktionsweise .....	97
3.5.2	RDA-Fähigkeit von Business Content DataSources .....	98
3.5.3	RDA-Fähigkeit von generischen DataSources .....	99
3.6	Delta-Mechanismen für Flatfiles, DB Connect und UDI .....	100
3.6.1	Definition von DataSources .....	101
3.6.2	Delta-Verfahren im Detail .....	103
3.6.3	Eigenschaften von DB Connect .....	105
3.6.4	Eigenschaften von Universal Data Integration .....	105
3.7	Delta-Mechanismen für Webservices .....	107
3.7.1	Anlegen von DataSources für Webservices .....	107
3.7.2	Delta-Verfahren für Webservices .....	109
3.8	SAP NetWeaver BW als Quellsystem .....	110
3.8.1	Data Mart Interface .....	110
3.8.2	Delta-Verfahren im Data Mart Interface .....	112
3.9	Delta-Mechanismen für SAP BusinessObjects Data Services (SAP NetWeaver BW 7.30) .....	112
3.10	Zusammenfassung .....	114
<b>4</b>	<b>Delta-Verarbeitung im BW-System .....</b>	<b>117</b>
4.1	Fortschreibungsmodi .....	117
4.1.1	Full-Update .....	118
4.1.2	Delta-Initialisierung .....	120
4.1.3	Delta-Update .....	124
4.1.4	InfoPackages in SAP NetWeaver BW 7.30 .....	124
4.1.5	Delta des Datentransferprozesses .....	125
4.1.6	Fehlermanagement im Datentransferprozess .....	128

4.2	Realtime Data Acquisition .....	132
4.2.1	Besonderheiten der Realtime Data Acquisition .....	132
4.2.2	Beispiel: Implementierung eines RDA-Szenarios .....	136
4.3	Felder ROCANCEL und ORECORDMODE .....	140
4.4	Delta-Harmonisierung .....	142
4.4.1	Initiale Beleganlage .....	143
4.4.2	Belegänderung .....	145
4.4.3	Belegstornierung .....	147
4.4.4	Fazit .....	148
4.5	Delta-Mechanismen für Stammdaten .....	149
4.6	Für Fortgeschrittene: Delta-Harmonisierung mit Full-Updates .....	150
4.6.1	Löschen des Datenziels vor dem Update .....	151
4.6.2	Erzeugen eigener Stornobelege .....	152
4.7	Zusammenfassung .....	153
<b>5</b>	<b>Delta-Mechanismen in der Datenmodellierung .....</b>	<b>155</b>
5.1	Grundlagen der Datenmodellierung .....	155
5.2	Integration von Delta-Verfahren in Datenmodelle .....	156
5.3	Delta-Mechanismen in der Layered, Scalable Architecture .....	158
5.3.1	Einführung in die Layered, Scalable Architecture .....	159
5.3.2	Integration von Delta-Verfahren in die LSA .....	162
5.4	Zusammenfassung .....	167
<b>6</b>	<b>Delta-Verfahren im täglichen Einsatz .....</b>	<b>169</b>
6.1	Fehleranalyse .....	169
6.1.1	Vorgehensmodell .....	169
6.1.2	Typische Fehlerquellen .....	173
6.1.3	Fortsetzen abgebrochener Delta-Verbuchungen .....	178
6.2	Simulation und Debugging .....	180
6.2.1	Simulation im Datentransferprozess .....	181
6.2.2	Debugging von Datentransferprozessen .....	184
6.2.3	Debugging-Berechtigungen .....	189
6.2.4	Simulation im 3.x-Datenfluss .....	190
6.2.5	Debugging im 3.x-Datenfluss .....	194
6.3	Zusammenfassung .....	196
<b>7</b>	<b>Zusammenfassung .....</b>	<b>199</b>

**Anhang ..... 203**

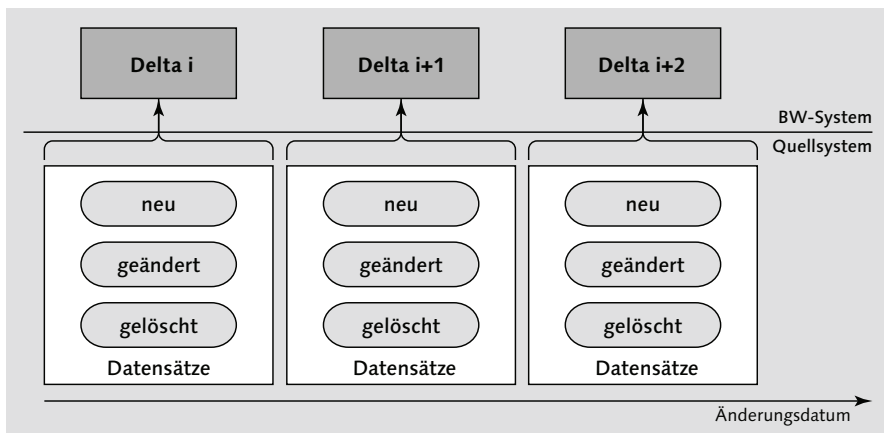
A	Quelltext und Beschreibung zu Programm ZGP_STORNO_CREATE ...	205
B	Übersicht über Datenbanktabellen .....	211
C	Übersicht über die implementierten Delta-Verfahren .....	213
D	Abkürzungsverzeichnis .....	215
E	Literaturverzeichnis .....	217
F	Der Autor .....	219
	Index.....	221

# 1 Einleitung

In jedem Data Warehouse findet ein Delta-Management statt – in einigen Fällen nur in geringem Umfang, in anderen dafür umso mehr. Dieses Buch soll alle Aspekte des Delta-Managements verständlich machen. Die Darstellung ist releaseunabhängig: Wir beschreiben sowohl Release SAP BW 3.x als auch SAP NetWeaver BW 7.x und geben bereits Einblicke in die Funktionsweise von SAP NetWeaver BW 7.30.

Von jeher ist das Delta-Management zentraler Bestandteil eines Data Warehouses. Doch worum geht es dabei eigentlich, und was versteht man unter Delta-Management?

*Delta-Management* ist die Bezeichnung für die Erstellung und den Umgang mit sogenannten Deltas. Ein Delta bezeichnet, im Gegensatz zu einer vollständigen Datenlieferung (Full-Update), Teillieferungen von Daten an das BW-System. Diese Teillieferungen bestehen aus neuen, geänderten und gelöschten Datensätzen, die seit der letzten Datenanforderung entstanden sind (siehe Abbildung 1.1).



**Abbildung 1.1** Neue, geänderte und gelöschte Datensätze seit der letzten Datenanforderung

In Bezug auf das Delta wird differenziert in die Art, in der der Datensatz geliefert wird, und in eine zeitliche Abgrenzung der letzten (erfolgreichen) Datenanforderung.

Der wichtigste Grund, sich über Deltas Gedanken zu machen, ist die schiere Menge an Datensätzen, die in den Quellsystemen be- und entsteht. Der Wunsch nach Echtzeitdaten, also Daten, die direkt nach ihrer Entstehung (also realtime) in BW sichtbar sind, wächst zunehmend. Unabhängig davon sollen in zentralen Unternehmensbereichen wie Finanzwesen oder Einkauf die Daten in der Regel nur einen Tag alt sein. Das heißt, dass binnen einer Nacht Daten aus dem Quellsystem in das BW-System transferiert werden müssen. Dieser Anspruch ist nur dann zu erfüllen, wenn Sie nicht jede Nacht die gesamten Daten des Quellsystems (den vollständigen Datenbestand der entsprechenden Anwendung) extrahieren, sondern sich auf neue, geänderte und gelöschte Datensätze konzentrieren.

Jetzt stellen Sie sich vor, Sie haben mehr als nur ein Quellsystem, was heutzutage den Regelfall darstellt. Dann müssen auch die neuen und geänderten Sätze aller angeschlossenen Quellsysteme in dieser einen Nacht bestimmt und extrahiert werden. Diese Delta-Lieferungen müssen dann je nach Vorgabe im BW-System zusammengeführt, harmonisiert und dem Fachbereich zur Auswertung zur Verfügung gestellt werden.

Sie erkennen also, dass eine *vollständige* Extraktion von Daten aus allen angeschlossenen Quellsystemen nicht zu realisieren ist. Daher beschäftigen wir uns mit dem Delta-Management und widmen dem Thema mit Fug und Recht ein eigenes Buch.

Im Delta-Management unterscheidet man grundsätzlich zwischen der Entstehung und der Verarbeitung der Delta-Lieferungen:

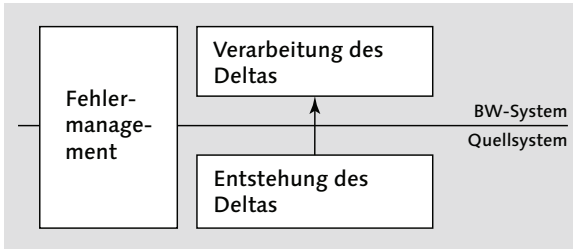
#### ► **Entstehung des Deltas**

Das Delta entsteht grundsätzlich im Quellsystem und wird dem BW-System auf die eine oder andere Weise mitgeteilt. Es gibt eine Vielzahl an Varianten, wie das Delta entsteht und dem BW-System zur Verfügung gestellt wird, wie wir in Kapitel 3, »Delta-Mechanismen im Quellsystem«, sehen werden.

#### ► **Verarbeitung des Deltas**

Je nach Datenlieferung, die wir aus dem Quellsystem erhalten haben, müssen wir im BW-System unterschiedlich darauf reagieren. Nicht jede Delta-Lieferung kann in jedes Datenziel fortgeschrieben werden (siehe Abschnitt 2.6, »Auswirkungen von Delta-Verfahren auf SAP NetWeaver BW«).

Abbildung 1.2 zeigt die beiden Komponenten Entstehung und Verarbeitung des Deltas auf einen Blick.



**Abbildung 1.2** Die drei Säulen Entstehung, Verarbeitung und Fehlerhandling des Delta-Managements

Unabhängig von diesen beiden Aspekten gibt es einen dritten wichtigen Bereich im Delta-Management: das Fehlermanagement. Es wäre naiv zu behaupten, ein Delta-Verfahren verlief grundsätzlich fehlerfrei, denn die Realität sieht anders aus. Es ergeben sich immer wieder Herausforderungen, die durch eine Vielzahl von Gründen entstanden sein können:

- ▶ Datensätze wurden im falschen Format übertragen.
- ▶ Wichtige Felder wurden nicht geliefert (befüllt) und können im BW-System nicht weiterverarbeitet werden (zum Beispiel Belegnummer oder Währung).
- ▶ Die Verbindung bei der Extraktion brach ab.
- ▶ Die letzte Datenverbuchung in Datenziele war noch nicht abgeschlossen, und das System kann mit der geplanten Verarbeitung nicht fortfahren.

Es gibt noch viele weitere Beispiele, warum Delta-Verbuchungen abrechnen können, an dieser Stelle haben wir nur die wichtigsten genannt.

Diese hier beschriebenen Aspekte möchten wir im vorliegenden Buch im Detail behandeln.

## 1.1 Inhalt dieses Buches

Dieses Buch beschreibt alle Aspekte des Delta-Managements – angefangen beim Quellsystem über das BW-System bis hin zum Fehlermanagement. Sie finden im Einzelnen folgende Kapitel:

In **Kapitel 2**, »Grundlagen«, geben wir Ihnen eine Einführung in die Funktionsweise der Delta-Mechanismen. Neben grundlegenden Objekten wie dem Standard-DataStore-Objekt (DSO), dem InfoPackage und dem Datentransferprozess konzentrieren wir uns auf das Delta-Verfahren und seine Bestand-



teile. Anschließend geben wir Ihnen einen Überblick darüber, wie Sie sich über die Delta-Fähigkeit von Extraktoren informieren können. Das Kapitel schließt mit einer Darstellung der Auswirkungen von Delta-Verfahren auf das BW-System.

