

IBM DB2 Universal Database



# Versetzen von Daten Dienstprogramme und Referenz

*Version 7*



IBM DB2 Universal Database



# Versetzen von Daten Dienstprogramme und Referenz

*Version 7*

**Hinweis**

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter „Anhang F. Bemerkungen“ auf Seite 337 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs  
*IBM DB2 Universal Database. Data Movement Utilities Guide and Referene*,  
IBM Form SC09-2955-00,

herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2000  
© Copyright IBM Deutschland Informationssysteme GmbH 2000

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:  
SW NLS Center  
Kst. 2877  
April 2000

---

# Inhaltsverzeichnis

<b>Zu diesem Handbuch</b> . . . . .	<b>vii</b>	<b>IMPORT - Befehl</b> . . . . .	<b>41</b>
Zielgruppe . . . . .	viii	Import - API . . . . .	49
Aufbau dieses Handbuchs . . . . .	viii	Datenstruktur SQLUIMPT-IN . . . . .	60
<b>Kapitel 1. EXPORT</b> . . . . .	<b>1</b>	Datenstruktur SQLUIMPT-OUT . . . . .	61
EXPORT - Übersicht . . . . .	2	Dateitypwerte (IMPORT). . . . .	63
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT . . . . .	3	Hinweise zu Zeichensatz und NLS . . . . .	74
Verwenden von EXPORT . . . . .	3	Beispiele für IMPORT-Sitzungen . . . . .	75
Vor der Verwendung von EXPORT . . . . .	3	CLP-Beispiele . . . . .	75
Aufrufen von EXPORT . . . . .	3	API-Beispiele. . . . .	77
Verwenden von EXPORT mit Identitätsspalten	4	Optimieren der IMPORT-Leistung. . . . .	77
Erneutes Erstellen einer exportierten Tabelle. . . . .	4	Rahmenbedingungen und Einschränkungen	78
Exportieren von großen Objekten (LOBs). . . . .	5	Fehlerbehebung . . . . .	78
Paralleles Exportieren von Daten . . . . .	5	<b>Kapitel 3. LOAD</b> . . . . .	<b>79</b>
EXPORT - Befehl . . . . .	9	LOAD - Übersicht . . . . .	80
EXPORT - API . . . . .	12	Änderungen der früheren Funktionsweise des Dienstprogramms LOAD (eingeführt in Version 6 und 7). . . . .	86
Datenstruktur SQLUEXPT-OUT . . . . .	18	Parallelität und Laden . . . . .	88
Dateitypwerte (EXPORT). . . . .	19	Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD . . . . .	89
Einschränkungen für Begrenzer . . . . .	20	Verwenden von LOAD . . . . .	90
Beispiele für EXPORT-Sitzungen . . . . .	22	Vor der Verwendung von LOAD . . . . .	90
CLP-Beispiele . . . . .	22	Aufrufen von LOAD . . . . .	90
API-Beispiele. . . . .	23	Verwenden von LOAD mit Identitätsspalten	91
Einschränkungen . . . . .	27	Verwenden von LOAD mit generierten Spalten . . . . .	93
Fehlerbehebung . . . . .	27	Überprüfung auf ungültige Integritätsbedingungen. . . . .	96
<b>Kapitel 2. IMPORT</b> . . . . .	<b>29</b>	Neustart einer unterbrochenen Ladeoperation	98
IMPORT - Übersicht . . . . .	30	Verwenden der Datei mit den Angaben zur Speicherposition der Ladekopie . . . . .	98
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT . . . . .	31	LOAD - Befehl. . . . .	101
Verwenden von IMPORT. . . . .	32	LOAD QUERY - Befehl . . . . .	116
Vor der Verwendung von IMPORT . . . . .	32	LOAD - API . . . . .	118
Aufrufen von IMPORT . . . . .	32	Datenstruktur SQLULOAD-IN . . . . .	128
Verwenden von IMPORT in einer Client/Server-Umgebung . . . . .	33	Datenstruktur SQLULOAD-OUT. . . . .	133
Verwenden von IMPORT mit gepufferten INSERT-Operationen . . . . .	34	db2LoadQuery - API Load Query . . . . .	135
Verwenden von IMPORT mit Identitätsspalten	34	Dateitypwerte (LOAD) . . . . .	140
Verwenden von IMPORT mit generierten Spalten. . . . .	36	Ausnahmetabelle . . . . .	157
Erneutes Erstellen einer exportierten Tabelle	38	Speicherausgangsdatei . . . . .	158
Importieren von großen Objekten (LOBs) . . . . .	39	Temporäre Dateien (LOAD) . . . . .	158
Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs). . . . .	40	Protokollsätze zum Dienstprogramm LOAD	159
Sperren von Tabellen während des Imports	40	Zeichensatz und NLS . . . . .	160
		Beispiele für LOAD-Sitzungen . . . . .	160

CLP-Beispiele . . . . .	160	Beispiele zum Versetzen von Daten zwischen typisierten Tabellen . . . . .	223
API-Beispiele . . . . .	166	Verwenden der Replikation zum Versetzen von Daten . . . . .	225
Statusangaben für anstehende Aktionen nach einer Ladeoperation . . . . .	173	Die IBM Replikations-Tools nach Komponenten . . . . .	227
Optimieren der Leistung von LOAD . . . . .	174	Versetzen von Daten mit der Data Warehouse-Zentrale . . . . .	228
Rahmenbedingungen und Einschränkungen	180		
Fehlerbehebung . . . . .	180		
<b>Kapitel 4. AutoLoader . . . . .</b>	<b>181</b>	<b>Anhang A. Lesen von Syntaxdiagrammen</b>	<b>231</b>
AutoLoader - Übersicht . . . . .	181	<b>Anhang B. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD . . . . .</b>	<b>235</b>
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von AutoLoader . . . . .	182	<b>Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD. . . . .</b>	<b>239</b>
Verwenden von AutoLoader . . . . .	183	ASCII-Format ohne universelle Zeilenbegrenzer (DEL) . . . . .	240
Vor der Verwendung von AutoLoader	183	Beispiel für DEL-Datei . . . . .	242
Aufrufen von AutoLoader . . . . .	183	Beschreibungen von DEL-Datentypen . . . . .	244
Laden in mehrere Datenbankpartitionen . . . . .	184	ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC) . . . . .	248
AutoLoader-Optionen . . . . .	185	Beispiel für ASC-Datei . . . . .	249
AutoLoader-Beispielsitzung . . . . .	194	Beschreibungen von ASC-Datentypen . . . . .	250
Migration und Kompatibilität zur früheren Version . . . . .	196	PC-Version des IXF-Dateiformats. . . . .	253
Hinweise und Tips für das Dienstprogramm AutoLoader. . . . .	197	PC/IXF-Satzarten . . . . .	255
Rahmenbedingungen und Einschränkungen	199	PC/IXF-Datentypen . . . . .	277
Fehlerbehebung für AutoLoader . . . . .	199	Beschreibung von PC/IXF-Datentypen	285
		Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken. . . . .	292
		Datentypspezifische Regeln für den Import von PC/IXF-Dateien in Datenbanken . . . . .	294
		Option FORCEIN . . . . .	298
		Unterschiede zwischen PC/IXF und Version 0 System /370 IXF. . . . .	308
		WSF (Worksheet File Format) . . . . .	309
<b>Kapitel 5. Versetzen von DB2 Data Links Manager-Daten . . . . .</b>	<b>201</b>	<b>Anhang D. Warnungen, Fehler- und Beendigungsnachrichten. . . . .</b>	<b>311</b>
Versetzen von DB2 Data Links Manager-Daten mit EXPORT . . . . .	201	<b>Anhang E. Verwenden der DB2-Bibliothek</b>	<b>313</b>
Versetzen von DB2 Data Links Manager-Daten mit IMPORT . . . . .	206	PDF-Dateien und gedruckte Bücher für DB2	313
Versetzen von DB2 Data Links Manager-Daten mit LOAD . . . . .	207	Informationen zu DB2 . . . . .	313
		Drucken der PDF-Handbücher . . . . .	326
		Bestellen der gedruckten Handbücher . . . . .	327
		DB2-Online-Dokumentation . . . . .	329
		Zugreifen auf die Online-Hilfefunktion	329
		Anzeigen von Online-Informationen . . . . .	331
		Verwenden der DB2-Assistenten . . . . .	333
<b>Kapitel 6. Versetzen von Daten zwischen Systemen . . . . .</b>	<b>209</b>		
Versetzen von Daten zwischen Plattformen	209		
PC/IXF-Dateiformat . . . . .	209		
ASCII-Format ohne universelle Zeilenbegrenzer (DEL) . . . . .	210		
WSF-Dateiformat . . . . .	211		
Versetzen von Daten mit dem Tool db2move	212		
Versetzen von Daten mit DB2 Connect . . . . .	217		
Verwenden der Dienstprogramme EXPORT und IMPORT . . . . .	217		
Versetzen von Daten zwischen typisierten Tabellen . . . . .	219		
Traversierfolge . . . . .	221		
Datenauswahl beim Versetzen von Daten	222		

Einrichten eines Dokument-Servers . . .	335	<b>Index . . . . .</b>	<b>343</b>
Suchen nach Online-Informationen . . .	336		
<b>Anhang F. Bemerkungen . . . . .</b>	<b>337</b>	<b>Kontaktaufnahme mit IBM . . . . .</b>	<b>347</b>
Neue deutsche Rechtschreibung . . . . .	340	Produktinformationen . . . . .	347
Änderungen in der IBM Terminologie . . .	340		
Marken . . . . .	341		





---

## Zu diesem Handbuch

Dieses Handbuch bietet Informationen zu den folgenden IBM DB2 Universal Database (UDB)-Dienstprogrammen zum Versetzen von Daten und ihrer Verwendung:

- Mit Hilfe der Dienstprogramme IMPORT und EXPORT werden Daten zwischen einer Tabelle oder Sicht und einem anderen Datenbank- oder Tabellenkalkulationsprogramm, zwischen DB2-Datenbanken sowie zwischen DB2-Datenbanken und Host-Datenbanken mit Hilfe von DB2 Connect versetzt. Mit dem Dienstprogramm EXPORT können Daten von einer Datenbank in Betriebssystemdateien versetzt werden. Diese Dateien können Sie dann verwenden, um diese Daten in eine andere Datenbank zu importieren oder zu laden.
- Mit dem Dienstprogramm LOAD können Daten in Tabellen versetzt, vorhandene Indizes erweitert und Statistikdaten generiert werden. Mit LOAD werden die Daten viel schneller als mit dem Dienstprogramm IMPORT versetzt, wenn es sich um umfangreiche Datenmengen handelt. Daten, die mit dem Dienstprogramm EXPORT aus Tabellen geladen wurden, können mit dem Dienstprogramm LOAD wieder geladen werden.
- Mit dem Dienstprogramm AutoLoader können umfangreiche Mengen von Daten geteilt und die geteilten Daten in verschiedene Partitionen einer partitionierten Datenbank geladen werden.
- DataPropagator (DPROP) ist eine Komponente von DB2 Universal Database, die das automatische Kopieren von Tabellenaktualisierungen in andere Tabellen anderer relationaler DB2-Datenbanken ermöglicht.
- Mit Hilfe der Data Warehouse-Zentrale können Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden.

Fremdprodukte zum Versetzen von Daten in und aus Datenbanken stehen ebenfalls zur Verfügung, allerdings werden sie im vorliegenden Handbuch nicht näher behandelt.

---

## Zielgruppe

Dieses Handbuch richtet sich an Datenbankadministratoren, Anwendungsprogrammierer und andere DB2 UDB-Benutzer, welche die folgenden Aufgaben ausführen:

- Laden von Daten in DB2-Tabellen aus Betriebssystemdateien
- Versetzen von Daten zwischen DB2-Datenbanken und zwischen DB2 und anderen Anwendungen (z. B. Tabellenkalkulationsprogramme)
- Archivieren von Daten

Es wird vorausgesetzt, daß Sie mit DB2 Universal Database, Structured Query Language (SQL) und der Betriebssystemumgebung vertraut sind, in der DB2 UDB ausgeführt wird. Basisinformationen zu DB2 UDB finden Sie im Handbuch *Systemverwaltung*. Informationen zu SQL finden Sie im Handbuch *SQL Reference*. Informationen zur Konfiguration, zum Aufruf und zur Verwendung des DB2 UDB-Befehlszeilenprozessors finden Sie im Handbuch *Command Reference*. Informationen zu den DB2 UDB-Anwendungsprogrammierschnittstellen (APIs) finden Sie im Handbuch *Administrative API Reference*. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, finden Sie im Handbuch *Application Building Guide*. Dieses Handbuch enthält keine Anleitungen zur Installation von DB2, da diese vom Betriebssystem abhängt. Installationsanleitungen finden Sie im Handbuch *Einstieg* für Ihr Betriebssystem.