**Kapitel 3**, »Delta-Mechanismen im Quellsystem«, konzentriert sich auf die Quellsysteme und ihre Eigenschaften in Bezug auf die Delta-Mechanismen. Wir besprechen zu Beginn SAP-Quellsysteme und ihre Eigenschaften wie generische DataSources, Entwicklung eigener Extraktoren unter Nutzung von speziellen Delta-Verfahren und die Realtime Data Acquisition. Anschließend gehen wir auf die Schnittstellen DB Connect, Webservices, Universal Data Integration, Data Services, Data Marts und auch die Dateischnittstelle ein. Jedes Quellsystem wird mit seinen Besonderheiten beim Delta-Handling vorgestellt.

Nach den Quellsystemen wenden wir uns in **Kapitel 4**, »Delta-Verarbeitung im BW-System«, dem BW-System zu. Zunächst behandeln wir die Arten der Datenanforderung wie das Full-Update, die Delta-Initialisierung und das Delta-Update. Anschließend erläutern wir die Realtime Data Acquisition in Bezug auf das BW-System. Darüber hinaus gehen wir auf die deltarelevanten Felder ROCANCEL und ORECORDMODE ein, die eine zentrale Bedeutung im BW-System haben. Zu guter Letzt beschreibt das Kapitel die Delta-Harmonisierung sowie die Entwicklung eigener Stornodatensätze bei der Nutzung von Full-Updates.

Die in den ersten Kapiteln vorgestellten Grundlagen wenden wir in **Kapitel 5**, »Delta-Mechanismen in der Datenmodellierung«, in der Datenmodellierung an. Zuerst erläutern wir kurz den Nutzen von schreiboptimierten DSOs; anschließend beschreiben wir Methoden zur Integration von Delta-Verfahren in Datenmodelle. Zuletzt gehen wir auf die von SAP propagierte Layered, Scalable Architecture (LSA) in Bezug auf die Delta-Integration ein.

**Kapitel 6**, »Delta-Verfahren im täglichen Einsatz«, befasst sich mit dem täglichen Betrieb von BW-Systemen und dem Fehlermanagement. Zur Analyse von Fehlern ist es wichtig, dass man nach einem Schema arbeitet, um alle möglichen Fehlerquellen zu betrachten. Eine Reihe von Fehlern tritt dabei häufiger auf; diesen haben wir einen separaten Abschnitt gewidmet. Nach der Vorstellung von Repeat-Deltas und Reparatur-Full-Requests wenden wir uns den fortgeschrittenen Verfahren Simulation und Debugging zu. Wir beschreiben sowohl InfoPackages als auch Datentransferprozesse und ihre Debugging-Methoden im Detail.

In **Kapitel 7**, »Zusammenfassung«, rekapitulieren wir die wichtigsten Informationen aus diesem Buch.

Im **Anhang** finden Sie neben dem Literaturverzeichnis Quelltexte aus den vorderen Kapiteln des Buches, eine Übersicht über Datenbanktabellen, die am Delta-Mechanismus beteiligt sind, eine Übersicht über Delta-Verfahren, ein Abkürzungs- und ein Stichwortverzeichnis (Index).

## 1.2 Zielgruppen des Buches

Das Buch richtet sich an folgende Zielgruppen:

- ▶ *Entwickler*, die mit der Einführung, Konsolidierung oder Wartung eines BW-Systems konfrontiert sind, erhalten umfassendes Wissen über das Delta-Management von der Entstehung über die Verarbeitung bis zum Fehlermanagement und der Korrektur von Fehlern.
- ▶ *Berater*, die mit der Einführung und/oder Wartung von Data Warehouses bei ihren Kunden betraut sind, erhalten Hinweise zur Einführung und Konzeption eines korrekten Delta-Managements sowie wertvolle Tipps zur Fehlerbehebung.
- ▶ *Administratoren*, die ein BW-System betreuen und warten müssen, erhalten tief gehende Einblicke in die Funktionsweise von Delta-Verfahren und können dadurch beurteilen, wie etwa ein Fehler entstanden ist und wie man ihn in Zukunft vermeiden kann.
- ▶ *Interessierte, Studierende und andere Anwender*, die ausführliche Informationen zu SAP NetWeaver BW, SAP-Quellsystemen und ihrem Umgang mit der Delta-Extraktion suchen, werden in diesem Buch von Grund auf in das Thema eingeführt.

## 3 Delta-Mechanismen im Quellsystem

Dieses Kapitel beschreibt alle Quellsysteme, die im Rahmen des Delta-Handlings genutzt werden können. Im vorangegangenen Kapitel haben Sie die Grundprinzipien der Delta-Extraktion kennengelernt. Sie wissen nun, was Delta-Verfahren sind und woraus sie bestehen und wie Sie sich über die Delta-Fähigkeit eines Extraktors informieren können. In diesem Kapitel erhalten Sie nun Informationen zur Extraktion an sich. Es beantwortet folgende Fragen: Wie arbeitet ein SAP-Quellsystem mit Deltas? Wie kann man eigene Delta-Verfahren implementieren, und ist es möglich, die Delta-Verfahren aus der Tabelle RODELTAM selbst zu verwenden?

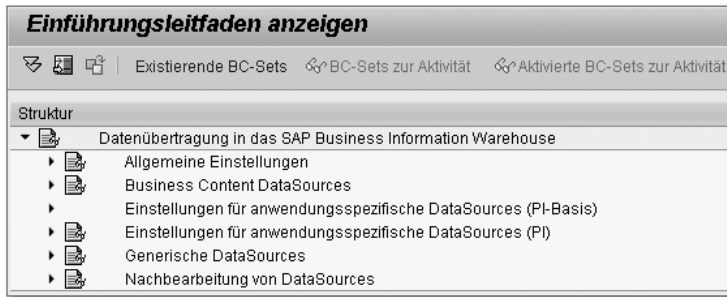
Wir möchten außerdem auf alle möglichen Typen von Quellsystemen eingehen und ihre Delta-Fähigkeit untersuchen:

- ▶ SAP-Quellsysteme
- ▶ DB Connect
- ▶ Flatfiles
- ▶ Webservices
- ▶ Universal Data Integration (UDI)
- ▶ SAP NetWeaver BW als Quellsystem (Data Mart)
- ▶ Data Services (ab SAP NetWeaver BW 7.30)

Im folgenden Abschnitt beginnen wir mit der Extraktion aus SAP-Quellsystemen.

### 3.1 Delta-Queue in der Service-API

Die Service-API (Service Application Programming Interface) ist die Schnittstelle, mit der SAP-Quellsysteme an SAP NetWeaver BW angeschlossen werden. Die Service-API, oder kurz SAPI, besteht aber noch aus viel mehr als nur der reinen Konnektierung an SAP NetWeaver BW-Systeme. So können Sie unter anderem DataSources aus dem Business Content so parametrisieren, dass sie Ihren Wünschen entsprechen (sofern vorgesehen). Die Service-API erreichen Sie über die Transaktion SBIW (Einführungsleitfaden anzeigen), wie Sie in Abbildung 3.1 sehen können.



**Abbildung 3.1** Einstiegsbild der Transaktion SBIW (Service-API)

Die grundlegenden Funktionen der Service-API sollen Sie hier nicht weiter beschäftigen, es sind lediglich die Delta-Funktionen der Service-API interessant. Von besonderer Bedeutung ist dabei der Mechanismus, mithilfe dessen die Delta-Extraktion aus SAP-Quellsystemen vom SAP-System durchgeführt wird: die Delta-Queue.

Die Delta-Queue ist das zentrale Objekt, das zur Verwaltung von neuen und geänderten Datensätzen im SAP-Quellsystem existiert. Die Delta-Queue wird nur in SAP-Quellsystemen genutzt; für andere Quellsysteme werden andere Mechanismen eingesetzt.

Doch was ist die Delta-Queue genau? Prinzipiell ist die Delta-Queue der Ablageort für neue und geänderte Datensätze. Wie Sie in Abschnitt 2.4.1, »Delta-Typ«, erfahren haben, können Daten bei der Extraktion entweder durch die Anwendung selbst (Push-Verfahren) oder erst bei der Extraktion (Pull-Verfahren) in die Delta-Queue geschrieben werden. Die Delta-Queue hat dabei drei wichtige Eigenschaften, die wir im Folgenden erläutern möchten:

- ▶ Ablageort für neue und geänderte Datensätze
- ▶ Speicherung des aktuellen und letzten Deltas
- ▶ Versorgung mehrerer BW-Zielsysteme (durch mehrere Queues)

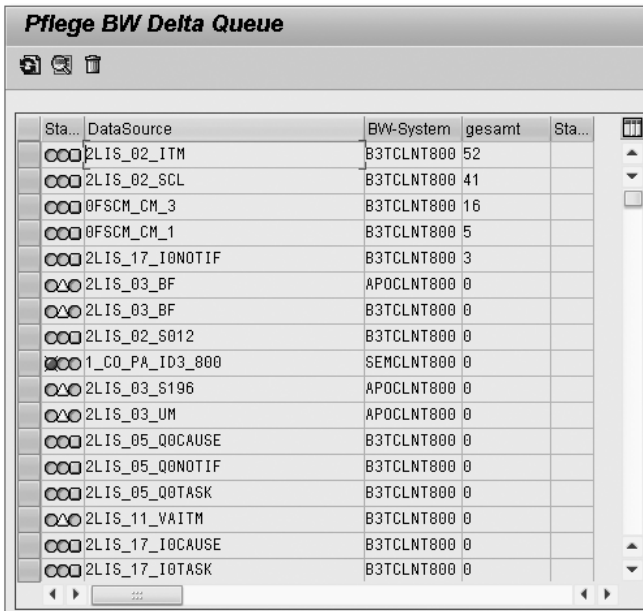
Befassen wir uns zuerst mit dem ersten Punkt, dem Ablageort für neue und geänderte Datensätze.

### 3.1.1 Ablageort für neue und geänderte Datensätze

Wenn eine DataSource in einem SAP-Quellsystem ein Delta-Verfahren implementiert hat, verwendet sie für das Delta-Update die Delta-Queue als Ab-

lageort für neue und geänderte Datensätze. Die Delta-Queue besteht dabei aus einer Reihe von Datenbanktabellen, die am Delta-Prozess beteiligt sind.

Es existiert immer eine Delta-Queue für eine DataSource und ein BW-Zielsystem (siehe Abschnitt 3.1.3, »Versorgung mehrerer BW-Zielsysteme«). Wenn Sie sich die aktuellen Delta-Queues im SAP-Quellsystem und ihren Status anzeigen lassen wollen, verwenden Sie die Transaktion RSA7 (Pflege BW Delta Queue), siehe Abbildung 3.2.



Sta...	DataSource	BW-System	gesamt	Sta...
000	2LIS_02_ITM	B3TCLNT800	52	
000	2LIS_02_SCL	B3TCLNT800	41	
000	0FSCM_CM_3	B3TCLNT800	16	
000	0FSCM_CM_1	B3TCLNT800	5	
000	2LIS_17_I0NOTIF	B3TCLNT800	3	
000	2LIS_03_BF	APOCLNT800	0	
000	2LIS_03_BF	B3TCLNT800	0	
000	2LIS_02_S012	B3TCLNT800	0	
000	1_CO_PA_ID3_800	SEMCLNT800	0	
000	2LIS_03_S196	APOCLNT800	0	
000	2LIS_03_UM	APOCLNT800	0	
000	2LIS_05_Q0CAUSE	B3TCLNT800	0	
000	2LIS_05_Q0NOTIF	B3TCLNT800	0	
000	2LIS_05_Q0TASK	B3TCLNT800	0	
000	2LIS_11_VA1TM	B3TCLNT800	0	
000	2LIS_17_I0CAUSE	B3TCLNT800	0	
000	2LIS_17_I0TASK	B3TCLNT800	0	

**Abbildung 3.2** Übersicht über die Delta-Queues im Quellsystem (Transaktion RSA7)

Den Status der Delta-Queue ermittelt das Programm hinter der Transaktion RSA7 durch die Datenbanktabelle ROOSPRMSC (Steuerparameter pro Channel einer DataSource). Betrachten Sie den Inhalt der Tabelle ROOSPRMSC zum Beispiel für die DataSource 2LIS\_02\_ITM, die in Abbildung 3.2 ganz oben steht, finden Sie den in Abbildung 3.3 gezeigten Eintrag.