---

## Aufbau dieses Handbuchs

Die folgenden Themen werden behandelt:

### **Kapitel 1**

Beschreibt das DB2-Dienstprogramm EXPORT, mit dem Daten aus DB2-Tabellen in Dateien versetzt werden.

### **Kapitel 2**

Beschreibt das DB2-Dienstprogramm IMPORT, mit dem Daten aus Dateien in DB2-Tabellen oder -Sichten versetzt werden.

### **Kapitel 3**

Beschreibt das DB2-Dienstprogramm LOAD, mit dem große Datenmengen aus Dateien in DB2-Tabellen versetzt werden.

### **Kapitel 4**

Beschreibt das Dienstprogramm AutoLoader, mit dem umfangreiche Mengen von Daten geteilt und die geteilten Daten in verschiedene Partitionen einer partitionierten Datenbank geladen werden können.

**Kapitel 5**

Beschreibt, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um DB2 File Manager-Daten zu versetzen.

**Kapitel 6**

Beschreibt, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um Daten zwischen verschiedenen Plattformen sowie in und aus DRDA-Host-Datenbanken zu übertragen. Auch DataPropagator (DPROP), eine weitere Methode zum Versetzen von Daten innerhalb eines Unternehmens, wird beschrieben. Dieses Kapitel enthält außerdem eine Einführung in die Data Warehouse-Zentrale, mit deren Hilfe Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden können.

**Anhang A**

Erläutert die in Syntaxdiagrammen verwendeten Konventionen.

**Anhang B**

Faßt wichtige Unterschiede zwischen den DB2-Dienstprogrammen LOAD und IMPORT zusammen.

**Anhang C**

Beschreibt externe Dateiformate, die von den Datenbankmanager-Dienstprogrammen EXPORT, IMPORT und LOAD unterstützt werden.

**Anhang D**

Bietet Informationen zur Auswertung von Nachrichten, die vom Datenbankmanager generiert werden, wenn eine Warn- oder Fehlerbedingung erkannt wird.



---

## Kapitel 1. EXPORT

In diesem Kapitel wird das DB2 UDB-Dienstprogramm EXPORT beschrieben, mit dem Daten aus einer DB2-Datenbank in eine oder mehrere Dateien geschrieben werden können, die außerhalb der Datenbank gespeichert werden. Die exportierten Daten können dann mit dem DB2-Dienstprogramm IMPORT bzw. LOAD in eine andere DB2-Datenbank importiert bzw. geladen werden, oder sie können in eine andere Anwendung (z. B. ein Tabellenkalkulationsprogramm) importiert werden.

Die folgenden Themen werden behandelt:

- „EXPORT - Übersicht“ auf Seite 2
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT“ auf Seite 3
- „Verwenden von EXPORT“ auf Seite 3
- „Verwenden von EXPORT mit Identitätsspalten“ auf Seite 4
- „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 4
- „Exportieren von großen Objekten (LOBs)“ auf Seite 5
- „Paralleles Exportieren von Daten“ auf Seite 5
- „EXPORT - Befehl“ auf Seite 9
- „EXPORT - API“ auf Seite 12
- „Datenstruktur SQLUEXPT-OUT“ auf Seite 18
- „Dateitypwerte (EXPORT)“ auf Seite 19
- „Beispiele für EXPORT-Sitzungen“ auf Seite 22
- „Einschränkungen“ auf Seite 27
- „Fehlerbehebung“ auf Seite 27.

Weitere Informationen zum Exportieren von DB2 Data Links Manager-Daten finden Sie in „Versetzen von DB2 Data Links Manager-Daten mit EXPORT“ auf Seite 201. Weitere Informationen zum Exportieren von Daten aus typisierten Tabellen finden Sie in „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 219. Weitere Informationen zum Exportieren von Daten aus einer DRDA-Server-Datenbank in eine Datei auf der DB2 Connect-Workstation und umgekehrt finden Sie in „Versetzen von Daten mit DB2 Connect“ auf Seite 217.

### EXPORT - Übersicht

Das Dienstprogramm EXPORT exportiert Daten aus einer Datenbank in eine Betriebssystemdatei, die eines von mehreren externen Dateiformaten haben kann.

Folgende Informationen werden benötigt, wenn Daten exportiert werden:

- Eine SQL-Anweisung SELECT, welche die zu exportierenden Daten angibt.
- Der Pfad und Name der Betriebssystemdatei, in der die exportierten Daten gespeichert werden.
- Das Format der Daten in der Eingabedatei. Mögliche Formate sind: IXF, WSF oder DEL. Siehe „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.
- Ein Nachrichtendateiname.
- Beim Exportieren von typisierten Tabellen müssen Sie möglicherweise die Traversierfolge für die untergeordneten Tabellen angeben. Bei Verwendung des IXF-Formats wird empfohlen, die Standardreihenfolge zu verwenden. Beachten Sie beim Angeben der Reihenfolge, daß die untergeordneten Tabellen der vorgegebenen Anordnung (PRE-ORDER) traversiert werden müssen. Beim Exportieren von typisierten Tabellen können Sie eine SELECT-Anweisung nicht direkt angeben. Stattdessen müssen Sie den Namen der untergeordneten Zieltabelle und (wahlfrei) eine WHERE-Klausel angeben. Das Dienstprogramm EXPORT verwendet diese Informationen zusammen mit der Traversierfolge zum Generieren und Ausführen der benötigten SELECT-Anweisung. Weitere Informationen finden Sie in „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 219.

Sie können auch folgendes angeben:

- Neue Spaltennamen (beim Exportieren in IXF- oder WSF-Dateien). Wenn Sie keine neuen Spaltennamen angeben möchten, werden die Spaltennamen in der vorhandenen Tabelle oder Sicht in der exportierten Datei verwendet.
- Weitere Optionen zur Anpassung der Exportoperation (siehe „Dateityp-werte (EXPORT)“ auf Seite 19).

Wenn Sie das Dienstprogramm EXPORT in einer Umgebung mit mehreren Datenbankpartitionen verwenden möchten, können Sie **db2batch** verwenden, um die Task in jeder Datenbankpartition auszuführen. Die SELECT-Anweisung muß in der Lage sein, nur die lokal gefundenen Daten zurückzugeben. Die Auswahlbedingung sieht folgendermaßen aus:

```
SELECT * FROM tabellenname WHERE NODENUMBER(spaltenname) = CURRENT NODE
```

Weitere Informationen zu **db2batch** finden Sie im Handbuch *Command Reference* oder *Systemverwaltung*.

---

## Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von EXPORT

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie Operationen von Dienstprogrammen und zur Datenbankmanagerpflege auf höherer Ebene zu gruppieren. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie die entsprechende Berechtigung besitzen, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung SYSADM oder DBADM bzw. das Zugriffsrecht CONTROL oder SELECT für jede Tabelle, die an dem Exportvorgang beteiligt ist.

---

## Verwenden von EXPORT

### Vor der Verwendung von EXPORT

Bevor Sie das Dienstprogramm EXPORT ausführen, müssen Sie mit der Datenbank, aus der die Daten exportiert werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm die Anweisung COMMIT absetzt, sollten Sie vor dem Aufrufen von EXPORT alle Transaktionen beenden und alle Sperren aufheben, indem Sie die Anweisung COMMIT oder ROLLBACK ausführen. Andere Anwendungen, die in separaten Verbindungen auf die Tabelle zugreifen, brauchen die Verbindung nicht zu trennen.

### Aufrufen von EXPORT

Das Dienstprogramm EXPORT kann wie folgt aufgerufen werden:

- Über den Befehlszeilenprozessor (CLP).  
Es folgt ein Beispiel für den Befehl EXPORT, der über den CLP abgesetzt wird:  

```
db2 export to staff.ixf of ixf select * from userid.staff
```
- Über das Notizbuch **Exportieren** in der Steuerzentrale. Gehen Sie wie folgt vor, um das Notizbuch **Exportieren** zu öffnen:
  1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie den Ordner **Tabellen** oder **Sichten** sehen.
  2. Klicken Sie den Ordner an, mit dem Sie arbeiten möchten. Alle vorhandenen Tabellen oder Sichten werden auf der rechten Seite des Fensters, dem Inhaltsteilfenster, angezeigt.

## Verwenden von EXPORT

3. Klicken Sie im Inhaltsteifenster die gewünschte Tabelle oder Sicht mit Maustaste 2 an, und wählen Sie **Exportieren** im Kontextmenü aus. Das Notizbuch **Exportieren** wird geöffnet.

Basisinformationen zur Steuerzentrale finden Sie im Handbuch *Systemverwaltung*. Zusatzinformationen bietet die Online-Hilfefunktion innerhalb der Steuerzentrale.

- Über die Anwendungsprogrammierschnittstelle (API) **sqluexpr**. Informationen zu dieser API finden Sie in „EXPORT - API“ auf Seite 12. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, finden Sie im Handbuch *Application Building Guide*.

---

## Verwenden von EXPORT mit Identitätsspalten

Mit dem Dienstprogramm EXPORT können Sie Daten aus einer Tabelle exportieren, die eine Identitätsspalte enthält. Hat die Anweisung SELECT, die für die Exportoperation angegeben wird, das Format "select \* from tabellenname" und wird die Option METHOD nicht verwendet, wird das Exportieren der Identitätsspaltenmerkmale in IXF-Dateien unterstützt. Anschließend können die Optionen REPLACE\_CREATE und CREATE des Befehls IMPORT verwendet werden, um die Tabelle einschließlich ihrer Identitätsspaltenmerkmale erneut zu erstellen. Wird eine solche IXF-Datei aus einer Tabelle mit einer als GENERATED ALWAYS definierten Identitätsspalte erstellt, kann die Daten-datei nur dann erfolgreich importiert werden, wenn der Wert identityignore angegeben wird. Andernfalls werden alle Zeilen zurückgewiesen (SQL3550W).

---

## Erneutes Erstellen einer exportierten Tabelle

Eine Tabelle kann durch Verwenden des Dienstprogramms EXPORT und Angabe des IXF-Dateiformats gesichert werden. Die gesicherte Tabelle (einschließlich der Indizes) kann dann mit Hilfe des Dienstprogramms IMPORT erneut erstellt werden (weitere Informationen, unter anderem auch eine Liste der Attribute, die das IXF-Dateiformat für eine Tabelle enthält, finden Sie unter „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 38).

Die Exportoperation schlägt fehl, wenn der verfügbare Speicherbereich auf dem Dateisystem, auf dem die exportierte Datei erstellt wird, für die zu exportierenden Daten nicht ausreicht. In diesem Fall müssen Sie die Menge der ausgewählten Daten durch Angeben von Bedingungen in der Klausel WHERE begrenzen, damit die Exportdatei auf dem Zieldateisystem Platz findet. Sie können das Dienstprogramm EXPORT mehrmals aufrufen, um alle Daten zu exportieren.



Die DEL- und ASC-Dateiformate enthalten zwar die Satzdaten, jedoch keine Beschreibungen der Zieltabelle. Um eine Tabelle mit Daten erneut zu erstellen, die in diesen Dateiformaten gespeichert wurden, erstellen Sie die Zieltabelle und verwenden anschließend das Dienstprogramm LOAD, IMPORT oder AutoLoader, um die Tabelle aus diesen Dateien zu füllen. Mit **db2look** (DB2 Statistics and DDL Extraction Tool, siehe Handbuch *Command Reference*) können Sie die ursprünglichen Tabellendefinitionen erfassen und die entsprechende Datendefinitionssprache generieren.

---

### Exportieren von großen Objekten (LOBs)

Beim Exportieren von Daten aus LOB-Spalten besteht die Standardaktion darin, die ersten 32 KB Daten auszuwählen und diese in die gleiche Datei wie die restlichen Spaltendaten zu stellen. Wenn der Änderungswert `lobsinfile` angegeben wurde (siehe „Dateitypwerte (EXPORT)“ auf Seite 19), wählt das Dienstprogramm EXPORT das gesamte LOB (bis zu 2 GB) aus und stellt es in eine separate Datei.

**Anmerkung:** Das IXF-Dateiformat speichert die LOB-Optionen der Spalte (beispielsweise über die Protokollierung der LOB-Spalte) nicht. Dies bedeutet, daß das Dienstprogramm IMPORT eine Tabelle mit einer LOB-Spalte, deren Größe mit 1 GB oder mehr definiert ist, nicht erneut erstellen kann.

---

### Paralleles Exportieren von Daten

Durch das parallele Exportieren von Daten wird die Datenübertragung reduziert, und das Schreiben der Ergebnismenge wird - ebenso wie die Generierung der formatierten Ausgabe - in einer effizienteren Weise auf die Knoten verteilt, als dies bei einer Exportoperation ohne Parallelverarbeitung der Fall wäre. Beim parallelen Exportieren von Daten (durch das Aufrufen mehrerer Exportoperationen, nämlich einer Operation für jede Partition einer Tabelle) werden die Daten extrahiert, auf den lokalen Knoten umgesetzt und anschließend in das lokale Dateisystem geschrieben. Im Gegensatz hierzu werden die Daten beim seriellen Exportieren (also beim Exportieren durch eine einzige Exportoperation) parallel extrahiert und anschließend an den Client geliefert, auf dem ein Einzelprozeß die Umsetzung vornimmt und die Ergebnismenge in ein lokales Dateisystem schreibt.

## Paralleles Exportieren von Daten

Mit dem Befehl **db2batch** können die Leistungskennzahlen und die Ausführungsdauer von SQL-Anweisungen überwacht werden. Dieses Dienstprogramm enthält außerdem eine Funktion für das parallele Exportieren in Umgebungen mit partitionierten Datenbanken, das folgendes ausführt:

- Es werden Abfragen ausgeführt, um die zu exportierenden Daten zu definieren.
- Auf jeder Partition wird eine Datei mit den exportierten Daten, die sich auf dieser Partition befinden, erstellt.

Abhängig davon, welche Tabellen abgefragt werden und wo sich diese Tabellen in der Umgebung mit partitionierten Datenbanken befinden, stehen für das parallele Exportieren zwei Methoden zur Verfügung:

- Das parallele Exportieren aus einer Einzelpartitionstabelle bzw. eine Verknüpfung oder Unterabfrage von mehreren durch Kollokation zusammengefaßten Tabellen (geben Sie hierzu die Option `-p s` für den Befehl **db2batch** an).

Es gibt zwei Situationen, in denen Tabellen als durch Kollokation zusammengefaßt betrachtet werden können:

- Die Tabellen befinden sich in einer Knotengruppe mit einer Einzelpartition, die für dieselbe Partition definiert ist.
- Die Tabellen gehören zu derselben Knotengruppe und enthalten Partitionierungsschlüssel mit identischer Spaltenanzahl und identischem Spaltentyp. Die entsprechenden Spalten des Partitionierungsschlüssels sind partitionskompatibel, und die Tabellen werden hinsichtlich des gesamten Partitionierungsschlüssels oder einer Obermenge des Partitionierungsschlüssels mit Gleichheitsattribut (`equijoin`) verknüpft.

In beiden Fällen kann die Abfrage für jede Partition ausgeführt werden, um die Exportdatendatei dieser Partition zu generieren. Zu diesem Zweck wird die Funktion `NODENUMBER` verwendet, die im folgenden beschrieben wird. (Bitte beachten Sie, daß das parallele Exportieren zurückgewiesen wird, wenn eine Tabelle nur in einer einzigen Partition vorhanden ist, da die Daten nur aus einer Partition abgerufen werden. Informationen dazu, wie Sie in einem solchen Fall das parallele Exportieren aktivieren können, finden Sie unter dem nächsten Listenpunkt.)

- Eine Anweisung `SELECT`, die für mehrere nicht durch Kollokation zusammengefaßte Tabellen ausgeführt wird. (Geben Sie im Befehl **db2batch** die Option `-p t tabellenname` oder die Option `-p d` an. Mit der ersten Option können Sie angeben, daß eine vorhandene Tabelle als Zwischenspeichertabelle verwendet werden soll. Die zweite Option bewirkt, daß das Dienstprogramm `EXPORT` eine Zwischenspeichertabelle erstellt.)

Das Dienstprogramm EXPORT verwendet eine Zwischenspeichertabelle, die durch die Exportabfrage gefüllt wird. Diese Zwischenspeichertabelle wird verwendet, um die Zeilen der „Exportergebnismenge“ zu lokalisieren, indem eine Anweisung INSERT für die Gesamtauswahlabfrage in der Zwischenspeichertabelle ausgeführt wird. Nachdem die Zwischenspeichertabelle erstellt wurde, generiert das Dienstprogramm EXPORT auf jeder Partition eine Exportdatendatei. Zu diesem Zweck wird die Anweisung

```
"select * WHERE NODENUMBER(spaltenname) = CURRENT NODE"
```

für die Zwischenspeichertabelle ausgeführt.

Eine Zwischenspeichertabelle kann außerdem zum parallelen Exportieren einer Einzelpartitionstabelle verwendet werden. In den meisten Fällen ist das Übertragen der Daten aus einer Zwischenspeichertabelle mit Einzelpartition in eine Zwischenspeichertabelle mit mehreren Partitionen und das anschließende parallele Exportieren der Zwischenspeichertabelle in alle Partitionen schneller als das serielle Exportieren der Einzelpartitionstabelle.

Das Dienstprogramm EXPORT führt parallel für jede Partition eine Abfrage aus, um die Daten auf dieser Partition abzurufen. Bei Verwendung des Befehls `db2batch -p s` wird die ursprüngliche SELECT-Abfrage parallel ausgeführt. Wenn der Befehl `db2batch -p t` bzw. der Befehl `db2batch -p d` verwendet wird, wird eine Zwischenspeichertabelle unter Verwendung der angegebenen Abfrage mit den Exportdaten geladen, und eine Abfrage `SELECT *` wird für die Zwischenspeichertabelle parallel auf jeder Partition ausgeführt, um die Daten zu exportieren. Um lediglich die Daten zu exportieren, die sich auf einer bestimmten Partition befinden, fügt der Befehl **db2batch** das Prädikat `NODENUMBER(spaltenname) = CURRENT NODE` zur Klausel `WHERE` der Abfrage hinzu, die für diese Partition ausgeführt wird. Für den Parameter *spaltenname* muß der qualifizierte Name oder der Namen ohne Qualifikationsmerkmal einer Tabellenspalte angegeben werden. Das Dienstprogramm EXPORT verwendet den ersten Spaltennamen in der ursprünglichen Abfrage, um diesen Parameter festzulegen.

## Paralleles Exportieren von Daten

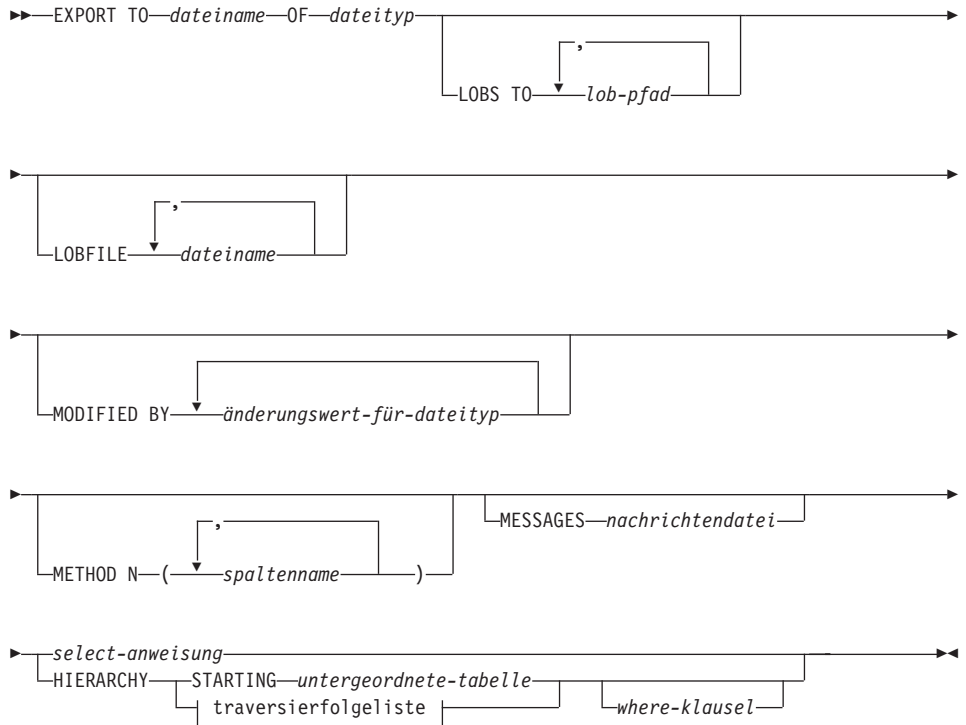
Für Abfragen, die durch das Dienstprogramm EXPORT verwendet werden, gelten die folgenden Einschränkungen:

- Bei Angabe von `db2batch -p s` darf die Abfrage nicht nur Spaltenfunktionen enthalten, da für das Prädikat `NODENUMBER spaltenname` ein Spaltenname benötigt wird.
- Bei Angabe von `db2batch -p s` müssen die Ergebnistabellen (wie z. B. MIN, MAX und AVG) auf einer Gruppierung basieren, die den Partitionierungsschlüssel enthält.
- Bei Angabe von `db2batch -p t` oder `db2batch -p d` darf die Abfrage keine Klausel ORDER BY enthalten, da ORDER BY für eine Gesamtauswahl innerhalb einer Anweisung INSERT durch DB2 UDB nicht unterstützt wird.

Wenn Sie bei Angabe von `-p s` im Befehl **db2batch** zusätzlich die Option `-r` verwenden, um Ergebnisausgabedateien zu erstellen, werden die Dateien auf jeder Partition sortiert, sofern Sie eine Klausel ORDER BY angegeben haben. Soll eine einzige sortierte Datei erstellt werden, fügen Sie die sortierten Dateien aus allen Partitionen in einer sortierten Datei zusammen. Verwenden Sie beispielsweise auf Systemen mit UNIX-Basis den Befehl `sort -m`, um die Dateien in einer einzigen sortierten Datei zusammenzufügen. Falls Sie Ihre Ausgabe an ein angehängtes NFS-Dateisystem senden, wird die Ausgabe nicht sortiert, und zwar selbst dann nicht, wenn Sie die Klausel ORDER BY angeben.

EXPORT - Befehl

Befehlssyntax



traversierfolgeliste:



Befehlsparameter

**HIERARCHY traversierfolgeliste**

Exportiert eine untergeordnete Hierarchie mit der angegebenen Traversierfolge. Alle untergeordneten Tabellen müssen in PRE-ORDER-Weise aufgelistet sein. Der Namen der ersten untergeordneten Tabelle wird als Zieltabellenname für die SELECT-Anweisung verwendet.

**HIERARCHY STARTING untergeordnete-tabelle**

Exportiert eine untergeordnete Hierarchie unter Verwendung der hierarchischen Standardreihenfolge (OUTER-Reihenfolge für ASC-,

## EXPORT - Befehl

DEL- oder WSF-Dateien oder die in PC/IXF-Datendateien gespeicherte Reihenfolge), beginnend mit *untergeordnete-tabelle*.

### **LOBFILE dateiname**

Gibt einen oder mehrere Basisdateinamen für LOB-Dateien an. Wenn der Namensbereich für den ersten Namen erschöpft ist, wird der zweite Name verwendet, und so weiter.

Bei der Erstellung von LOB-Dateien während einer Exportoperation werden die Dateinamen erzeugt, indem der aktuelle Basisdateiname aus dieser Liste an den aktuellen Pfad (aus *lob-pfad*) und daran wiederum eine dreistellige Folgennummer angefügt wird. Beispiel: Wenn der aktuelle LOB-Pfad das Verzeichnis `/u/foo/lob/pfad` und der aktuelle LOB-Dateiname `bar` ist, heißen die erstellten LOB-Dateien `/u/foo/lob/pfad/bar.001`, `/u/foo/lob/pfad/bar.002` und so weiter.

### **LOBS TO lob-pfad**

Gibt einen oder mehrere Pfade zu Verzeichnissen an, in denen die LOB-Dateien gespeichert werden sollen. Wenn der Dateibereich im ersten Pfad verbraucht ist, wird der zweite Pfad verwendet und so weiter.

### **MESSAGES nachrichtendatei**

Gibt das Ziel für Warnungen und Fehlernachrichten an, die bei einer Exportoperation auftreten. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm EXPORT an sie angefügt. Wenn *nachrichtendatei* weggelassen wird, werden die Nachrichten in die Standardausgabe geschrieben.

### **METHOD N spaltenname**

Gibt einen oder mehrere Spaltennamen an, die in der Ausgabedatei verwendet werden sollen. Wenn dieser Parameter nicht angegeben wird, werden die Spaltennamen in der Tabelle verwendet. Dieser Parameter ist nur für WSF- und IXF-Dateien gültig. Er ist nicht gültig beim Exportieren von hierarchischen Daten.