An den Feldern können Sie folgende Informationen ablesen:

► **Feld INITRNR**

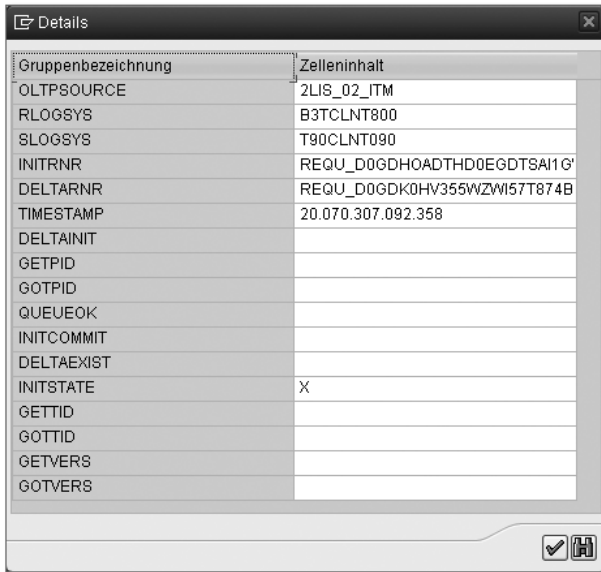
An Feld INITRNR erkennen Sie die Request-Nummer der Delta-Anforderung, die durch das Starten eines InfoPackages im BW-System vergeben wurde.

► **Feld DELTAINIT**

Das Feld DELTAINIT gibt an, ob derzeit eine Delta-Initialisierung (siehe Abschnitt 4.1.2) stattfindet. Für die DataSource 2LIS\_02\_ITM ist das Feld leer, das heißt, im Moment läuft kein Delta-Init.

► **Feld INITSTATE**

Das Feld INITSTATE gibt an, dass eine Delta-Initialisierung stattgefunden hat. Es wird gesetzt, nachdem die Delta-Initialisierung erfolgreich durchgelaufen ist.



Gruppenbezeichnung	Zelleninhalt
OLTPSOURCE	2LIS_02_ITM
RLOGSYS	B3TCLNT800
SLOGSYS	T90CLNT090
INTRNR	REQU_D0GDH0ADTHD0EGDTSAl1G'
DELTARNR	REQU_D0GDK0HV355WZWI57T874B
TIMESTAMP	20.070.307.092.358
DELTAINIT	
GETPID	
GOTPID	
QUEUEOK	
INITCOMMIT	
DELTAEXIST	
INITSTATE	X
GETTID	
GOTTID	
GETVERS	
GOTVERS	

**Abbildung 3.3** Eintrag zur DataSource 2LIS\_02\_ITM in der Tabelle ROOSPRMSC

Zu jeder DataSource steht in der Transaktion RSA7 (Pflege BW Delta Queue) in der Spalte GESAMT eine Menge, der man auf den ersten Blick die Anzahl Datensätze in der Delta-Queue zur DataSource zuordnen würde. Im Beispiel der DataSource 2LIS\_02\_ITM könnte man also denken, hier seien 52 Datensätze enthalten. Dies ist aber nur in einigen wenigen Fällen korrekt!

Grundsätzlich steht in der Spalte GESAMT die Anzahl LUWs (Logical Unit of Work). Eine LUW können Sie mit einem Request vergleichen, sie beinhaltet also im Regelfall mehr als einen Datensatz. Hier spielt die Eigenschaft des Delta-Verfahrens eine entscheidende Rolle. So erscheint nur dann eine Zahl größer 0 in der Spalte GESAMT, wenn die DataSource ein Push-Verfahren verwendet. Warum? Nur im Fall der Push-Verfahren schreibt die Anwendung die Daten direkt in die Delta-Queue, in allen anderen Fällen wird erst bei der

Extraktion bestimmt, welche Sätze neu sind oder sich geändert haben. Daher deutet eine Null nicht zwangsläufig darauf hin, dass in der Delta-Queue keine Sätze vorhanden sind. Achten Sie also in diesem Fall immer auf das implementierte Delta-Verfahren, im Speziellen den verwendeten Delta-Typ. Die Größe der LUW ist dabei unterschiedlich, je nach vorhandenem Delta-Verfahren kann eine LUW aus einem oder mehreren Tausend Datensätzen bestehen.

#### Besonderheit: Die Anzahl der Datensätze ist gleich der Anzahl der LUWs

Wie Sie in Abschnitt 3.2, »Logistik-Extraktion«, noch erfahren werden, ist in manchen Fällen die Anzahl der LUWs gleich der Anzahl der Datensätze in der Delta-Queue. In einigen Fällen wird nämlich pro Datensatz genau eine LUW erstellt, jede LUW besteht also aus einem Satz. Dies ist zum Beispiel bei der Delta-Direkt-Verbuchung in der Logistik der Fall.

Was passiert nun, wenn Sie die Delta-Extraktion starten oder sich die Datensätze der Delta-Queue in der Transaktion RSA7 (Pflege BW Delta Queue) anzeigen lassen wollen? Der Status der Delta-Queue ist in der Tabelle ROOSPRMSC abgelegt. Da die Delta-Queue auf der Queue-Funktionalität innerhalb der RFC-Technologie im SAP NetWeaver Application Server (kurz qRFC) basiert, sind die Daten zur Delta-Queue in der Tabelle TRFCQOUT (Beschreibung der tRFC-Queues (Ausgangsqueue)) abgelegt.

#### RFC-Verbindungen im SAP-System

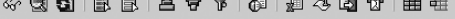
Die häufigste Verbindung zwischen zwei SAP-Systemen ist die RFC-Verbindung (Remote Function Call). Das SAP-System unterscheidet dabei eine Reihe verschiedener RFC-Typen, die wir kurz vorstellen möchten:

- ▶ sRFC: Bezeichnet den synchronen RFC zweier Systeme.
- ▶ tRFC: Der transaktionale RFC arbeitet asynchron, das heißt, der aufgerufene Baustein muss beim Aufruf nicht zwangsläufig verfügbar sein. Der Aufruf wird unter einer Transaktionskennung (TID) entsprechend abgespeichert und landet in einer Warteschlange.
- ▶ qRFC: Der queued RFC dient dazu, eine Menge von tRFC-Aufrufen in einer Logical Unit of Work (LUW) zu bündeln, um hier die Serialisierung sicherstellen zu können.

Für weitere Informationen verweisen wir Sie auf die SAP-Bibliothek  
[http://help.sap.com/saphelp\\_nw04/helpdata/de/6f/1bd5b6a85b11d6b28500508b5d5211/content.htm](http://help.sap.com/saphelp_nw04/helpdata/de/6f/1bd5b6a85b11d6b28500508b5d5211/content.htm).

Die Anzahl der LUWs, die Sie in der Spalte GESAMT der Transaktion RSA7 ablesen können, entspricht damit der Anzahl Datensätze in der Tabelle

TRFCQOUT für diese DataSource. In unserem Beispiel der DataSource 2LIS\_02\_ITM hat die Tabelle TRFCQOUT also für die DataSource 2LIS\_02\_ITM 52 Datensätze. Sie können dies leicht über das Feld QNAME durch die Tabelle TRFCQOUT nachprüfen. In Abbildung 3.4 wurden die Datensätze entsprechend selektiert.

Data Browser: Tabelle TRFCQOUT 52 Treffer											
<div></div>											
	MANDT	ARFCIPID	ARFCPID	ARFCTIME	ARFCTIDCNT	QNAME	DEST	QCOUNT	HPQNAME	NOSEND	QSTATE
	800	0A165C49	03FC	4B28AFDC	0007	BW8002LIS_02_ITM	B3TCLNT800	12609576600000010001		X	READY
	800	0A165C49	055C	4B62F7A0	0004	BW8002LIS_02_ITM	B3TCLNT800	12647771200000020000		X	READY
	800	0A165C49	0A08	4BF50D2F	0E33	BW8002LIS_02_ITM	B3TCLNT800	12743508950000010000		X	READY
	800	0A165C49	0A88	4BF13381	0004	BW8002LIS_02_ITM	B3TCLNT800	12740985650000010001		X	READY
	800	0A165C49	0A88	4BF139D8	000B	BW8002LIS_02_ITM	B3TCLNT800	12741001840000010001		X	READY
	800	0A165C49	0A88	4BF13BD0	0010	BW8002LIS_02_ITM	B3TCLNT800	12741007020000010001		X	READY
	800	0A165C49	0A88	4BF13CC2	0017	BW8002LIS_02_ITM	B3TCLNT800	127410093000000120001		X	READY
	800	0A165C49	0A88	4BF13D5E	001C	BW8002LIS_02_ITM	B3TCLNT800	1274101087000000010001		X	READY
	800	0A165C49	0A88	4BF236A7	0023	BW8002LIS_02_ITM	B3TCLNT800	12741649030000070001		X	READY
	800	0A165C49	0A88	4BF23E66	0028	BW8002LIS_02_ITM	B3TCLNT800	12741668870000010001		X	READY
	800	0A165C49	0A88	4BF23F94	002F	BW8002LIS_02_ITM	B3TCLNT800	127416718800000120001		X	READY
	800	0A165C49	0A88	4BF2406D	0034	BW8002LIS_02_ITM	B3TCLNT800	12741674060000010001		X	READY
	800	0A165C49	0A88	4BF3BC05	0039	BW8002LIS_02_ITM	B3TCLNT800	12742645820000010001		X	READY
	800	0A165C49	0BCC	4BCDAAB2	0007	BW8002LIS_02_ITM	B3TCLNT800	1271789779000000010001		X	READY
	800	0A165C49	0BCC	4BD03335	000C	BW8002LIS_02_ITM	B3TCLNT800	1271935800000000010001		X	READY
	800	0A165C49	0BCC	4BD03386	0012	BW8002LIS_02_ITM	B3TCLNT800	1271935879000000010001		X	READY

**Abbildung 3.4** LUWs zur DataSource 2LIS\_02\_ITM aus der Tabelle TRFCQOUT

Zur Extraktion wird also jede LUW aus der Tabelle TRFCQOUT zur Data-Source bestimmt. Anschließend werden die Datensätze jeder LUW über die ARFC\*-Parameter (Tabelle TRFCQOUT) aus Tabelle ARFCSSTATE bzw. ARFCSDATA bestimmt. Im Beispiel verwenden Sie den ersten Datensatz aus der Tabelle TRFCQOUT mit den folgenden Parametern zur Bestimmung des Zustands der LUW und der darin enthaltenen Datensätze (siehe Tabelle 3.1).