### **MODIFIED BY änderungswert-für-dateityp**

Gibt zusätzliche Optionen an (siehe Tabelle 2 auf Seite 19).

### **OF dateityp**

Gibt das Format der Daten in der Ausgabedatei an:

- DEL (ASCII-Format ohne universelle Zeilenbegrenzer), das von einer Vielzahl von Datenbankmanager- und Dateimanagerprogrammen verwendet wird.
- WSF (Arbeitsblattformat), das von Programmen wie den folgenden verwendet wird:
  - Lotus 1-2-3
  - Lotus Symphony

**Anmerkung:** Beim Exportieren von BIGINT- oder DECIMAL-Daten können nur Werte, die innerhalb des Bereichs des Typs DOUBLE liegen, korrekt exportiert werden. Es werden zwar auch Werte exportiert, die nicht in diesem Bereich liegen, beim Zurückimportieren oder -laden dieser Werte können jedoch je nach Betriebssystem falsche Daten entstehen.

- IXF (Integrated Exchange Format, PC-Version). In diesem Format werden die meisten Tabellenattribute sowie vorhandene Indizes in der IXF-Datei gesichert, sofern keine Spalten in der SELECT-Anweisung angegeben werden. Bei diesem Format kann die Tabelle erneut erstellt werden, während sie bei den anderen Formaten bereits vorhanden sein muß, damit Daten in sie importiert werden können. Eine Liste der Attribute, die das IXF-Dateiformat zu einer Tabelle enthält, finden Sie unter „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 38.

Weitere Informationen zu Dateiformaten finden Sie in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

#### **select-anweisung**

Gibt die SELECT-Anweisung an, welche die zu exportierenden Daten zurückgibt. Wenn die SELECT-Anweisung zu einem Fehler führt, wird eine Nachricht in die Nachrichtendatei (oder in die Standardausgabe) geschrieben. Bei dem Fehlercode SQL0012W, SQL0347W, SQL0360W, SQL0437W oder SQL1824W wird die Exportoperation fortgesetzt, andernfalls wird sie gestoppt.

#### **TO dateiname**

Gibt den Namen der Datei an, in welche die Daten exportiert werden sollen. Wenn der vollständige Pfad zu der Datei nicht angegeben wird, verwendet das Dienstprogramm EXPORT das aktuelle Verzeichnis und das Standardlaufwerk als Ziel.

Wenn der Name einer Datei, die bereits vorhanden ist, angegeben wird, überschreibt das Dienstprogramm EXPORT den Inhalt der Datei. Die Informationen werden nicht angefügt.

### C-API-Syntax

```
/* File: sqlutil.h */
/* API: Export */
/* ... */
SQL_API_RC SQL_API_FN
sqluexpr (
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    sqlu_media_list * pLobFileList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluexpt_out* pOutputInfo,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

### Generische API-Syntax

```
/* File: sqlutil.h */
/* API: Export */
/* ... */
SQL_API_RC SQL_API_FN
sqlgexpr (
    unsigned short DataFileNameLen,
    unsigned short FileTypeLen,
    unsigned short MsgFileNameLen,
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    sqlu_media_list * pLobFileList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluexpt_out* pOutputInfo,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

### API-Parameter

#### DataFileNameLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der Datendatei in Byte darstellt.



**FileTypeLen**

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Dateityps in Byte darstellt.

**MsgFileNameLen**

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der Nachrichtendatei in Byte darstellt.

**pDataFileName**

Eingabe. Eine Zeichenfolge, die den Pfad und den Namen einer externen Datei enthält, in die die Daten exportiert werden sollen.

**pLobPathList**

Eingabe. Eine *sqlu\_media\_list* unter Verwendung von *media\_type* `SQLU_LOCAL_MEDIA` und der Struktur *sqlu\_media\_entry*, die Pfade auf dem Client angibt, in denen die LOB-Dateien gespeichert werden sollen.

Wenn der Dateibereich im ersten Pfad in dieser Liste erschöpft ist, wird von der API der zweite Pfad verwendet, und so weiter.

Weitere Informationen finden Sie unter „SQLU-MEDIA-LIST“ im Handbuch *Administrative API Reference*.

**pLobFileList**

Eingabe. Eine *sqlu\_media\_list* unter Verwendung von *media\_type* `SQLU_CLIENT_LOCATION` und der Struktur *sqlu\_location\_entry*, die Basisdateinamen enthält.

Wenn der Namensbereich bei Verwendung des ersten Namens in dieser Liste erschöpft ist, wird von der API der zweite Name verwendet, und so weiter.

Weitere Informationen finden Sie unter „SQLU-MEDIA-LIST“ im Handbuch *Administrative API Reference*.

Bei der Erstellung von LOB-Dateien während einer Exportoperation werden die Dateinamen erzeugt, indem der aktuelle Basisdateiname aus dieser Liste an den aktuellen Pfad (aus *pLobFilePath*) und daran wiederum eine dreistellige Folgenummer angefügt wird. Beispiel: Wenn der aktuelle LOB-Pfad das Verzeichnis `/u/foo/lob/pfad` und der aktuelle LOB-Dateiname `bar` ist, heißen die erstellten LOB-Dateien `/u/foo/lob/pfad/bar.001`, `/u/foo/lob/pfad/bar.002` usw.

**pDataDescriptor**

Eingabe. Zeiger auf eine *sqldcol*-Struktur, die die Spaltennamen für die Ausgabedatei angibt. Der Wert des Feldes *dcolmeth* bestimmt, wie die restlichen in diesem Parameter bereitgestellten Informationen vom Dienstprogramm EXPORT ausgewertet werden. Gültige Werte für diesen Parameter (definiert in `sqlutil`) sind:

### SQL\_METH\_N

Namen. Gibt Spaltennamen an, die in der Ausgabedatei verwendet werden sollen.

### SQL\_METH\_D

Standardeinstellung. Vorhandene Spaltennamen aus der Tabelle sollen in der Ausgabedatei verwendet werden. In diesem Fall werden die Anzahl der Spalten und die Spaltenspezifikationsmatrix ignoriert. Die Spaltennamen werden aus der Ausgabe der Anweisung SELECT abgeleitet, die in *pActionString* angegeben ist.

Weitere Informationen finden Sie unter „SQLDCOL“ im Handbuch *Administrative API Reference*.

### **pActionString**

Eingabe. Zeiger auf eine *sqlchar*-Struktur, die eine gültige dynamische SQL-Anweisung SELECT enthält. Die Struktur enthält ein 2-Byte-Langfeld, gefolgt von den Zeichen, die die SELECT-Anweisung bilden. Die SELECT-Anweisung gibt die Daten an, die aus der Datenbank extrahiert und in die externe Datei geschrieben werden sollen.

Die Spalten für die externe Datei (aus *pDataDescriptor*) und die Datenbankspalten aus der SELECT-Anweisung werden gemäß ihrer jeweiligen Listen-/Strukturpositionen abgeglichen. Die erste aus der Datenbank gewählte Spalte mit Daten wird in die erste Spalte der externen Datei gestellt, und ihr Spaltenname wird dem ersten Element der externen Spaltenmatrix entnommen.

Weitere Informationen finden Sie unter „SQLCHAR“ im Handbuch *Administrative API Reference*.

**Anmerkung:** Eine Beschreibung der für typisierte Tabellen zu verwendenden Syntax finden Sie in „EXPORT - Befehl“ auf Seite 9.

### **pFileType**

Eingabe. Eine Zeichenfolge, die das Format der Daten in der externen Datei angibt. Unterstützte externe Dateiformate (definiert in *sqlutil*) sind:

### SQL\_DEL

ASCII-Format ohne universelle Zeilenbegrenzer für den Datenaustausch mit dBase, BASIC und den IBM Personal Decision Series-Programmen und vielen anderen Datenbank- und Dateimanagern.

**SQL\_WSF**

Arbeitsblattformate zum Datenaustausch mit den Programmen Lotus Symphony und 1-2-3.

**SQL\_IXF**

PC-Version des Integrated Exchange Format. Dies ist das bevorzugte Format für den Export von Daten aus einer Tabelle. In diesem Format exportierte Daten können später in dieselbe Tabelle oder in eine andere Datenbankmanagertabelle importiert oder geladen werden.

**pFileTypeMod**

Eingabe. Ein Zeiger auf eine *sqliacol*-Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbeitungsoptionen angeben. Wenn dieser Zeiger NULL ist oder die Struktur, auf die gezeigt wird, null Zeichen enthält, wird diese Aktion als Auswahl einer Standardspezifikation gewertet.

Nicht alle Optionen können mit allen unterstützten Dateitypen verwendet werden.

Weitere Informationen finden Sie unter „SQLCHAR“ im Handbuch *Administrative API Reference* und unter „Dateitypwerte (EXPORT)“ auf Seite 19.

**pMsgFileName**

Eingabe. Eine Zeichenfolge, die den Bestimmungsort für vom Dienstprogramm zurückgegebene Fehlermeldungen, informative Nachrichten und Warnungen enthält. Dies kann der Pfad und der Name einer Betriebssystemdatei oder einer Standardeinheit sein. Wenn die Datei bereits vorhanden ist, wird sie überschrieben. Wenn sie nicht vorhanden ist, wird eine neue Datei erstellt.

**CallerAction**

Eingabe. Eine vom Aufrufenden angeforderte Aktion. Gültige Werte (definiert in *sqlutil*) sind:

**SQLU\_INITIAL**

Erstaufruf. Dieser Wert muß beim ersten Aufruf der API verwendet werden.

Wenn der Erstaufruf oder ein nachfolgender Aufruf zurückgegeben wird und erfordert, daß die aufrufende Anwendung eine Aktion ausführt, bevor die angeforderte Exportoperation ausgeführt wird, muß die Aufruferaktion auf eine der folgenden Optionen gesetzt werden:

**SQLU\_CONTINUE**

Verarbeitung fortsetzen. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Ers-

taufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, daß die vom Dienstprogramm angeforderte Benutzeraktion beendet wurde und das Dienstprogramm die Verarbeitung der Erstanforderung fortsetzen kann.

### SQLU\_TERMINATE

Verarbeitung beenden. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebedingung zu reagieren). Er gibt an, daß die vom Dienstprogramm angeforderte Benutzeraktion nicht ausgeführt wurde und das Dienstprogramm die Verarbeitung der Erstanforderung beenden soll.

### pOutputInfo

Ausgabe. Gibt die Anzahl der in die Zielfile exportierten Datensätze zurück. Weitere Informationen zu dieser Struktur finden Sie in „Datenstruktur SQLUEXPT-OUT“ auf Seite 18.

### pReserved

Zur zukünftigen Verwendung reserviert.

### pSqlca

Ausgabe. Ein Zeiger auf die Struktur *sqlca*. Weitere Informationen zu dieser Struktur finden Sie im Abschnitt über SQLCA des Handbuchs *Administrative API Reference*.

## REXX-API-Syntax

```
EXPORT :stmt TO datafile OF filetype  
[MODIFIED BY :filemod] [USING :dcoldata]  
MESSAGES msgfile [ROWS EXPORTED :number]
```

```
CONTINUE EXPORT
```

```
STOP EXPORT
```

## REXX-API-Parameter

**stmt** Eine REXX-Host-Variable, die eine gültige dynamische SQL-Anweisung SELECT enthält. Die Anweisung gibt die Daten an, die aus der Datenbank extrahiert werden sollen.

### datafile

Name der Datei, in die die Daten exportiert werden sollen.

**filetype**

Das Format der Daten in der Exportdatei. Folgende Dateiformate werden unterstützt:

**DEL** ASCII-Format ohne universelle Zeilenbegrenzer

**WSF** Arbeitsblattformat

**IXF** PC-Version des Integrated Exchange Format

**filetmod**

Eine Host-Variable, die zusätzliche Verarbeitungsoptionen enthält (siehe „Dateitypwerte (EXPORT)” auf Seite 19).

**dcoldata**

Eine REXX-Verbund-Host-Variable, die die Spaltennamen enthält, die in der Exportdatei verwendet werden sollen. Im folgenden steht XXX für den Namen der Host-Variablen:

**XXX.0** Anzahl der Spalten (Anzahl der Elemente im restlichen Teil der Variablen).

**XXX.1** Erster Spaltenname.

**XXX.2** Zweiter Spaltenname.

**XXX.3** und so weiter.

Wenn dieser Parameter NULL ist oder kein Wert für *dcoldata* angegeben wurde, verwendet das Dienstprogramm die Spaltennamen aus der Datenbanktabelle.

**nachrichtendatei**

Datei, Pfad oder Einheitenname für Fehlernachrichten und Warnungen.

**anzahl**

Eine Host-Variable, die die Anzahl der exportierten Zeilen enthält.

## Datenstruktur SQLUEXPT-OUT

---

### Datenstruktur SQLUEXPT-OUT

Diese Struktur wird verwendet, um Informationen von der EXPORT-API zu übergeben (siehe „EXPORT - API“ auf Seite 12).

*Tabelle 1. Felder in der Struktur SQLUEXPT-OUT*

Feldname	Datentyp	Beschreibung
SIZEOFSTRUCT	INTEGER	Größe der Struktur.
ROWSEXPORTED	INTEGER	Anzahl der aus der Datenbank in die Zielfeld exportierten Datensätze.

## Sprachsyntax

### C-Struktur

```
/* File: sqlutil.h */
/* Structure: SQL-UExPT-OUT */
/* ... */
SQL_STRUCTURE sqluexpt_out
{
    sqluint32      sizeofStruct;
    sqluint32      rowsExported;
};
/* ... */
```

### COBOL-Struktur

```
* File: sqlutil.cbl
01 SQL-UExPT-OUT.
   05 SQL-SIZE-OF-UExPT-OUT PIC 9(9) COMP-5 VALUE 8.
   05 SQL-ROWSEXPORTED PIC 9(9) COMP-5 VALUE 0.
*
```

## Dateitypwerte (EXPORT)

Tabelle 2. Gültige Dateitypwerte (EXPORT)

Wert	Beschreibung
<b>Alle Dateiformate</b>	
lobsinfile	<i>lob-pfad</i> gibt den Pfad zu den Dateien an, die LOB-Werte enthalten.
<b>DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)</b>	
chardelx	<p>x ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird statt doppelter Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.<sup>a</sup></p> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden:</p> <pre>modified by chardel''</pre>
coldelx	<p>x ist ein einzelner Zeichenspaltenbegrenzer. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird statt eines Kommas verwendet, um das Ende einer Spalte anzugeben.<sup>a</sup></p> <p>Im folgenden Beispiel bewirkt coldel;, daß das Dienstprogramm EXPORT jedes Semikolon (;) als Spaltenbegrenzungszeichen wertet:</p> <pre>db2 "export to temp of del modified by coldel; select * from staff where dept = 20"</pre>
datesiso	Datumsformat. Bewirkt, daß alle Datumswerte im ISO-Format exportiert werden ( <i>JJJJ-MM-TT</i> ). <sup>b</sup>
decplusblank	Zeichen für Pluszeichen. Bewirkt, daß positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, Dezimalwerten ein Pluszeichen voranzustellen.
decptx	x ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird statt eines Punktes als Dezimalzeichen verwendet. <sup>a</sup>
dldelx	<p>x ist ein DATALINK-Begrenzer (Einzelzeichen). Der Standardwert ist ein Semikolon (;). Das angegebene Zeichen wird anstelle eines Semikolons als Trennzeichen zwischen Feldern für einen DATALINK-Wert verwendet. Es ist erforderlich, weil ein DATALINK-Wert mehrere Teilwerte enthalten kann. <sup>a</sup></p> <p><b>Anmerkung:</b> x darf nicht mit dem Zeichen übereinstimmen, das als Zeilen-, Spalten- oder Zeichenfolgebegrenzer angegeben wurde.</p>

## Dateitypwerte (EXPORT)

Tabelle 2. Gültige Dateitypwerte (EXPORT) (Forts.)

Wert	Beschreibung
nodoubledel	Unterdrückt die Erkennung von doppelten Zeichenbegrenzern. Weitere Informationen finden Sie unter „Einschränkungen für Begrenzer“.
<b>WSF-Dateiformat</b>	
1	Erstellt eine WSF-Datei, die mit Lotus 1-2-3, Release 1, oder Lotus 1-2-3, Release 1a, kompatibel ist. <sup>c</sup> Dies ist der Standardwert.
2	Erstellt eine WSF-Datei, die mit Lotus Symphony, Release 1.0, kompatibel ist. <sup>c</sup>
3	Erstellt eine WSF-Datei, die mit Lotus 1-2-3, Version 2, oder Lotus Symphony, Release 1.1, kompatibel ist. <sup>c</sup>
4	Erstellt eine WSF-Datei, die DBCS-Zeichen enthält.
<b>Anmerkungen:</b>	
1. Das Dienstprogramm EXPORT setzt keine Warnung ab, wenn versucht wird, nicht unterstützte Dateitypen mit der Option MODIFIED BY zu verwenden. Wenn dies versucht wird, schlägt die Exportoperation fehl, und es wird ein Fehlercode zurückgegeben.	
2. <sup>a</sup> „Einschränkungen für Begrenzer“ listet Einschränkungen auf, die für zur Überschreibung von Begrenzern verwendbare Zeichen gelten.	
3. <sup>b</sup> Normalerweise schreibt das Dienstprogramm EXPORT	
• Datumsdaten im Format <i>JJJJMMTT</i>	
• Daten des Typs char(date) im Format <i>“JJJJ-MM-TT”</i>	
• Zeitdaten im Format <i>“HH.MM.SS”</i>	
• Zeitmarkendaten im Format <i>“JJJJ-MM-TT-HH.MM.SS.uuuuuu”</i>	
Daten in allen Spalten des Typs DATETIME, die in der Anweisung SELECT für die Exportoperation angegeben sind, stehen ebenfalls in diesen Formaten.	
4. <sup>c</sup> Diese Dateien können auch für ein bestimmtes Produkt erstellt werden, indem L für Lotus 1-2-3 oder S für Symphony in der Parameterzeichenfolge <i>dateitypwert</i> angegeben wird. Es kann nur ein Wert bzw. Produktbezeichner angegeben werden.	

### Einschränkungen für Begrenzer

Der Benutzer muß darauf achten, daß das gewählte Begrenzungszeichen nicht Teil der zu versetzenden Daten ist. Andernfalls können unvorhersehbare Fehler auftreten. Beim Versetzen von Daten gelten für Spalten-, Zeichenfolgen-, DATALINK- und Dezimalzeichenbegrenzern die folgenden Einschränkungen:

- Begrenzer schließen sich gegenseitig aus.



- Unzulässige Begrenzer sind binäre Nullen, Zeilenvorschubzeichen, Rücklauf- oder Leerzeichen.
- Das Standarddezimalzeichen (.) kann nicht als Zeichenfolgebegrenzer verwendet werden.
- Die folgenden Zeichen werden durch eine ASCII- und EBCDIC-Codepage anders angegeben:
  - Das DBCS-Endezeichen (0x0F) und das DBCS-Startzeichen (0x0E) können nicht als Begrenzer für eine EBCDIC-MBCS-Datendatei verwendet werden.
  - Begrenzer für MBCS-, EUC- oder DBCS-Codepages dürfen nicht größer sein als 0x40. Hiervon ausgenommen ist das Standarddezimalzeichen für EBCDIC-MBCS-Daten (0x4b).
  - Standardbegrenzer für Datendateien in ASCII-Codepages oder EBCDIC-MBCS-Codepages sind:
    - " (0x22, doppeltes Anführungszeichen; Zeichenfolgebegrenzer)
    - , (0x2c, Komma; Spaltenbegrenzer)
  - Standardbegrenzer für Datendateien in EBCDIC-SBCS-Codepages sind:
    - " (0x7F, doppeltes Anführungszeichen; Zeichenfolgebegrenzer)
    - , (0x6B, Komma; Spaltenbegrenzer)
  - Das Standarddezimalzeichen für ASCII-Datendateien ist 0x2e (Punkt).
  - Das Standarddezimalzeichen für EBCDIC-Datendateien ist 0x4B (Punkt).
  - Wenn sich die Codepage des Servers von der des Clients unterscheidet, sollte die hexadezimale Darstellung von Begrenzern, die von den Standardwerten abweichen, angegeben werden. Beispiel:

```
db2 load from ... modified by charde10x0C colde1X1e ...
```

Die folgenden Informationen zur Unterstützung für die Erkennung von doppelten Zeichenbegrenzern in DEL-Dateien gelten für die Dienstprogramme EXPORT, IMPORT und LOAD:

- Zeichenbegrenzer sind innerhalb der zeichenorientierten Felder einer DEL-Datei zulässig. Dies gilt für Felder des Typs CHAR, VARCHAR, LONG VARCHAR oder CLOB (außer, wenn lobsinfile angegeben ist). Jedes Paar von Zeichenbegrenzern, das sich zwischen den eigentlichen Zeichenbegrenzern befindet, wird in die Datenbank importiert oder geladen. Beispiel:

```
"What a ""nice"" day!"
```

wird importiert als:

```
What a "nice" day!
```

Bei einem Export gelten die Regeln umgekehrt. Beispiel:

```
I am 6" tall.
```

wird in eine DEL-Datei exportiert als:

## Dateitypwerte (EXPORT)

"I am 6" tall."

- In einer DBCS-Umgebung wird das Pipe-Zeichen (|) nicht als Zeichenbegrenzer unterstützt.

---

## Beispiele für EXPORT-Sitzungen

### CLP-Beispiele

Das folgende Beispiel zeigt, wie Daten aus der Tabelle STAFF der Beispieldatenbank SAMPLE (mit der der Benutzer verbunden sein muß) im IXF-Format in die Datei myfile.ixf exportiert werden. Wenn die Datenbankverbindung nicht über DB2 Connect hergestellt wurde, werden die Indexdefinitionen (falls vorhanden) in der Ausgabedatei gespeichert. Andernfalls werden nur die Daten gespeichert.

```
db2 export to myfile.ixf of ixf messages msgs.txt select * from staff
```

Das folgende Beispiel zeigt, wie Daten über Mitarbeiter in Abteilung 20 aus der Tabelle STAFF der Beispieldatenbank SAMPLE (mit der der Benutzer verbunden sein muß) im IXF-Format in die Datei awards.ixf exportiert werden.

```
db2 export to awards.ixf of ixf messages msgs.txt select * from staff
where dept = 20
```

Das folgende Beispiel zeigt, wie große Objekte in eine DEL-Datei exportiert werden:

```
db2 export to myfile.del of del lobs to mylobs
lobfile lobs1, lobs2 modified by lobsinfile
select * from emp_photo
```

Das folgende Beispiel zeigt, wie große Objekte in eine DEL-Datei exportiert werden, wobei ein zweites Verzeichnis für Dateien angegeben wird, die im ersten Verzeichnis möglicherweise keinen Platz haben:

```
db2 export to myfile.del of del
lobs to /db2exp1, /db2exp2 modified by lobsinfile
select * from emp_photo
```

Das folgende Beispiel zeigt den Export von Daten in eine DEL-Datei, wobei ein einfaches Anführungszeichen als Zeichenfolgebegrenzer, ein Semikolon als Spaltenbegrenzer und ein Komma als Dezimalzeichen verwendet wird. Die gleiche Konvention muß beim Rückimportieren der Daten in die Datenbank verwendet werden:

```
db2 export to myfile.del of del
modified by chardel'' coldel; decpt,
select * from staff
```

### API-Beispiele

Das folgende Beispielprogramm zeigt, wie folgende Aufgaben ausgeführt werden:

- Exportieren von Informationen aus der Tabelle STAFF der Beispieldatenbank SAMPLE in die Datei EXPTABLE.DEL
- Importieren dieser Informationen aus der Textdatei mit begrenzter Satzlänge in eine neue Tabelle, IMPTABLE

Zusatzinformationen zur Beispieldatenbank SAMPLE finden Sie im Handbuch *Systemverwaltung*.