Feldname	Inhalt
ARFCIPID	0A165C49
ARFCPID	03FC
ARFCTIME	4B28AFDC
ARFCTIDCNT	0007

**Tabelle 3.1** Bestimmung der Datensätze mit ARFC\*-Parametern aus der Tabelle TRFCQOUT

Die Datenbanktabelle ARFCSSTATE gibt dabei den Status der LUW an, Tabelle ARFCSDATA beinhaltet die Daten zu dieser LUW. Die Daten in der

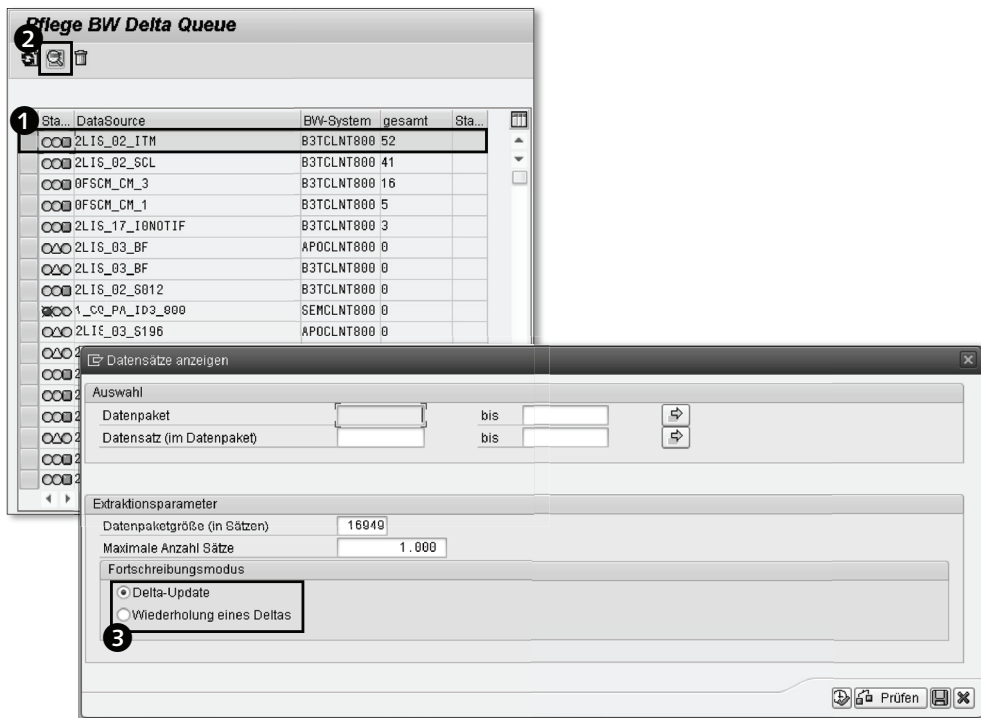




### 3.1.2 Speicherung des aktuellen und letzten Deltas

Die Delta-Queue erlaubt es, abgebrochene Delta-Anforderungen aus dem BW-System zu wiederholen. Dies ist möglich, da eine Delta-Queue nicht nur die aktuellen, noch nicht ins BW-System übertragenen Datensätze beinhaltet, sondern auch das letzte erfolgreiche Delta. Das letzte Delta wird automatisch angefordert, wenn eine Delta-Anforderung im BW-System abbricht und ein neues Delta angefordert wird. Man spricht dann vom sogenannten *Repeat-Delta* (siehe Abschnitt 6.1.3, »Fortsetzen abgebrochener Delta-Verbuchungen«).

In der Transaktion RSA7 können Sie sich sowohl über die aktuellen Datensätze als auch das letzte erfolgreiche Delta informieren. Markieren Sie dazu die DataSource (❶), und klicken Sie auf **DETAIL** (❷). Anschließend öffnet sich ein Pop-up, in dem Sie sich das aktuelle oder das letzte Delta anzeigen lassen können (❸), wie Abbildung 3.6 zeigt.



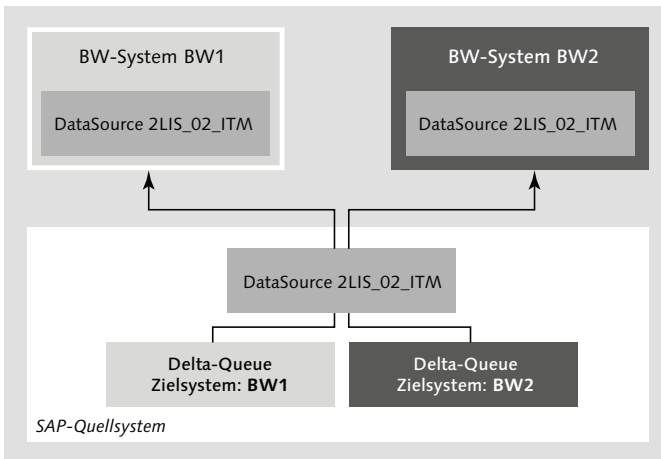
**Abbildung 3.6** Anzeige des aktuellen oder letzten Deltas einer Delta-Queue

In Abschnitt 6.1.3 beschreiben wir das Repeat-Delta noch genauer.

### 3.1.3 Versorgung mehrerer BW-Zielsysteme

Wir haben in Abschnitt 2.4.1, »Delta-Typ«, schon darauf hingewiesen, dass eine Delta-Queue immer DataSource- und BW-Zielsystem-spezifisch ist. Die DataSource-Spezifik ist einleuchtend, doch was hat es mit dem BW-Zielsystem auf sich?

Ganz einfach: Ein SAP-Quellsystem kann als Datenquelle unterschiedlicher BW-Systeme dienen. So könnten Sie zwei BW-Systeme besitzen, das eine haben Sie zum Beispiel für tagesgenaues operationales Reporting aufgebaut, das zweite zur Abbildung von Monatsscheiben. Beide BW-Systeme verwenden aber die gleiche DataSource, um Daten aus dem SAP-System zu bestimmen (siehe Abbildung 3.7).



**Abbildung 3.7** Versorgung mehrerer BW-Zielsysteme

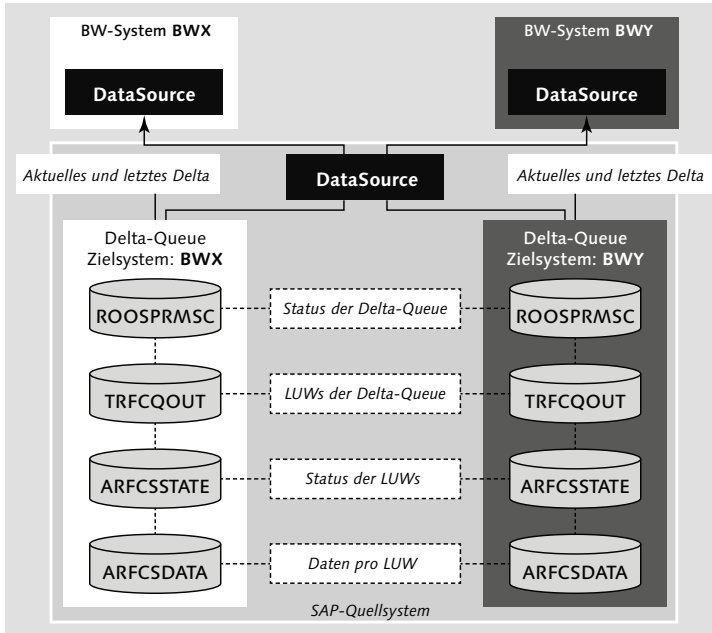
Die Service-API kann mit dieser Anforderung umgehen, indem sie zwei Delta-Queues anlegt, eine für BW-System 1, die andere für BW-System 2. Dadurch kommen sich beide Systeme bei der Extraktion nicht in die Quere.

#### Achtung: Löschen von Delta-Queues

Grundsätzlich haben Sie die Möglichkeit, Delta-Queues zu löschen, entweder über die Transaktion RSA7 im Quellsystem oder über das Löschen der Delta-Initialisierung aus dem BW-System (siehe Abschnitt 6.1.3, »Fortsetzen abgebrochener Delta-Verbuchungen«). Nach dem Löschen von Delta-Queues ist immer eine neue Delta-Initialisierung erforderlich, in manchen Fällen ist diese nur schwer oder mit großem Aufwand wiederherstellbar (siehe Abschnitt 3.2, »Logistik-Extraktion«), besonders dann, wenn die Systemhoheit nicht bei Ihnen liegt.

Löschen Sie Delta-Queues nur nach grundsätzlicher Abwägung der Vor- und Nachteile, und führen Sie sich immer die Auswirkungen vor Augen. In Abschnitt 6.1.3 gehen wir noch genauer auf die Risiken des Löschens von Delta-Queues ein.

Abbildung 3.8 stellt alle Eigenschaften und Komponenten der Delta-Queue zusammenfassend dar.



**Abbildung 3.8** Eigenschaften und Tabellen der Delta-Queue (Zusammenfassung)

Sie haben nun die Delta-Queue als zentralen Mechanismus der Delta-Extraktion kennengelernt. Wir können uns nun den weiteren Möglichkeiten und Besonderheiten in der Extraktion aus SAP-Quellsystemen widmen.

### 3.2 Logistik-Extraktion

Grundsätzlich arbeiten alle Delta-Verfahren mit der Delta-Queue, auch die Logistik macht hier keine Ausnahme. Allerdings gibt es bei der Logistik einige Besonderheiten, die die Extraktion immer wieder vor Herausforderungen stellt – vor allem dann, wenn man etwas an der bestehenden Fortschreibung ändern möchte. In diesem Abschnitt stellen wir daher die Logistik-Datenextraktion vor und gehen auf ihre Besonderheiten ein.

3.2.1 Logistik-Applikation

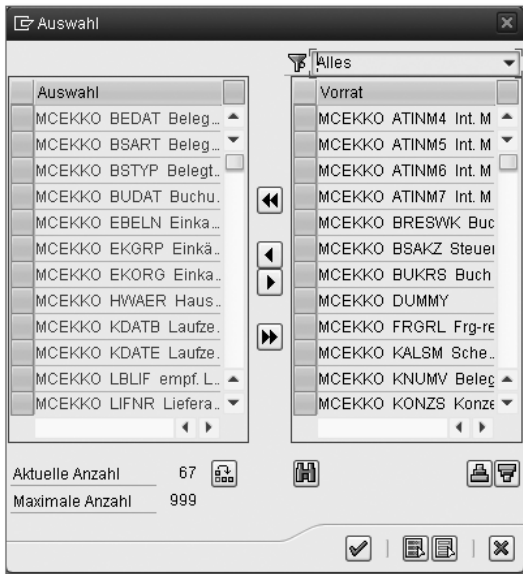
Vielleicht können Sie sich noch an das Logistik-Informationssystem (LIS) erinnern, eine Reporting-Plattform, die zur Auswertung von Logistik-Applikationen verwendet wird. Das LIS nutzt für den Umgang mit Logistik-Daten die sogenannten LIS-Kommunikationsstrukturen, die Sie sich vereinfacht als interne Tabellen vorstellen können. Auch die Logistik-Extraktion arbeitet mit den LIS-Kommunikationsstrukturen, LIS selbst findet aber keine Anwendung. Sie müssen LIS für eine Extraktion weder einrichten noch nutzen.

Generell unterscheidet man in der Logistik-Datenextraktion verschiedene Applikationen, zum Beispiel den Vertrieb oder den Einkauf. Jede Applikation hat eine zweistellige Nummer erhalten, die Sie in der Transaktion LBWE (LO-Datenextraktion: Customizing Cockpit) einsehen können. Jeder Applikation sind Extraktstrukturen zugeordnet, auf deren Basis Sie eine DataSource generieren können. Abbildung 3.9 zeigt Ihnen das Customizing Cockpit.

LO-Datenextraktion: Customizing Cockpit				
Quelldaten	Struktur	DataSource	Verbuchung...	Verbuchungs-Mod...
Logistik-Applikationen				
02 : Einkauf			Job-Steuer...	Delta direkt
Extraktstrukturen				
MC02M_DACC: Extraktion Einkauf (Konto)	Pflege	2LIS_02_ACC		inaktiv
MC02M_DCGR: erbrachte Leistung: Belieferungen der	Pflege	2LIS_02_CGR		aktiv
MC02M_OHDR: Extraktion Einkauf (Kopf)	Pflege	2LIS_02_HDR		aktiv
MC02M_OITM: Extraktion Einkauf (Position)	Pflege	2LIS_02_ITM		aktiv
MC02M_OSCN: Extraktion Einkauf (Einteilung)	Pflege	2LIS_02_SCL		aktiv
MC02M_OSCN: erbrachte Leistung: Bestätigung der E	Pflege	2LIS_02_SCN		aktiv
MC02M_OSGR: erbrachte Leistung: Belieferungen der	Pflege	2LIS_02_SGR		aktiv
03 : Bestandscontrolling			Job-Steuer...	Delta direkt
04 : Fertigung			Job-Steuer...	Delta direkt
05 : Qualitätsmanagement			Job-Steuer...	Delta direkt
06 : Rechnungsprüfung			Job-Steuer...	Delta direkt
08 : Transport			Job-Steuer...	Delta direkt
11 : SD-Verkauf BW			Job-Steuer...	Delta direkt
12 : LE-Versand BW			Job-Steuer...	Delta direkt
13 : SD-Faktura BW			Job-Steuer...	Delta direkt
17 : Instandhaltung BW			Job-Steuer...	Delta direkt
18 : Kundenservice BW			Job-Steuer...	Delta direkt
40 : Warenwirtschaft			Job-Steuer...	Delta direkt
43 : BW: Retail POS Daten: Kassierer			Job-Steuer...	Delta direkt
44 : BW: Retail POS Daten: Kassenbons			Job-Steuer...	Delta direkt
45 : Agenturgeschäft			Job-Steuer...	Delta direkt
46 : Global Trade			Job-Steuer...	Delta direkt
20 : Workflow				
09 : Externe Daten (ohne SAP Bezug)				

Abbildung 3.9 Transaktion LBWE (LO-Datenextraktion: Customizing Cockpit)

Die Extraktstrukturen der DataSources nehmen immer den Umweg über die LIS-Kommunikationsstrukturen. Wenn Sie einer DataSource aus dem Logistik-Umfeld ein Feld hinzufügen möchten, können Sie das Feld über das LO-Customizing-Cockpit (Transaktion LBWE) direkt aus den Kommunikationsstrukturen übernehmen. Abbildung 3.10 zeigt Ihnen die Erweiterung der Extraktstruktur für die Extraktion von Einkaufspositionsdaten (MC02M\_OITM).