Die Quellendatei für dieses Beispielprogramm (impexp.sqc) befindet sich im Verzeichnis \sql11b\samples\c. Sie enthält sowohl DB2-API-Aufrufe als auch Aufrufe mit eingebettetem SQL. Die Prozedurdatei bldvaemb.cmd, die sich im gleichen Verzeichnis befindet, enthält die Befehle zur Erstellung dieses und anderer Beispielprogramme. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, und Zusatzinformationen zu Kompilierungs- und Programmverbindungsoptionen finden Sie im Handbuch *Application Building Guide*. Gehen Sie wie folgt vor, um das Beispielprogramm impexp aus der Quellendatei impexp.sqc unter OS/2 zu erstellen:

1. Kopieren Sie die Dateien impexp.sqc, bldvaemb.cmd, util.c und util.h in ein Arbeitsverzeichnis.
2. Setzen Sie, falls der Datenbankmanager nicht aktiv ist, den Befehl db2start ab.
3. Geben Sie bldvaemb impexp sample ein. Die folgenden Dateien werden generiert:

```

impexp.bnd
impexp.c
util.obj
impexp.obj
impexp.exe

```

Geben Sie impexp ein, um das Beispielprogramm (ausführbare Datei) zu starten. Es kann unter Umständen nützlich sein, einige der generierten Dateien näher zu betrachten, etwa die Nachrichtendatei und die ASCII-Datendatei mit begrenzter Satzlänge.

```

/*****
**
** Source File Name = impexp.sqc 1.4
**
** PURPOSE :
**   This program is an example of how APIs are implemented in order to
**   export and import tables and table data. The order of the program
**   is as follows:

```

## Beispiele für EXPORT-Sitzungen

```
**          - export a table to a comma-delimited text file
**          - import the comma-delimited text file to a DB2 table
**          This program needs the embedded SQL calls in order to connect to
**          an existing database, then to create a temporary table to work with.
**
** STRUCTURES USED :
**      sqldcol
**      sqlchar
**      sqluexpt_out
**      sqluimp_in
**      sqluimp_out
**      sqlca
**
** APIs USED :
**      IMPORT TO          sqluimpt_api
**      EXPORT            sqlgexpt
**
** FUNCTIONS DECLARED :
**      'C' COMPILER LIBRARY :
**          stdio.h - printf
**          string.h - fgets, strncpy
**
**      DBMS LIBRARY :
**          sqlenv.h - see "APIs USED" above
**
**      OTHER :
**          internal :
**              list_dcs :          Displays a directory of databases
**
**          external :
**              check_error :      Checks for SQLCODE error, and prints out any
**              [in UTIL.C]        related information available.
**
**      EXTERNAL DEPENDANCIES :
**          - Ensure existence of database (SAMPLE) for precompile purposes.
**          - Precompile with the SQL precompiler (PREP in DB2)
**          - Bind to a database (BIND in DB2)
**          - Compile and link with the IBM Cset++ compiler (AIX and OS/2)
**            or the Microsoft Visual C++ compiler (Windows)
**            or the compiler supported on your platform.
**
**      *****/
**      #include <stdio.h>
**      #include <stdlib.h>
**      #include <string.h>
**      #include <sqlenv.h>
**      #include <sqlutil.h>
**      #ifndef DB2MAC
**      #include <malloc.h>
**      #endif
**      #include "util.h"
**
**      #ifdef DB268K
**      /* Need to include ASLM for 68K applications */
**      #include <LibraryManager.h>
```

```

#endif

#define CHECKERR(CE_STR)  if (check_error (CE_STR, &sqlca) != 0) return 1;

EXEC SQL INCLUDE SQLCA;
int main (int argc, char *argv[]) {
    short int          callerAction = 0;
    struct sqldcol      columnData;
    struct sqlchar      *columnStringPointer;
    struct sqluexpt_out outputInfo;
    struct sqluimpt_in  impInput;
    struct sqluimpt_out impOutput;

    char                datafile[] = "EXPTABLE.DEL";
    char                statement[] = "select name, id from staff";
    char                impStatement[] = "insert into imptable (name, id)";
    char                msgfile_x[] = "EXPMMSG.TXT";
    char                msgfile_m[] = "IMPMSG.TXT";
    char                fileFormat[] = "DEL";

    EXEC SQL BEGIN DECLARE SECTION;
        char userid[9];
        char passwd[19];
    EXEC SQL END DECLARE SECTION;

#ifdef DB268K
    /* Before making any API calls for 68K environment,
       need to initial the Library Manager */
    InitLibraryManager(0,kCurrentZone,kNormalMemory);
    atexit(CleanupLibraryManager);
#endif

    /* need to preset the size of structure field and counts */
    outputInfo.sizeOfStruct = SQLUEXPT_OUT_SIZE;
    impInput.sizeOfStruct = SQLUIMPT_IN_SIZE;
    impOutput.sizeOfStruct = SQLUIMPT_OUT_SIZE;
    impInput.restartcnt = impInput.commitcnt = 0;

    /*****
    * need to allocate the proper amount of space for the SQL statement *
    *****/
    columnStringPointer = (struct sqlchar *)malloc(strlen(statement)
        + sizeof (struct sqlchar));
    columnStringPointer->length = strlen(statement);
    strncpy (columnStringPointer->data, statement, strlen(statement));
    /* DELimited format can not have specified names, therefore the
       column method is 'D'efault */
    columnData.dcolmeth = 'D';

    if (argc == 1) {
        EXEC SQL CONNECT TO sample;
        CHECKERR ("CONNECT TO SAMPLE");
    }
    else if (argc == 3) {
        strcpy (userid, argv[1]);

```

## Beispiele für EXPORT-Sitzungen

```
        strcpy (passwd, argv[2]);
        EXEC SQL CONNECT TO sample USER :userid USING :passwd;
        CHECKERR ("CONNECT TO SAMPLE");
    }
    else {
        printf ("\nUSAGE: impexp [userid passwd]\n\n");
        return 1;
    } /* endif */

printf ("exporting NAME and ID from STAFF table into file '%s'\n", datafile);
/*****\
 * EXPORT API called *
 \*****/
sqluexpr (datafile, NULL, NULL, &columnData, columnStringPointer,
          fileFormat, NULL, msgfile_x, 0, &outputInfo, NULL, &sqlca);
CHECKERR ("exporting table");
printf ("rows exported %d\n", outputInfo.rowsExported);
free (columnStringPointer);

/*****\
 * need to allocate the proper amount of space for the SQL statement *
 \*****/
columnStringPointer = (struct sqlchar *)malloc(strlen(impStatement)
        + sizeof (struct sqlchar));
columnStringPointer->length = strlen(impStatement);
strcpy (columnStringPointer->data, impStatement, strlen(impStatement));

printf ("creating a temporary table 'imptable' to import into\n");
/* create a temporary table to import into */
EXEC SQL CREATE TABLE imptable (name VARCHAR(15), id INT);
CHECKERR ("CREATE TABLE");

printf ("importing the file '%s' into the 'imptable'\n", datafile);
/*****\
 * IMPORT API called *
 \*****/
sqluimpr (datafile, NULL, &columnData, columnStringPointer, fileFormat,
          NULL, msgfile_m, 0, &impInput, &impOutput, NULL, NULL, &sqlca);
CHECKERR ("importing table");
printf ("rows imported %d\nnumber of rows committed %d\n",
        impOutput.rowsInserted, impOutput.rowsCommitted);

free (columnStringPointer);

/* drop the table */
EXEC SQL DROP TABLE imptable;
CHECKERR ("DROP TABLE");

EXEC SQL CONNECT RESET;
CHECKERR ("CONNECT RESET");
}
/* end of program : impexp.sqc */
```

### Einschränkungen

Für das Dienstprogramm EXPORT gelten die folgenden Rahmenbedingungen:

- Dieses Dienstprogramm bietet keine Unterstützung für Kurznamen.
- Tabellen mit Spalten strukturierten Typs werden nicht unterstützt.

---

### Fehlerbehebung

Bei DB2-Operationen wie dem Exportieren, Importieren, Laden, Binden oder Wiederherstellen von Daten können Sie angeben, daß Nachrichtendateien erstellt werden, die Fehlernachrichten, Warnungen und informative Nachrichten zu diesen Operationen enthalten. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an.

Diese Nachrichtendateien sind gewöhnliche ASCII-Dateien. Jede Nachricht in einer Nachrichtendatei beginnt in einer neuen Zeile und enthält Informationen, die von der DB2-Funktion zum Abrufen von Nachrichten bereitgestellt werden. Sie können sie mit dem bei Ihrem Betriebssystem üblichen Verfahren drucken oder mit einem beliebigen ASCII-Editor anzeigen.





---

## Kapitel 2. IMPORT

In diesem Kapitel wird das DB2 UDB-Dienstprogramm IMPORT beschrieben, das die SQL-Anweisung INSERT verwendet, um Daten aus einer Eingabedatei in eine Tabelle oder Sicht zu schreiben. Wenn die Zieltabelle oder -sicht bereits Daten enthält, können Sie die importierten Daten entweder an die vorhandenen Daten anfügen oder die vorhandenen Daten ersetzen.

Die folgenden Themen werden behandelt:

- „IMPORT - Übersicht“ auf Seite 30
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT“ auf Seite 31
- „Verwenden von IMPORT“ auf Seite 32
- „Verwenden von IMPORT mit gepufferten INSERT-Operationen“ auf Seite 34
- „Verwenden von IMPORT mit Identitätsspalten“ auf Seite 34
- „Verwenden von IMPORT mit generierten Spalten“ auf Seite 36
- „Erneutes Erstellen einer exportierten Tabelle“ auf Seite 38
- „Importieren von großen Objekten (LOBs)“ auf Seite 39
- „Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs)“ auf Seite 40
- „Sperren von Tabellen während des Imports“ auf Seite 40
- „IMPORT - Befehl“ auf Seite 41
- „Import - API“ auf Seite 49
- „Dateitypwerte (IMPORT)“ auf Seite 63
- „Hinweise zu Zeichensatz und NLS“ auf Seite 74
- „Beispiele für IMPORT-Sitzungen“ auf Seite 75
- „Optimieren der IMPORT-Leistung“ auf Seite 77
- „Rahmenbedingungen und Einschränkungen“ auf Seite 78
- „Fehlerbehebung“ auf Seite 78.

Weitere Informationen zum Importieren von DB2 Data Links Manager-Daten finden Sie in „Versetzen von DB2 Data Links Manager-Daten mit IMPORT“ auf Seite 206. Weitere Informationen zum Importieren von Daten aus typisierten Tabellen finden Sie in „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 219.

Weitere Informationen zum Importieren von Daten aus einer Datei auf der DB2 Connect-Workstation in eine DRDA-Server-Datenbank und umgekehrt finden Sie in „Versetzen von Daten mit DB2 Connect“ auf Seite 217.

---

## IMPORT - Übersicht

Das Dienstprogramm IMPORT fügt Daten aus einer Eingabedatei in eine Tabelle oder aktualisierbare Sicht ein. Wenn die Tabelle oder Sicht, die die importierten Daten aufnehmen soll, bereits Daten enthält, können Sie die importierten Daten entweder an die vorhandenen Daten anfügen oder die vorhandenen Daten ersetzen.

Folgende Informationen werden benötigt, wenn Daten importiert werden:

- Der Pfad und der Name der Eingabedatei.
- Der Name oder Aliasname der Zieltabelle oder -sicht.
- Das Format der Daten in der Eingabedatei. Mögliche Formate sind: IXF, WSF, DEL oder ASC. Siehe „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.
- Ob die Eingabedaten in die Tabelle oder Sicht eingefügt werden oder ob die vorhandenen Daten in der Tabelle oder Sicht durch die Eingabedaten aktualisiert oder ersetzt werden sollen.
- Ein Nachrichtendateiname, wenn das Dienstprogramm über die Anwendungsprogrammierschnittstelle (API) **sqlimpr** aufgerufen wird.
- Beim Arbeiten mit typisierten Tabellen müssen Sie möglicherweise die Methode oder Reihenfolge für den Durchgang durch alle strukturierten Typen angeben. Die Reihenfolge des Durchgangs durch alle übergeordneten und untergeordneten Tabellen in der Hierarchie von oben nach unten und von links nach rechts wird als *Traversierfolge* bezeichnet. Diese Reihenfolge ist wichtig beim Versetzen von Daten zwischen Tabellenhierarchien, weil dabei im Verhältnis zu anderen Daten ermittelt wird, wohin die Daten versetzt werden.

Beim Arbeiten mit typisierten Tabellen müssen Sie möglicherweise auch die Liste der untergeordneten Tabellen angeben. Diese Liste gibt an, in welche untergeordneten Tabellen und Attribute Daten importiert werden sollen.

Weitere Informationen finden Sie in „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 219.

Sie können auch folgendes angeben:

- Die Methode, die zum Importieren der Daten verwendet werden soll: Spaltenposition, Spaltenname oder relative Spaltenposition.
- Die Anzahl der mit INSERT einzufügenden Zeilen, bevor die Tabellenänderungen festgeschrieben werden. Wenn Sie in regelmäßigen Abständen COMMIT-Operationen ausführen, reduziert sich die Anzahl der Zeilen, die verloren gehen, wenn während der Importoperation ein Fehler auftritt und eine ROLLBACK-Operation ausgeführt wird. Dadurch wird auch verhindert, daß DB2-Protokolle bei der Verarbeitung einer großen Eingabedatei voll werden.
- Die Anzahl der Datensätze in der Datei, die vor Beginn der Importoperation übersprungen werden sollen. Im Falle eines Fehlers können Sie die Importoperation unmittelbar nach der letzten Zeile, die erfolgreich importiert und festgeschrieben wurde, erneut starten.
- Die Namen der Spalten innerhalb der Tabelle oder Sicht, in die die Daten eingefügt werden sollen.