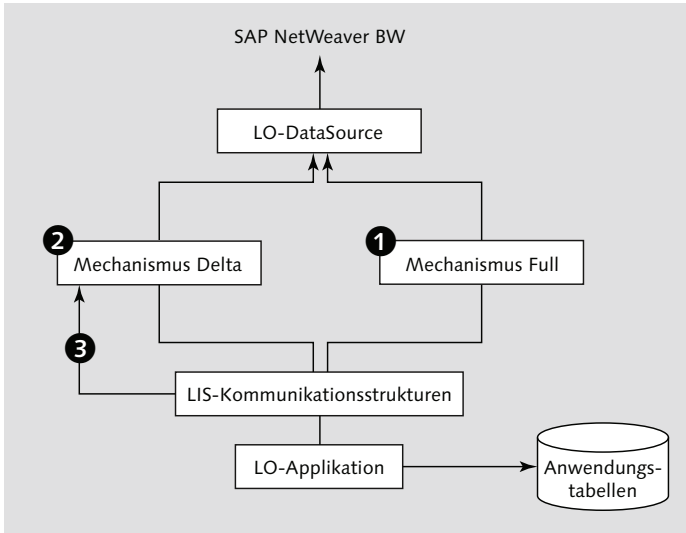


**Abbildung 3.10** Erweitern von Extraktstrukturen in der Logistik

Auf der rechten Seite erkennen Sie den Vorrat der Kommunikationsstrukturen, auf der linken Seite sind die derzeit übernommenen Felder dargestellt.

In der Logistik-Extraktion unterscheiden wir die Delta-Initialisierung bzw. das Full-Update (❶) von der Delta-Anforderung (❷). Für beide Anforderungsmodi kommen unterschiedliche Techniken zum Einsatz. Abbildung 3.11 zeigt die Logistik-Extraktion in einer Übersicht.

Hinter Schritt (❸) verbergen sich die sogenannten *Verbuchungsmethoden*, die angeben, wie der Mechanismus Delta befüllt werden kann. Den Delta-Mechanismus haben Sie dabei schon kennengelernt, wir müssen ihn nicht weiter vertiefen, es handelt sich um die Delta-Queue. Auch die Logistik arbeitet mit der Delta-Queue und den aus Abschnitt 2.4 bekannten Delta-Verfahren, die ebenfalls in der Tabelle RODELTAM aufgelistet sind.



**Abbildung 3.11** Prinzip der Logistik-Extraktion

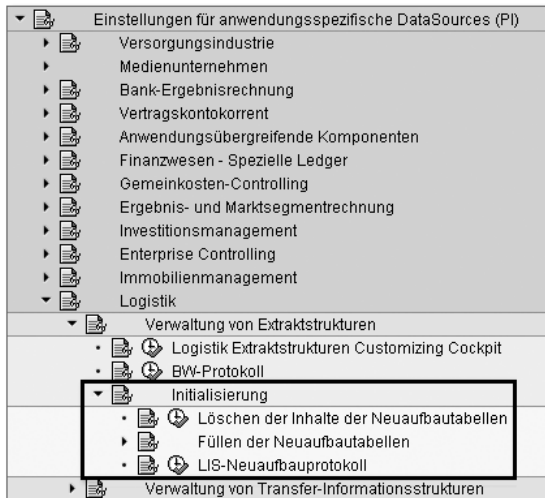
In den folgenden Abschnitten beleuchten wir die übrigen beiden Schritte im Detail und gehen auf ihre Besonderheiten in Bezug auf die Extraktion ein. Den Full-Mechanismus werden wir allerdings nur anreißen, da wir uns in diesem Buch auf die Delta-Mechanismen konzentrieren.

### 3.2.2 Delta-Initialisierung/Full-Update

Bevor Daten im Delta-Verfahren extrahiert werden können, ist eine sogenannte *Delta-Initialisierung* erforderlich. Sie dient dazu, die Delta-Queue bereitzustellen, in der anschließend die neuen und geänderten Datensätze abgelegt werden können. Dies ist auch in der Logistik-Applikation der Fall.

Allerdings wird eine Delta-Initialisierung immer eine leere Menge selektieren, wenn Sie den Mechanismus nicht vorher »in Gang gesetzt« haben. Im Gegensatz zu anderen Applikationen ist es in der Logistik notwendig, die sogenannte *Neuaufbautabelle* zu füllen. Die Neuaufbautabelle ist ein Zwischenspeicher, der die Daten aus den Logistik-Applikationen aufnimmt und an das BW-System überträgt. Sie ist ausschließlich für die Delta-Initialisierung erforderlich. Das bedeutet: Bevor Sie eine Delta-Initialisierung (und damit ein Delta-Update) durchführen können, müssen Sie die Neuaufbautabellen Ihrer Logistik-Applikation befüllen. Dies geschieht in der Service-API in der Transaktion SBIW (Einführungsleitfaden anzeigen) im Menübaum DATENÜBERTRAGUNG IN DAS SAP BUSINESS INFORMATION WAREHOUSE • EIN-

STELLUNGEN FÜR ANWENDUNGSSPEZIFISCHE DATA SOURCES (PI) • LOGISTIK • VERWALTUNG VON EXTRAKTSTRUKTUREN • INITIALISIERUNG (siehe Abbildung 3.12).



**Abbildung 3.12** Löschen und Füllen der Neuaufbautabellen

Nach einer Delta-Initialisierung oder einem Full-Update wird die Neuaufbautabelle nicht mehr benötigt. Sie können sie anschließend wieder löschen.

Es gilt, in diesem Zusammenhang zwei Aspekte zu beachten:

- ▶ Um Datenkonsistenz zu gewährleisten, ist eine Sperre aller an der Logistik-Applikation beteiligten Transaktionen notwendig. Andernfalls können nicht alle Datensätze berücksichtigt werden, und es entstehen Lücken, die auch durch ein Delta-Update nicht gefüllt werden können. Stellen Sie beim Neuaufbau also sicher, dass Sie die Transaktionssperre durchgeführt haben oder zumindest in einer verbuchungsfreien Zeit (sofern vorhanden) agieren.
- ▶ Im Lebenszyklus eines BW-Systems kommt es immer wieder vor, dass Sie eine neue Delta-Initialisierung durchführen müssen. Die Gründe hierfür sind zahlreich, Sie möchten zum Beispiel Inkonsistenzen korrigieren oder die Extraktstruktur erweitern. In all diesen Fällen ist ein Neuaufbau notwendig, das heißt, die Neuaufbautabellen müssen gefüllt werden. Das führt zu einem erhöhten Abstimmbedarf mit der zuständigen Basisabteilung des Quellsystems, den Sie unbedingt einkalkulieren müssen.

Für weitere Informationen zur Logistik-Extraktion über die Neuaufbautabelle verweisen wir Sie auf Kessler, *Datenbeschaffung für SAP NetWeaver BW*, 2011.



### 3.2.3 Verbuchungsmethoden

Auch die Logistik-Applikation bedient sich der Delta-Queue als zentralem Mechanismus, das heißt, hier werden alle neuen und geänderten Datensätze abgelegt. Die Delta-Queue steht bereit, wenn die Delta-Initialisierung erfolgreich durchgelaufen ist, oder schon einen Schritt vorher, wenn Sie mit der Early-Delta-Initialisierung arbeiten (siehe Abschnitt 4.1.2).

Wir zeigen nun, wie die Daten aus der Logistik in die Delta-Queue gelangen. Wir schauen uns also an, wie der Delta-Typ D (Push-Verfahren) implementiert und genutzt wird. Das Stichwort hierzu lautet *Verbuchungsmethoden*.

#### Exkurs: V1-, V2- und V3-Verbuchungsprozesse

Jedes SAP-System unterscheidet grundsätzlich die zeitkritische Verbuchung von der zeitunkritischen. Erstere wird *V1-Verbuchung* genannt, Letztere *V2-Verbuchung*. Eine *V3-Verbuchung* ist ein Sammellauf, der ausschließlich V2-Verbuchungen zusammenfasst. Diese drei Verbuchungsarten kommen sehr häufig in SAP-Transaktionen vor, so auch in der Logistik.

*Beispiel:* Es findet eine Änderung am Materialstamm in der Logistik statt. Während der Speicherung des Materials ist es wichtig, dass niemand anders auf das Material zugreift, die Änderung am Materialstamm ist also zeitkritisch und wird im Rahmen einer V1-Verbuchung durchgeführt. Als zeitunkritisch hingegen wird zum Beispiel eine Statistik betrachtet, die im Rahmen der Materialstammänderung aktualisiert wird. Diese muss nicht sofort erfolgen.

Im Rahmen der Logistik-Extraktion ist es wichtig, zu wissen, dass eine V2-Verbuchung nicht von einer erfolgreichen oder erfolglosen V1-Verbuchung abhängt. Sobald eine Speicherung von Daten stattgefunden hat, die eine V1- und eine V2-Verbuchung beinhaltet, werden beide unabhängig voneinander bearbeitet. Zuletzt kann die V2-Verbuchung auch in einem Sammellauf eingeplant werden, was man dann als V3-Verbuchung oder V3-Sammellauf bezeichnet.

Für genauere Informationen zu den Verbuchungsprozessen verweisen wir Sie auf die ausführliche Dokumentation in der SAP-Bibliothek

[http://help.sap.com/saphelp\\_nw70ehp1/helpdata/de/e5/de86e135cd11d3acb00000e83539c3/content.htm](http://help.sap.com/saphelp_nw70ehp1/helpdata/de/e5/de86e135cd11d3acb00000e83539c3/content.htm).

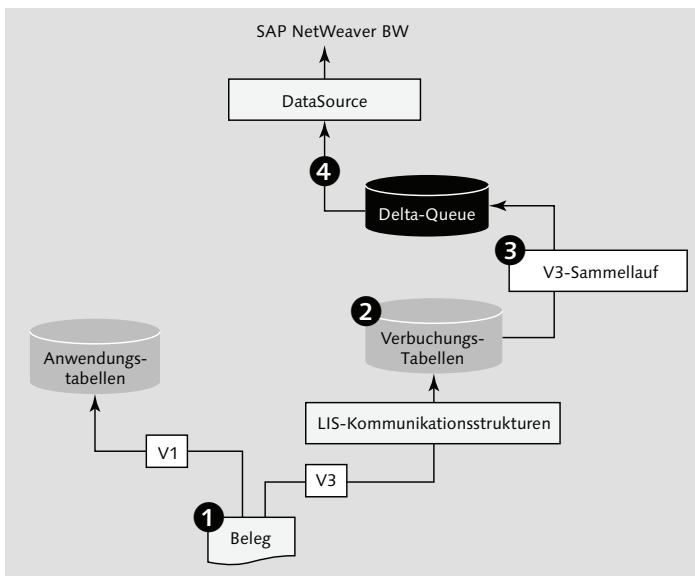
Die Logistik-Extraktion unterscheidet vier verschiedene Verbuchungsmethoden, also Verfahren, wie die Daten aus der Anwendung in die Delta-Queue gelangen.

- ▶ serialisierte V3-Verbuchung
- ▶ unserialisierte V3-Verbuchung
- ▶ Delta Direkt
- ▶ Delta Queued

Die Verbuchungsmethode *serialisierte V3-Verbuchung* ist seit Service-API-Release PI\_BASIS 2003.1 aus dem System entfernt worden. Wir sprechen sie dennoch an, da sie zum einen einen guten Überblick und ersten Eindruck in die Materie der Verbuchungsmethoden erlaubt und wir zum anderen auch frühere Systeme nicht unberücksichtigt lassen wollen.

### Serialisierte V3-Verbuchung

Die serialisierte V3-Verbuchung arbeitet, wie man am Namen erkennen kann, in serieller Form und stellt einen Sammellauf dar. Zum besseren Verständnis erläutern wir den Ablauf anhand eines Schaubilds (siehe Abbildung 3.13).



**Abbildung 3.13** Schaubild der serialisierten V3-Verbuchung

In einem ersten Schritt **(1)** entsteht ein Beleg in einer Applikation, zum Beispiel eine Bestellung im Einkauf. Diese Information muss so schnell wie möglich in den Anwendungstabellen hinterlegt werden, das heißt, eine V1-Verbuchung übernimmt die Verbuchung in den Anwendungstabellen, zum Beispiel EKKO/EKPO.

Die serialisierte V3-Verbuchung arbeitet nun mit sogenannten Verbuchungstabellen, in denen die Belege der Applikation zwischengespeichert werden. Diese Belege landen **(2)** über eine V3-Verbuchung über die LIS-Kommunikationsstrukturen in den Verbuchungstabellen. Sie werden dann über einen

V3-Sammellauf, der von Ihnen einzuplanen ist, in die Delta-Queue geschrieben (3). Aus dieser findet die Extraktion für das BW-System statt.

Die serialisierte V3-Verbuchung hatte eine Reihe von Schwächen, weshalb sie mit Service-API-Release PI\_BASIS 2003.1 aus dem System verbannt wurde. Vor allem zwei Probleme sind hier zu nennen:

► **Abhängigkeit von der V2-Verbuchung**

In Schritt (2) aus Abbildung 3.13 sehen Sie, dass der Beleg über eine V3-Verbuchung in den Verbuchungstabellen landet. Diese V3-Verbuchung ist abhängig von einer erfolgreichen V2-Verbuchung. Das hat zur Folge, dass ein Beleg über eine erfolgreiche V1-Verbuchung zwar in den Anwendungstabellen landet, aber nicht in der Verbuchungstabelle, wenn die V2-Verbuchung erfolglos war. Zur Erinnerung: V2-Verbuchungen sind zeitunkritisch und haben mit der reinen Beleginformation nichts zu tun, sie können für die Verbuchung also unberücksichtigt gelassen werden. Wichtig sollte eine Abhängigkeit zur V1-Verbuchung sein.

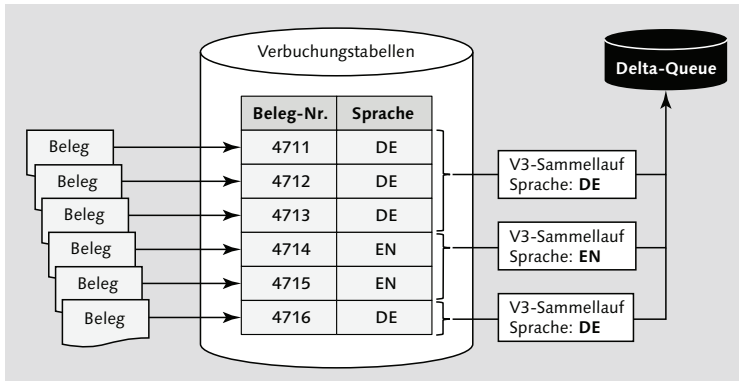
► **Serialisierte Sprachenverarbeitung**

Neben der V2-Abhängigkeit gab es bei der serialisierten V3-Verbuchung eine weitere Einschränkung. Der V3-Sammellauf, der die Beleginformationen aus den Verbuchungstabellen in die Delta-Queue schrieb, war nur für *eine* Sprache ausgelegt. Hatten Sie also Nutzer im System, die mit unterschiedlichen Anmeldesprachen arbeiteten (zum Beispiel Deutsch, Englisch und Japanisch), kam der V3-Sammellauf damit nicht zurecht. Was passierte? Nehmen Sie an, die ersten drei Belege in den Verbuchungstabellen sind in deutscher Sprache, anschließend folgen zwei englischsprachige Belege, anschließend wieder ein deutscher Beleg (siehe Abbildung 3.14). Der Sammellauf extrahiert, da er serialisiert arbeitet, zuerst die drei deutschen Belege. Im nächsten Schritt erkennt er, dass der nächste Beleg kein deutscher mehr ist, plant sich aus und dafür den englischen Sammellauf ein. Dieser kann die folgenden zwei Belege abarbeiten, muss sich aber bei Beleg 6 wieder ausplanen, da er mit der deutschen Sprache nicht zurechtkommt. Dadurch konnte die serialisierte V3-Verbuchung nicht mit großen Datenvolumina umgehen, die in der Logistik aber sehr häufig vorkommen.

Neben diesen beiden Hauptproblemen gab es ein paar Nebenprobleme wie zum Beispiel die notwendige Voraussetzung der exakten Synchronisation aller Uhrzeiten der beteiligten Applikationsserver.

Durch die serialisierte V3-Verbuchung entstanden eine Menge LUWs in der Delta-Queue. Nach jedem Sprachwechsel in den Verbuchungstabellen wurde

eine neue LUW erzeugt, die im schlechtesten Fall aus nur einem Datensatz bestand.



**Abbildung 3.14** Sprachenabhängigkeit des V3-Sammellaufs

### Unserialisierte V3-Verbuchung

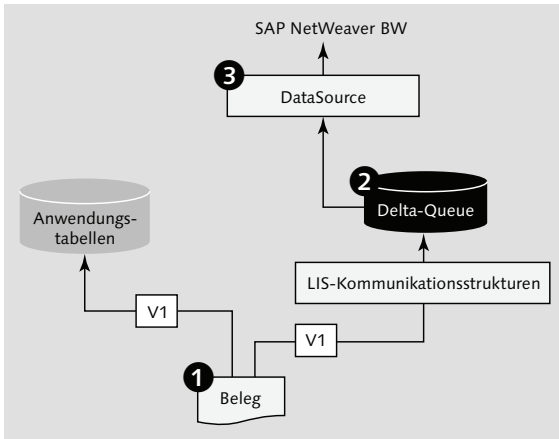
Die unserialisierte V3-Verbuchung arbeitete im Endeffekt analog der serialisierten V3-Verbuchung, die Abbildung zur serialisierten V3-Verbuchung hat auch hier Gültigkeit, weshalb sie nicht wiederholt wird. Das Einzige, was sich bei Einführung der unserialisierten V3-Verbuchung änderte, war die Umstellung von serialisiert zu unserialisiert. Die Schwäche der Sprachenabhängigkeit des V3-Sammellaufs verschwand also, da der V3-Sammellauf alle Daten einer Sprache unabhängig von ihrer Entstehung selektierte und in die Delta-Queue schrieb. Dadurch konnte die unserialisierte V3-Verbuchung mit höheren Datenvolumina arbeiten, jedoch blieb die Abhängigkeit von der V2-Verbuchung, weshalb es zu Inkonsistenzen kommen konnte. Die Anzahl der LUWs in der Delta-Queue verringerte sich mit Einführung der unserialisierten V3-Verbuchung, es gab nur noch so viele LUWs, wie es unterschiedliche Sprachen der Belege gab. Doch auch dieses Verfahren ist nicht die Ultima Ratio.

Analog zur serialisierten V3-Verbuchung müssen Sie auch hier einen Sammellauf einplanen, konkret den Report RSM13005, der die V3-Verbuchungseinträge abarbeitet.

### Delta Direkt

Mit Erscheinen des Service-API-Releases PI\_BASIS 2003.1 wurde neben der unserialisierten V3-Verbuchung auch Delta Direkt zur Verfügung gestellt. Delta Direkt hat seinen Namen dadurch erhalten, dass die Daten nicht mehr

über Verbuchungstabellen in die Delta-Queue gelangen, sondern direkt nach der Entstehung in der Delta-Queue landen. Abbildung 3.15 stellt Delta Direkt grafisch dar.



**Abbildung 3.15** Verbuchungsmethode »Delta Direkt«

Man hatte aus den Schwächen der V3-Verbuchungsmethoden gelernt und zuerst die Abhängigkeit von der V2-Verbuchung eliminiert. Ein entstehender Beleg (1) landet also zu Beginn über eine V1-Verbuchung in den Anwendungstabellen und gelangt im gleichen Zug, ebenfalls über eine V1-Verbuchung, direkt in die Delta-Queue (2). Man spricht in diesem Zusammenhang auch von einer »aufgebohrten V1-Verbuchung«. Die Daten gelangen also ohne Umweg direkt nach ihrer Entstehung in die Delta-Queue, es ist kein einzuplanender Sammellauf mehr notwendig, der die Daten in die Delta-Queue schreibt. Anschließend können sie über eine Delta-Anforderung (3) in das angeschlossene SAP NetWeaver BW-System transferiert werden.

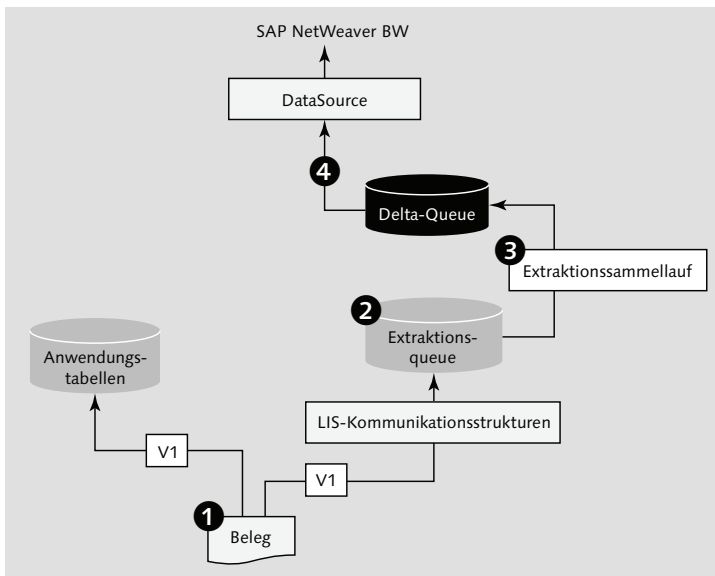
Obwohl die Schwäche der Abhängigkeit von der V2-Verbuchung nicht mehr existiert, hat Delta Direkt dennoch Schwierigkeiten im Umgang mit großen Datenmengen. Pro entstehendem neuen oder geänderten Datensatz wird eine neue LUW in der Delta-Queue aufgemacht. Sie erinnern sich an Abschnitt 3.1, »Delta-Queue in der Service-API«, hier haben wir erwähnt, dass der Sonderfall Logistik in der Verbuchungsmethode Delta Direkt dafür sorgt, dass in der RSA7 die exakte Anzahl Datensätze ausgewiesen wird. Dies liegt daran, dass jeder neue Datensatz eine neue LUW erzeugt.

SAP empfiehlt den Einsatz der Methode Delta Direkt bis zu einem aufkommenden Datenvolumen von 10.000 Datensätzen zwischen zwei Delta-Anfor-

derungen. Wenn Sie zum Beispiel täglich Daten extrahieren, sollten in diesem Abstand von einem Tag nur maximal 10.000 Datensätze in der Delta-Queue in Form von LUWs stehen. Bei (deutlich) größeren Datenmengen kann es zu Abbrüchen bei der Extraktion kommen.

### Delta Queued

Eine Alternative zur Beschränkung der geringen Sätze pro Datenübertragung ist die Verbuchungsmethode Delta Queued, die auch mit großen Datenmengen zurechtkommt. Abbildung 3.16 demonstriert die Vorgehensweise von Delta Queued.