---

### **Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von IMPORT**

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie Operationen von Dienstprogrammen und zur Datenbankmanagerpflege auf höherer Ebene zu gruppieren. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie die entsprechende Berechtigung besitzen, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Zum Erstellen einer neuen Tabelle mit dem Dienstprogramm IMPORT benötigen Sie die Berechtigung SYSADM, DBADM oder das Zugriffsrecht CREATE-TAB für die Datenbank. Zum Ersetzen von Daten in einer vorhandenen Tabelle oder Sicht benötigen Sie die Berechtigung SYSADM, DBADM oder das Zugriffsrecht CONTROL für die Tabelle oder Sicht. Zum Anhängen von Daten an eine vorhandene Tabelle oder Sicht benötigen Sie die Zugriffsrechte SELECT und INSERT für die Tabelle oder Sicht.

## Verwenden von IMPORT

---

### Verwenden von IMPORT

#### Vor der Verwendung von IMPORT

Bevor Sie das Dienstprogramm IMPORT ausführen, müssen Sie mit der Datenbank, in die die Daten importiert werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm die Anweisung COMMIT oder ROLLBACK absetzt, sollten Sie vor dem Aufrufen von IMPORT alle Transaktionen beenden und alle Sperren aufheben, indem Sie die Anweisung COMMIT oder ROLLBACK ausführen.

#### Aufrufen von IMPORT

Das Dienstprogramm IMPORT kann wie folgt aufgerufen werden:

- Über den Befehlszeilenprozessor (CLP).

Es folgt ein Beispiel für den Befehl IMPORT, der über den CLP abgesetzt wird:

```
db2 import from stafftab.ixf of ixf insert into userid.staff
```

- Über das Notizbuch **Importieren** in der Steuerzentrale. Gehen Sie wie folgt vor, um das Notizbuch **Importieren** zu öffnen:
  1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie den Ordner **Tabellen** sehen.
  2. Klicken Sie den Ordner **Tabellen** an. Alle vorhandenen Tabellen werden auf der rechten Seite des Fensters, dem Inhaltsteilfenster, angezeigt.
  3. Klicken Sie im Inhaltsteilfenster die gewünschte Tabelle mit Maustaste 2 an, und wählen Sie **Importieren** im Kontextmenü aus. Das Notizbuch **Importieren** wird geöffnet.

Basisinformationen zur Steuerzentrale finden Sie im Handbuch *Systemverwaltung*. Zusatzinformationen bietet die Online-Hilfefunktion innerhalb der Steuerzentrale.

- Über eine Anwendungsprogrammierschnittstelle (API) **sqlimpr**. Informationen zu dieser API finden Sie in „Import - API“ auf Seite 49. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, finden Sie im Handbuch *Application Building Guide*.

### Verwenden von IMPORT in einer Client/Server-Umgebung

Wenn Sie eine Datei in eine ferne Datenbank importieren, kann eine gespeicherte Prozedur aufgerufen werden, um den Import auf dem Server auszuführen. In folgenden Situationen wird jedoch keine gespeicherte Prozedur aufgerufen:

- Die Anwendung und die Datenbank verwenden unterschiedliche Codepages.
- Die Datei, die importiert wird, ist eine aus mehreren Teilen bestehende PC/IXF-Datei.
- Die Methode, die zum Importieren der Daten verwendet werden soll, beruht entweder auf dem Spaltennamen oder der relativen Spaltenposition.
- Die bereitgestellte Zielspaltenliste ist länger als 4 KB.
- Ein OS/2- oder DOS-Client importiert eine Datei von einer Diskette.
- Die Klausel LOBS FROM oder der Wert `lobsinfile` ist angegeben.
- Die Klausel NULL INDICATORS ist für ASC-Dateien angegeben.

Beim Importieren mit Hilfe einer gespeicherten Prozedur werden in der Nachrichtendatei Nachrichten in der auf dem Server installierten Standardsprache erstellt. Die Nachrichten liegen in der Sprache der Anwendung vor, wenn Client und Server dieselbe Sprache benutzen.

Das Dienstprogramm IMPORT erstellt zwei temporäre Dateien im Unterverzeichnis `tmp` des Verzeichnisses `sqllib` (oder des Verzeichnisses, das durch die Registrierungsdatenbankvariable **DB2INSTPROF** angegeben wird). Eine Datei ist für die Daten vorgesehen, die andere Datei für die Nachrichten, die das Dienstprogramm IMPORT generiert.

Wenn beim Schreiben oder Öffnen von Daten auf dem Server ein Fehler gemeldet wird, prüfen Sie, ob folgende Bedingungen erfüllt sind:

- Das Verzeichnis existiert.
- Der Plattenspeicherplatz für die Dateien ist groß genug.
- Der Exemplareigner hat Schreibzugriff auf das Verzeichnis.

## Verwenden von IMPORT mit gepufferten INSERT-Operationen

---

### Verwenden von IMPORT mit gepufferten INSERT-Operationen

In einer partitionierten Datenbankumgebung kann das Dienstprogramm IMPORT für die Verwendung gepuffertener INSERT-Operationen aktiviert werden. Das verringert die Nachrichtenübertragung, die beim Importieren der Daten stattfindet, was zu einer besseren Leistung führt. Da jedoch keine Details zu einer fehlgeschlagenen gepufferten INSERT-Operation zurückgegeben werden, sollte diese Option nur aktiviert werden, wenn die Meldung von Fehlern nicht von Bedeutung ist.

Verwenden Sie das DB2-Dienstprogramm BIND, um gepufferte INSERT-Operationen anzufordern. Das IMPORT-Paket db2uimp.bnd muß mit der Option INSERT BUF erneut an die Datenbank gebunden werden. Beispiel:

```
db2 connect to ihre-datenbank
db2 bind db2uimp.bnd insert buf
```

**Anmerkung:** Die Funktion für gepufferte INSERT-Operationen wird bei jeder Importoperation inaktiviert, bei der der Parameter INSERT\_UPDATE angegeben ist.

---

### Verwenden von IMPORT mit Identitätsspalten

Mit dem Dienstprogramm IMPORT können Sie Daten in eine Tabelle importieren, die eine Identitätsspalte enthält. Sofern keine identitätsbezogenen Dateitypwerte verwendet werden, liegen der Ausführung des Dienstprogramms die folgenden Regeln zugrunde:

- Ist die Identitätsspalte als GENERATED ALWAYS definiert, wird für eine Tabellenzeile immer dann ein Identitätswert generiert, wenn die entsprechende Zeile in der Eingabedatei keinen Wert für die Identitätsspalte enthält oder einen Nullwert explizit vorgibt. Ist für die Identitätsspalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Ist die Identitätsspalte als GENERATED BY DEFAULT definiert, verwendet das Dienstprogramm IMPORT benutzerdefinierte Werte, sofern diese vorhanden sind. Wenn diese Daten fehlen oder explizit Nullwerte sind, wird ein Wert generiert.

Das Dienstprogramm IMPORT führt keine zusätzliche Gültigkeitsprüfung der benutzerdefinierten Identitätswerte durch, die über die übliche Gültigkeitsprüfung für Werte mit dem Datentyp der Identitätsspalte (also SMALLINT, INT, BIGINT oder DECIMAL) hinausgeht. Falls Werte doppelt vorhanden sind, wird keine entsprechende Meldung ausgegeben. Außerdem kann der Wert compound=x beim Importieren von Daten in eine Tabelle mit einer Identitätsspalte nicht verwendet werden.

## Verwenden von IMPORT mit Identitätsspalten

Das Dienstprogramm IMPORT unterstützt zwei Dateitypwerte, die die Verwendung dieses Dienstprogramms für Tabellen mit einer Identitätsspalte vereinfachen:

- Der Wert `identitymissing` vereinfacht das Importieren einer Tabelle mit einer Identitätsspalte, wenn die Eingabedatendatei für die Identitätsspalte keine Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 char(30),
                    c2 int generated by default as identity,
                    c3 real,
                    c4 char(1))
```

Ein Benutzer möchte Daten aus einer Datei (`import.del`) in die Tabelle `TABLE1` importieren. Diese Daten wurden aus einer Tabelle exportiert, die keine Identitätsspalte enthält. Es folgt ein Beispiel für eine solche Datei.

```
Robert, 45.2, J
Mike, 76.9, K
Leo, 23.4, I
```

Eine Methode für das Importieren dieser Datei wäre das explizite Auflisten der zu importierenden Spalten durch den folgenden Befehl `IMPORT`:

```
db2 import from import.del of del replace into table1 (c1, c3, c4)
```

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Importieren der Datei ist die folgende Verwendung des Dateityp-werts `identitymissing`:

```
db2 import from import.del of del modified by identitymissing
replace into table1
```

- Der Wert `identityignore` stellt in gewisser Hinsicht das Gegenteil des Werts `identitymissing` dar: Er weist das Dienstprogramm `IMPORT` an, daß die in der Eingabedatendatei vorhandenen Werte für die Identitätsspalte ignoriert werden sollen und daß für jede Zeile ein Identitätswert generiert werden soll. Beispiel: Ein Benutzer möchte die folgenden Daten aus einer Datei (`import.del`) in die wie oben beschrieben definierte Tabelle `TABLE1` importieren:

```
Robert, 1, 45.2, J
Mike, 2, 76.9, K
Leo, 3, 23.4, I
```

Wenn die benutzerdefinierten Werte 1, 2 und 3 nicht für die Identitätsspalte verwendet werden sollen, könnte der Benutzer den folgenden Befehl `IMPORT` absetzen:

```
db2 import from import.del of del method P(1, 3, 4) replace into table1
(c1, c3, c4)
```

## Verwenden von IMPORT mit Identitätsspalten

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Wert `identityignore` vereinfacht die Syntax folgendermaßen:

```
db2 import from import.del of del modified by identityignore replace
into table1
```

Beim Exportieren einer Tabelle mit einer Identitätsspalte in eine IXF-Datei können die Optionen `REPLACE_CREATE` und `CREATE` des Befehls `IMPORT` verwendet werden, um die Tabelle, einschließlich ihrer Identitätsspaltenmerkmale, erneut zu erstellen. Wird eine IXF-Datei aus einer Tabelle mit einer als `GENERATED ALWAYS` definierten Identitätsspalte erstellt, kann die Datendatei nur dann erfolgreich importiert werden, wenn der Wert `identityignore` angegeben wird. Andernfalls werden alle Zeilen zurückgewiesen (SQL3550W).

---

## Verwenden von IMPORT mit generierten Spalten

Mit dem Dienstprogramm `IMPORT` können Sie Daten in eine Tabelle importieren, die generierte Spalten enthält, welche keine Identitätsspalten sind.

Sofern keine Dateitypwerte verwendet werden, die sich auf generierte Spalten beziehen, liegen der Ausführung des Dienstprogramms die folgenden Regeln zugrunde:

- Für eine generierte Spalte wird immer dann ein Wert generiert, wenn die entsprechende Zeile in der Eingabedatei keinen Wert für die Spalte enthält oder einen Nullwert explizit vorgibt. Ist für eine generierte Spalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Generiert der Server einen Nullwert für eine generierte Spalte, die keinen Nullwert enthalten darf, wird die Datenzeile, zu der dieses Feld gehört, zurückgewiesen (SQL0407N). Dies könnte beispielsweise dann eintreten, wenn eine generierte Spalte, die keinen Nullwert enthalten darf, als Summe zweier Tabellenspalten definiert wurde und für diese Spalten in der Eingabedatei Nullwerte zur Verfügung gestellt wurden.

Das Dienstprogramm `IMPORT` unterstützt zwei Dateitypwerte, die die Verwendung dieses Dienstprogramms für Tabellen mit generierten Spalten vereinfachen:

- Der Wert `generatedmissing` vereinfacht das Importieren einer Tabelle mit generierten Spalten, wenn die Eingabedatendatei für alle in der Tabelle vorhandenen generierten Spalten keine Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:



## Verwenden von IMPORT mit generierten Spalten

```
create table table1 (c1 int,
                    c2 int,
                    g1 int generated always as (c1 + c2),
                    g2 int generated always as (2 * c1),
                    c3 char(1))
```

Ein Benutzer möchte Daten aus einer Datei (load.del) in die Tabelle TABLE1 importieren. Diese Daten wurden aus einer Tabelle exportiert, die keine generierten Spalten enthält. Es folgt ein Beispiel für eine solche Datei.