**Abbildung 3.16** Verbuchungsmethode »Delta Queued«

Das Schaubild hat in etwa die Form der (serialisierten und unserialisierten) V3-Verbuchung, auch hier wird wieder mit einem Zwischenobjekt gearbeitet, der sogenannten *Extraktionsqueue*. Ein entstehender Beleg **(1)** wird über eine V1-Verbuchung in die Anwendungstabellen geschrieben. Er landet aber auch in der Extraktionsqueue, einem Zwischenpuffer für alle entstehenden Belege **(2)**. Auch diese Verbuchung ist eine V1-Verbuchung, sodass Datenkonsistenz gewährleistet werden kann. Die Daten werden dann über einen Extraktionssammellauf in die Delta-Queue übernommen, pro Sammel-lauf entsteht eine LUW in der Delta-Queue. Die Anzahl der LUWs in der Delta-Queue sinkt im Vergleich zu Delta Direkt also deutlich.

Für Delta Queued ist also im Gegensatz zu Delta Direkt weiterhin ein Sammellauf erforderlich, den Sie einplanen müssen. Diesen können Sie direkt aus dem Extraktstrukturen-Customizing-Cockpit einplanen. Der für die V3-Verbuchung zuständige Report RSM13005 führt hier zu keinem Ergebnis, da er ausschließlich V3-Verbuchungseinträge abarbeitet. SAP empfiehlt die stündliche Einplanung des Sammellaufs für das Verfahren Delta Queued.

#### Zeitliche Einplanung des Sammellaufs

Bitte beachten Sie unbedingt folgenden Aspekt: Planen Sie den Sammellauf grundsätzlich erst dann ein, wenn die Delta-Initialisierung erfolgreich abgeschlossen wurde. Gerade wenn Sie über einen Neuaufbau nachdenken, wird der Schritt des Ausplanens des Sammellaufs gerne vergessen. Bei aktivem Sammellauf ohne erfolgreiche Delta-Initialisierung steht die Delta-Queue noch nicht bereit, die neuen und geänderten Datensätze landen im Nirwana und gehen für das BW-System verloren!

Wir empfehlen Ihnen, unbedingt SAP-Hinweis 505700 zu lesen, der den Umgang mit den drei Verbuchungsmethoden unserialisierte V3-Verbuchung, Delta Direkt und Delta Queued erläutert und Handlungsempfehlungen gibt. Der Hinweis ist releaseunabhängig und auch heute noch gültig, er beinhaltet alle wesentlichen Aspekte.

#### Logistik-Datenextraktion und Realtime Data Acquisition

Wenn Sie die Realtime Data Acquisition (RDA) im Zusammenhang mit der Logistik-Extraktion verwenden möchten, ist dies nur dann sinnvoll, wenn Sie mit der Verbuchungsmethode Delta Direkt arbeiten. In den Verfahren Delta Queued und der unserialisierten V3-Verbuchung sind Sammelläufe dafür verantwortlich, die Daten in die Delta Queue zu schreiben. Wenn Sie die Daten minütlich über RDA extrahieren wollen, der Sammellauf aber nur zweimal am Tag eingeplant ist, dürfen Sie kein Realtime-Szenario erwarten. Die Delta-Queue ist schlichtweg einen halben Tag lang leer.

Wir beenden damit die Logistik-Datenextraktion und werden uns nun mit der generischen Datenextraktion und ihrem Zusammenhang mit den Delta-Mechanismen beschäftigen.

## 3.3 Generisches Delta

In diesem Abschnitt behandeln wir das generische Delta, also die Implementierung eigener Delta-Verfahren. Wir werden im ersten Schritt die Funktionsweise des generischen Deltas vorstellen und im Anschluss ein paar Sätze

# Index

ORECORDMODE 14, 20, 117, 140, 153,  
154, 201, 210  
3.x-DataSources 37  
7.x-DataSources 37

## A

---

ABAP Objects 187  
Ablauflogik 88  
Addition 42, 177  
additives Delta 72  
additives Image 32, 43, 46, 104, 140,  
177  
Ad-hoc-Auswertungen 162  
After Image 32, 41, 43, 46, 140, 177  
Aggregate 149  
Aggregate-Rollup 172  
Aggregation 157  
Aggregationsart 47, 104, 112  
Aggregationsverhalten 41, 176  
aktive Daten 143, 209  
Aktive-Daten-Tabelle 19, 44, 98, 141,  
155  
Aktivierung 19, 152  
Aktivierungslauf 143, 150, 152, 210  
Aktivierungs-Queue 19, 141, 143, 209  
ALE-Delta 96  
ALE-Fortschreibungszeiger 30, 93, 150,  
200  
ALV Grid 183  
Analyseprozessdesigner 19  
Änderungsbelegobjekt 94  
Änderungsbelegfortschreibung 94  
Änderungszeiger 95  
Änderungszeitstempel 71  
Anreicherung 158, 161  
Anwendungskomponente 101, 137  
Anwendungs-Log 174  
APD 19  
APO-Plug-In Bestellung 78  
Application Management 169  
Applikation 59, 161, 164  
Applikationsserver 140

ARFCSDATA 54, 211  
ARFCSSTATE 54, 211  
ASSIGN\_TYPE\_CONFLICT 176  
Attributsänderungslauf 149  
Auswirkungen 42

## B

---

Before Image 32, 41, 46, 140  
Belegarten 164  
Berechtigungen 189  
Berechtigungsobjekt 202  
Bestätigungstabelle 133, 154, 200  
Betrieb 14  
Bewegungsdaten 21, 101  
BI Java Software Development Kit 107  
BI-Java-Konnektor 106  
Breakpoint 181, 184, 186, 194  
Business Content 21, 36, 38, 49, 99  
Business Transaction Event 77, 200  
Business Transformation Layer 161, 162  
BW-Zielsystem 50

## C

---

CHANGE\_POINTERS\_READ 95  
Change-Log 46, 112, 141, 143, 154, 165,  
167  
Change-Log-Tabelle 19, 201  
CIF → Corporate Information Factory  
CL\_RSTRAN\_TRFN\_CMD 188  
Codepage 177  
convert-Unterprogramme 195  
Corporate Information Factory 159  
Corporate Memory 157, 161, 163, 164  
CTBW\_GET\_KONFIG\_DATA 92  
CUBE 112  
Customizing Cockpit 59  
CX\_SY\_OPEN\_SQL\_DB 175  
CX\_SY\_PROVIDE\_INTERVAL\_OVERLAP  
176



## D

---

Daemon 98, 132, 200  
 Data Acquisition Layer 160, 165  
 Data Mart 14, 158, 162  
 Data Mart Interface 110, 201  
 Data Propagation Layer 161  
 Data Services 14, 112  
 Data Warehouse 158, 161  
 Data-Mart-Szenarien 148  
 DATAPAKID 210  
 DataSource 14, 20, 28, 29, 46, 79, 110, 137, 211  
 DataSource-Typ 37  
 DataStore-Objekt 18, 43, 111, 133  
     *für direktes Schreiben* 18  
 Dateischnittstelle 14, 100, 200  
 Datenanforderung 14, 22  
 Datenbank 131  
 Datenbanktabelle 15, 18, 102  
 Datenbank-View 79  
 Dateneingang 160  
 Datenfluss 21  
 Datenformat 128  
 Datenkonsistenz 73, 151  
 Datenlauf 89  
 Datenmodell 26, 155  
 Datenpaket 25, 156, 181  
 Datenpaketgröße 89  
 Datenpaketnummer 210  
 Datensatznummer 156  
 Datenselektion 118  
 Datentransferprozess 13, 14, 22, 25, 45, 46, 125, 153, 181, 201  
 Datenübertragung 153  
 Datenverarbeitung 160  
 Datenverbuchung 22  
 Datenziel 12, 18, 24, 46, 138, 151, 158  
 DB Connect 14, 100, 105, 200  
 DBIF\_RSQL\_SQL\_ERROR 175  
 DD03L 209, 211  
 Deadlock 175  
 Debugger 184, 186, 194  
     *klassischer* 186  
     *neuer* 186  
 Debugging 14, 169, 172, 184, 202  
 Debugging-Berechtigung 189  
 Delete Image 33, 44, 46, 140, 153, 210  
 Delta Direkt 53, 66

Delta Images 104  
 Delta mit After Images 103  
 Delta mit Delta Images 103  
 Delta nur einmal abholen 127, 151  
 Delta nur über Full-Upload 103  
 Delta Queued 68  
 Delta-Anforderung 60  
 deltabestimmendes Feld 71, 137, 200  
 Delta-Fähigkeit 14, 36, 41, 47, 114  
 deltaharmonisierende Schicht 163, 167, 201  
 Delta-Harmonisierung 14, 45, 46, 112, 141, 142, 154, 155, 166, 201  
 DELTAINIT 52  
 Delta-Initialisierung 14, 52, 57, 60, 85, 89, 120, 137, 153, 199, 201  
     *mit Daten* 117, 122, 137  
     *ohne Daten* 117, 123, 128  
 Delta-Integration 167  
 Delta-Management 11, 15  
 Delta-Queue 29, 50, 58, 60, 65, 78, 104, 114, 120, 133, 154, 176, 199, 211  
 deltarelevantes Feld 141  
 Delta-Typ 28, 40, 46, 63, 104, 199  
 Delta-Update 14, 89, 117, 124, 154, 178, 201  
 Delta-Verfahren 13, 15, 17, 21, 28, 46, 76, 101, 102, 109, 112, 154, 161, 162, 167, 173, 199  
 Detailanalyse 172, 201  
 Digestible Data 161  
 direkte Fortschreibung 191  
 Direktzugriff 102  
 Dokumentation 163  
 Domäne 21, 91  
 DSO → DataStore-Object  
 DTP → Datentransferprozess  
 DTP-Monitor 185

## E

---

Early Watch Alert 149  
 Early-Delta-Initialisierung 63, 117, 123, 154  
 echtes Delta 26, 46  
 EDW 159  
 eigene Stornobelege 201  
 eigene Stornosätze 151

Eigenentwicklung 87  
 Einführungsleitfaden 49  
 Einzelschritt 187  
 Endroutine 172  
 Enterprise Data Warehouse 159  
 ERROR\_PASSED\_TO\_MESS\_HANDLER  
     90  
 Event 77  
 EXECUTE 188  
 Expertenmodus 182  
 Expertenroutine 172, 188  
 Export-DataSource 110  
 Extraktionsmethode 37, 70  
 Extraktionsperformance 199  
 Extraktions-Queue 68  
 Extraktions-Sammellauf 68  
 Extraktor 29, 37  
 Extraktstruktur 20, 37, 59, 85, 93, 94,  
     111  
 Extraktstrukturen-Customizing-Cockpit  
     69

## F

---

Fehler 201  
 Fehleranalyse 14, 169  
 Fehlerbehandlung 23, 46, 169  
 Fehler-DTP 24, 130  
 Fehlermanagement 13, 23, 125, 128,  
     133, 169  
 Fehlerprotokolle 188  
 Fehlerstack 23, 46, 126, 128, 130, 171,  
     173  
 Fehlerstack-Tabelle 131  
 Feld DELTA 36  
 Feldliste 101, 102, 113  
 Feldroutine 172  
 Feldsymbol 176, 188, 205, 209  
 FI 33, 73  
 FILO 103, 113  
 FIL1 103, 113  
 Filter 182  
 Flatfile 100  
 Flatfile-Delta 31  
 flexible Fortschreibung 191  
 Format 103  
 Fortschreibung 118

Fortschreibungsmodi 47, 89, 91, 117,  
     153  
 Fortschreibungsregel 112, 190, 195  
 Frequenz 139  
 Full-Only-Verfahren 163  
 Full-Repair-Request 123  
 Full-Update 11, 14, 28, 60, 89, 117, 118,  
     153, 179, 210  
 Funktionsbaustein 21, 77, 87, 108

## G

---

generiertes Programm 188, 194  
 generische DataSource 14, 21, 38, 137  
 generische Extraktoren 200  
 generisches Delta 30, 69, 70, 100, 200  
 GETWA\_NOT\_ASSIGNED 176

## H

---

Harmonisierung 158, 161, 164  
 Harmonization Layer 163  
 HCM 27  
 Hierarchien 21, 101  
 Hintergrundprogramm 98, 132, 137  
 Hintergrundprozess 140  
 Historie 148, 150, 151, 157, 161, 205

## I

---

IBM DB2/390 105  
 IBM DB2/400 105  
 IBM DB6 105  
 IF\_RSBK\_CMD\_T 188  
 Image 40, 140  
 Importing-Parameter  
     I\_CALLMODE 92  
     I\_DATAPAKID 91  
     I\_PRIVATE\_MODE 92  
     I\_REQUNR 90  
 InfoCube 41, 43, 141, 155, 162  
 InfoObject 149, 190  
 InfoPackage 13, 14, 22, 38, 46, 104,  
     108, 118, 137, 178, 190, 202  
 InfoProvider 17, 46, 151  
     ohne Persistenz 18

Information Lifecycle Management 149  
Informix 105  
InfoSet 106, 161  
InfoSource 166, 185  
InfoSource 3.x 112, 190  
Initialisierungslauf 88  
Initialisierungsselektionen 121  
INITRNR 51  
INITSTATE 52  
Inkonsistenz 150, 176, 178  
Insellösungen 159  
Insert only 34  
Invertierung 33

## J

---

Java Database Connectivity 106  
JDBC-Konnektor 106

## K

---

Kalendertag 71, 200  
keine Serialisierung 34  
Klasse 188  
Kommunikationsfehler 177  
Komprimierung 127  
Konsolidierung 25  
Konvertierungsroutine 103, 189  
kundeneigener Namensraum 19  
Kurzdump 171, 173, 174

## L

---

Langtext 41  
Langzeitspeicherung 160  
Laufzeitfehler 175  
Layered, Scalable Architecture 14, 26,  
155, 159, 162, 167, 201  
LCL\_TRANSFORM 188  
LIS 59  
LIS-Kommunikationsstruktur 59, 64  
LO-Customizing-Cockpit 60  
Logical Unit of Work 52, 65, 67, 68, 211  
Logistik 180, 199  
Logistik-Datenextraktion 58, 119  
Logistik-Informations-System 59

Löschen der Initialisierungsselektion  
122  
Löschung 33  
LRSSMU36 176  
LSA → Layered, Scalable Architecture  
LUW → Logical Unit of Work

## M

---

Max DB 105  
Maximum 42  
MESSAGE\_TYPE\_X 175  
Microsoft SQL Server 105  
Minimum 42  
Modellierung 31, 201  
Monitor 173  
MultiProvider 161, 170  
Musterfunktionsbaustein 79, 86

## N

---

Nachrichtenmonitor 193  
Nachrichtentypen 95  
Namenskonvention 19, 209  
Near Line Storage 149, 162  
Neuaufbautabelle 61, 114, 119, 180,  
199  
Neue-Daten-Tabelle 19, 143, 209  
Neuzustand 103  
*für veränderte Datensätze* 72  
New Image 32, 41, 42, 46, 140  
NLS 157  
NO\_MORE\_DATA 90  
Notfallbenutzer 189  
numerischer Zeiger 71, 200  
Nummernkreis 95  
Nummernkreisobjektpflege 95

## O

---

Obergrenze 73  
Objektversion 36  
OBJVERS 149  
ODBO-Konnektor 106  
ODS → Operational Data Store  
offener Request 97, 109, 132, 200

OLE DB for OLAP 106  
 Operational Data Store 112, 162  
 ORA-00060 175  
 Oracle 105

## P

---

P&S 77  
 Paket 190  
 Paketgröße 25  
 Parameter 90, 153  
 Passwort 177  
 Persistent Staging Area 108, 157, 160, 190  
 persistente InfoProvider 18  
 Persistenz 161  
 Personalmanagement 87  
 Primärschlüssel 156  
 Produkt 77, 82  
 Programm 152, 194, 205  
 Propagator 161  
 Prozesskette 98, 124, 151, 152  
 Prozessschnittstelle 77  
 Prozessübersicht 140  
 PSA → Persistent Staging Area  
 Pseudo-Delta 17, 26, 27, 46, 163  
 Publish&Subscribe 77  
 Pull-Mechanismus 108  
 Pull-Verfahren 29, 50  
 Push-InfoPackages 108  
 Push-Mechanismus 107  
 Push-Verfahren 29, 41, 50, 52

## Q

---

QNAME 54  
 qRFC 53  
 Quality and Harmonization Layer 160  
 Quellsystem 14, 200  
 Quellsystemverbindung 177  
 Query 18, 170

## R

---

Ralph Kimball 159  
 Range 93

RAW-Format 55  
 RDA → Realtime Data Acquisition  
 RDA-Daemon 139  
 RDA-Verfügbarkeit 98  
 RDBMS 105  
 REALTIME 98  
 Realtime Data Acquisition 14, 69, 72, 97, 108, 132, 154, 200  
 Realtime Data Acquisition InfoPackage 138  
 Realtime-fähig 37, 72, 100  
 Realtime-Reporting 26  
 RECORD 210  
 Record Mode 28  
 Referenzarchitektur 159  
 Regel-ID 188  
 Reihenfolge 28, 34, 178  
 Reparatur-Full-Request 14, 119, 179, 201  
 Repeat-Delta 14, 56, 97, 150, 178, 201  
 Reporting 129, 132  
 Reporting Layer 161, 162  
 Request 24, 108, 135, 143, 171  
 requestbasiertes Delta 126  
 Request-ID 127, 156  
 Request-Nummer 51, 145, 210  
 Reverse Image 33, 41, 44, 46, 140  
 Revisionssicherheit 131  
 RFC 53  
 RFC-Verbindung 177  
 ROCANCEL 14, 140, 154, 201  
 RODELTAM 28, 35, 36, 40, 47, 49, 60, 76, 93, 104, 148, 199, 211  
 Rohdaten 167  
 ROOSOURCE 36, 37, 47, 72, 81, 98, 211  
 ROOSPRMSC 51, 85, 211  
 rote Requests 173  
 Routine 172  
 RSA7 30  
 RSAX\_BIW\_GET\_DATA 88  
 RSAX\_BIW\_GET\_DATA\_SIMPLE 87  
 RSCRT\_CONFIRM 133  
 RSDS 36, 37, 47, 211  
 RSM13005 66  
 RSO2 38  
 RSOLTPSOURCE 36, 37, 47  
 RSTRAN\_MASTER\_TMPL 188  
 RSTRAN\_MASTER\_TMPL\_EXPERT 188  
 Rückkehr 187

## S

---

S\_DEVELOP 189  
 Sammellauf 64  
 SAP BusinessObjects 112, 200  
 SAP Developer Network 107  
 SAP ERP HCM 158  
 SAP InfoSet 21  
 SAP NetWeaver Application Server 53  
 SAP NetWeaver Application Server Java 106  
 SAP NetWeaver BW 17, 23, 98, 124, 125  
 SAP NetWeaver BW 7.30 98, 101, 112, 124, 131, 136, 186, 191, 200  
 SAP Queries 106  
 SAP-Bibliothek 38, 47, 53, 63, 136, 186  
 SAPI 49  
 SAPLRSSM 140, 176  
 SAP-Namensraum 19  
 SAP-Quellsystem 14, 50  
 SAP-Query-Konnektor 106  
 SAP-Standardfunktionsbaustein 85  
 SAP-Zeitmerkmal 195  
 Satztyp 20, 28, 31, 40, 46, 140, 145, 154, 199  
 Scheduler 119  
 Schema 14  
 Schlüssel 24, 41  
 schlüsselgleiche Sätze 42  
 Schlüsselvergleiche 209  
 schreiboptimiertes DSO 14, 18, 155, 160, 164, 167  
 Schrittnummer 188  
 Schwellwert 98, 109, 129, 131, 135, 138  
 segmentierte Daten 101  
 Select-Options 93  
 Selektionsfelder 89, 92, 193  
 Selektionsintervall 73  
 Selektionsparameter 27, 80, 118, 120, 123  
 semantische Gruppen 24, 125  
 serialisierte V3-Verbuchung 64  
 Serialisierung 28, 34, 46, 199  
 Serialisierung der Datenpakete 34, 41  
 Serialisierung der Requests 34  
 Service 159  
 Service-API 36, 57, 61, 64, 66, 124  
 Servicename 108

Sicherheitsintervalle 70, 72, 73  
 SID 210  
 Simulation 14, 169, 177, 180, 190, 191, 201  
     *der Verbuchung* 194  
 Simulationsrequest 184  
 Simulieren 182  
 Snapshot-Delta 127  
 Sofortstart 138  
 Speicherplatzbedarf 148  
 Speicherverteilung 131  
 Sprachenabhängigkeit 66  
 Sprachwechsel 65  
 sRFC 53  
 Stabilität 27  
 Stammdaten 149, 191  
 Stammdatenattribute 21, 96, 101  
 Stammdatenextraktion 30, 95  
 Stammdatentexte 21, 101  
 Standard-DSO 13, 18, 45, 46, 98, 141, 142, 151, 155, 161, 164, 167, 201  
 Standard-DSO-Schicht 158  
 Standard-DTP 138  
 Standardisierung 163  
 Startroutine 172, 193, 194  
 Status 54  
 Stornierung 33  
 Stornobelege 205  
 Stornodatensätze 14  
 Stornoinformationen 149  
 Stornosätze 150, 154, 210  
 System-Log 171, 174  
 Systemverbindung 177

## T

---

Tabelle 21  
 Tabellenkategorie 131  
 Tablespace 131  
 Tables-Parameter  
     *E\_T\_DATA* 92  
     *E\_T\_SOURCE\_STRUCTURE\_NAME* 93  
     *I\_T\_FIELDS* 92  
     *I\_T\_SELECT* 92  
 TCDOB 94  
 technische Gruppe 140  
 technischer Schlüssel 19, 155

TID 53  
 time\_conversion 195  
 TLOGO-Objekt 22  
 TLOGO-Typ 22, 37  
 Top-down-Analyse 170, 201  
 TOP-Include 90  
 Transaktionskennung 53  
 Transaktionssperre 62, 123  
 Transferstruktur 193  
 TRANSFORM 188  
 Transformation 45, 111, 125, 138, 172  
 Transformations-ID 188  
 tRFC 53  
 TRFCQOUT 53, 211

## U

---

Überschreiben 43, 177  
 Übertragungsregel 112, 190  
 UD Connect 107  
 UNCAUGHT\_EXCEPTION 176  
 unechtes Delta 26  
 Universal Data Connect 107  
 Universal Data Integration 14, 100, 105, 200  
 unserialisierte V3-Verbuchung 66  
 Untergrenze 73, 75  
 Unternehmensgedächtnis 161  
 Unterprogramm 194  
 update\_infocube 195  
 URL 107  
 UTC-Zeitstempel 71

## V

---

V1-Verbuchung 63, 64, 67  
 V2-Verbuchung 63, 65  
 V3-Sammellauf 63, 65, 66  
 V3-Verbuchung 63  
 Variante 153  
 Verarbeitung 12  
 Verarbeitungsmodus 128  
 Verbindungsabbruch 171, 201  
 Verbindungsprobleme 169, 177  
 Verbuchung 129, 184  
 Verbuchung simulieren 192  
 verbuchungsfreie Zeit 62, 123, 173

Verbuchungsmethode 60, 63, 199  
 Verbuchungsprozesse 63  
 Verbuchungstabellen 64  
 Verdichtung 127  
 Verfahren ODS 46  
 Version 149  
 View 21, 102  
 VirtualProvider 102  
 Vorgehensmodell 170, 201  
 Vorlage-InfoObjects 102  
 Vorschau 103  
 Vorschlag 102, 104, 113

## W

---

Währungsumrechnungen 128  
 Warehouse Management 28  
 Warengruppenschlüssel 164  
 Wartbarkeit 97  
 Webservice 14, 107, 200  
 Workaround 99

## X

---

XML for Analysis 106  
 XMLA-Konnektor 106  
 XML-DataSource 107  
 X-Tabelle 86

## Y

---

Y-Tabelle 86

## Z

---

Zeiger 176  
 Zeitfenster 26, 122  
 Zeitstempel 71, 74, 200  
 ZGP\_STORNO\_CREATE 153, 205  
 Zielgruppen 15  
 Zielstruktur 188  
 Zwischenschritt 193  
 Zwischenspeicher 23, 119, 129, 182, 193