```
1, 5, J
2, 6, K
3, 7, I
```

Eine Methode für das Importieren dieser Datei wäre das explizite Auflisten der zu importierenden Spalten durch den folgenden Befehl IMPORT:

```
db2 import from import.del of del replace into table1 (c1, c2, c3)
```

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Importieren der Datei ist die folgende Verwendung des Dateityp-werts generatedmissing:

```
db2 import from import.del of del modified by generatedmissing
replace into table1
```

- Der Wert generatedignore stellt in gewisser Hinsicht das Gegenteil des Werts generatedmissing dar: Er weist das Dienstprogramm IMPORT an, daß die in der Eingabedatendatei vorhandenen Werte für alle generierten Spalten ignoriert und für jede Zeile Werte generiert werden sollen. Beispiel: Ein Benutzer möchte die folgenden Daten aus einer Datei (import.del) in die wie oben beschriebene definierte Tabelle TABLE1 importieren:

```
1, 5, 10, 15, J
2, 6, 11, 16, K
3, 7, 12, 17, I
```

Die benutzerdefinierten Werte 10, 11, und 12 (für g1) sowie 15, 16 und 17 (für g2), die keine Nullwerte sind, bewirken, daß die Zeile zurückgewiesen wird (SQL3550W). Um dies zu verhindern, könnte der Benutzer den folgenden Befehl IMPORT absetzen:

```
db2 import from import.del of del method P(1, 2, 5)
replace into table1 (c1, c2, c3)
```

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Wert generatedignore vereinfacht die Syntax folgendermaßen:

```
db2 import from import.del of del modified by generatedignore
replace into table1
```

### Erneutes Erstellen einer exportierten Tabelle

Mit dem Dienstprogramm IMPORT können Sie eine Tabelle erneut erstellen, die mit dem Dienstprogramm EXPORT gesichert wurde. Die Tabelle muß in eine IXF-Datei exportiert worden sein, und die bei der Exportoperation verwendete SELECT-Anweisung muß bestimmte Bedingungen erfüllt haben (z. B. dürfen keine Spaltennamen in der SELECT-Klausel verwendet werden; es ist nur `select *` erlaubt). Beim Erstellen einer Tabelle aus einer IXF-Datei bleiben nicht alle Attribute der Originaltabelle erhalten. Zum Beispiel werden referentielle Integritätsbedingungen, Fremdschlüsseldefinitionen und benutzerdefinierte Datentypen nicht beibehalten. Attribute der Originaltabelle, die erhalten bleiben, sind folgende:

- Name und Definition des Primärschlüssels
- Namen und Definitionen von eindeutigen Integritätsbedingungen, jedoch keine anderen Arten von Integritätsbedingungen oder Auslösern
- Spalteninformationen:
  - Spaltenname
  - Spaltendatentypen, einschließlich benutzerdefinierte einzigartige Datentypen, die als Basistyp beibehalten werden
  - Identitätsmerkmale
  - Längen (außer bei Typen `lob_file`)
  - Codepage (sofern zutreffend)
  - DATALINK-Optionen
  - Identitätsoptionen
  - Angabe, ob die Spalte laut Definition Nullwerte enthalten darf oder nicht
  - Standardwerte für Konstanten, sofern vorhanden, jedoch keine anderen Arten von Standardwerten
- Indexinformationen:
  - Indexname
  - Name des Indexerstellers
  - Spaltennamen sowie die Angabe, ob die jeweilige Spalte in aufsteigenden oder absteigender Reihenfolge sortiert wird
  - Angabe, ob der Index als eindeutig definiert ist
  - Angabe, ob es sich um einen Clusterindex handelt
  - Angabe, ob der Index Rückwärtsdurchsuchen zuläßt
  - Werte für *pctfree*
  - Werte für *minpctused*

Die folgenden Attribute der Originaltabelle bleiben *nicht* erhalten:

- Angabe, ob die Quelle eine normale Tabelle, eine Übersichtstabelle, eine Sicht oder eine Spaltengruppe aus einer oder allen diesen Quellen war

- Tabelleninformationen:
  - Definition der Übersichtstabelle (sofern zutreffend)
  - Optionen für die Übersichtstabelle (sofern zutreffend)
  - Optionen für den Tabellenbereich (diese Informationen können jedoch durch den Befehl IMPORT angegeben werden)
- Spalteninformationen:
  - Gegebenenfalls vorhandene Standardwerte (außer Werte für Konstanten)
  - LOB-Optionen (sofern vorhanden)
  - Verweisklausel der Anweisung CREATE TABLE (sofern vorhanden)
  - Referentielle Integritätsbedingungen (sofern vorhanden)
  - Prüfung auf Integritätsbedingungen (sofern vorhanden)
  - Optionen für generierte Spalten (sofern vorhanden)
- 
- Indexinformationen:
  - INCLUDE-Spalten (sofern vorhanden)

---

### Importieren von großen Objekten (LOBs)

Beim Importieren von Daten in LOB-Spalten können die betreffenden Daten entweder aus derselben Datei, aus der auch die übrigen Spaltendaten entnommen werden, oder aus separaten Dateien kommen. Im zweiten Fall gibt es eine Datei für jedes LOB-Exemplar.

Die Spalte in der Haupteingabedatei enthält entweder die Importdaten (Standardwert) oder den Namen einer Datei, in der die Importdaten gespeichert sind.

#### **Anmerkungen:**

1. Wenn LOB-Daten in der Haupteingabedatendatei gespeichert sind, sind nur 32 KB Daten zulässig. Warnungen, daß Daten abgeschnitten werden, werden ignoriert.
2. Alle LOB-Daten müssen in der Hauptdatei oder jedes große Objekt muß in einer separaten Datei gespeichert sein. Die Hauptdatei kann keine Mischung aus LOB-Daten und Dateinamen enthalten. LOB-Werte werden mit Hilfe des Wertes `lobsinfile` (siehe „Dateitypwerte (IMPORT)“ auf Seite 63) und der Klausel `LOBS FROM` (siehe „IMPORT - Befehl“ auf Seite 41) aus separaten Dateien importiert.

## Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs)

---

### Importieren von benutzerdefinierten einzigartigen Datentypen (UDTs)

Das Dienstprogramm IMPORT setzt die benutzerdefinierten einzigartigen Datentypen (UDTs) automatisch in ähnliche Basisdatentypen um. Dies erspart Ihnen das explizite Umsetzen der UDTs in die Basisdatentypen. Durch die Umsetzung werden Vergleiche zwischen UDTs und den Basisdatentypen in SQL ermöglicht.

---

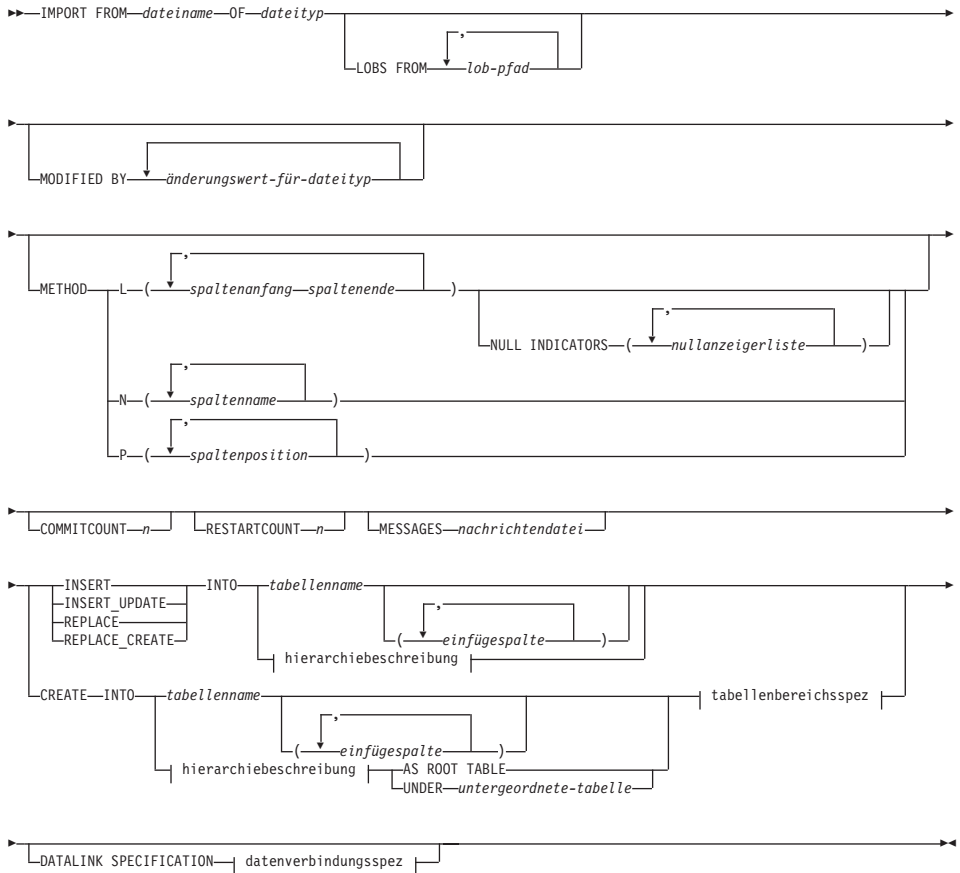
### Sperren von Tabellen während des Imports

In der Standardeinstellung ist das Dienstprogramm IMPORT an die Datenbank mit der Isolationsstufe RR (Repeatable Read - Wiederholtes Lesen) gebunden. Wird eine große Anzahl von Zeilen in eine Tabelle importiert, kann sich die vorhandene Sperre zu einer exklusiven Sperre erweitern. Wenn eine andere Anwendung, die mit derselben Tabelle arbeitet, einige Zeilensperren hält, tritt ein gegenseitiges Sperren auf, falls sich die Sperre zu einer exklusiven Sperre erweitert. Um dies zu verhindern, fordert das Dienstprogramm IMPORT zu Beginn der Verarbeitung eine exklusive Sperre für die Tabelle an.

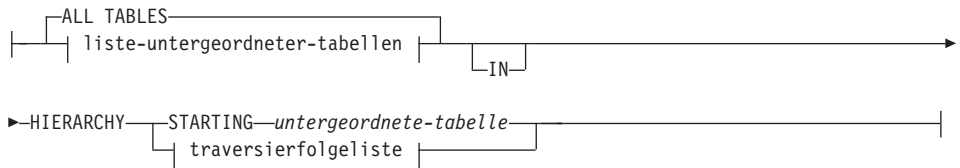
Das Halten einer Sperre für die Tabelle hat zwei Auswirkungen. Erstens wartet das Dienstprogramm IMPORT, wenn andere Anwendungen eine Tabellensperre halten oder für die Zieltabelle der Importoperation Zeilensperren vorhanden sind, bis alle diese Anwendungen ihre Änderungen festschreiben oder rückgängig machen. Zweitens warten während der Ausführung der Importoperation alle anderen Anwendungen, die eine Sperre anfordern, bis die Importoperation vollständig ausgeführt wurde.

IMPORT - Befehl

Befehlssyntax

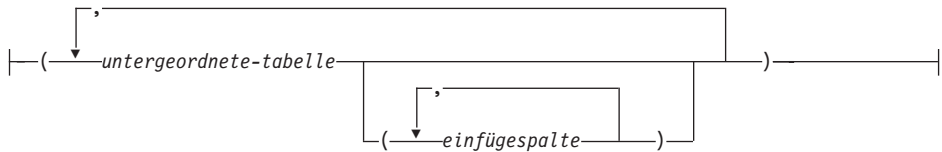


hierarchiebeschreibung:



liste-untergeordneter-tabellen:

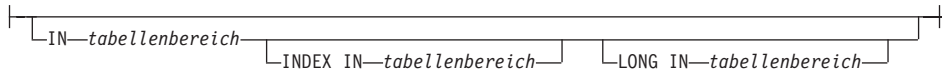
## IMPORT - Befehl



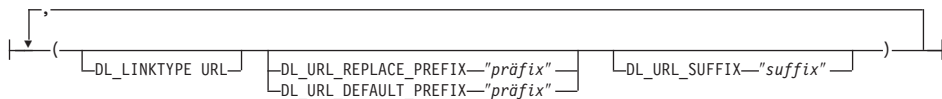
### traversierfolgeliste:



### tabellenbereichsspez:



### datenverbindungsspez:



## Befehlsparameter

### ALL TABLES

Ein implizites Schlüsselwort nur für die Hierarchie. Beim Importieren einer Hierarchie werden standardmäßig alle Tabellen importiert, die in einer Traversierfolge angegeben sind.

### AS ROOT TABLE

Erstellt eine oder mehrere untergeordnete Tabellen als eigenständige Tabellenhierarchie.

### COMMITCOUNT *n*

Führt eine COMMIT-Operation aus, nachdem jeweils *n* Datensätze importiert wurden.

### CREATE

Erstellt die Tabellendefinition und den Zeileninhalt. Wenn die Daten aus einer DB2-Tabelle, untergeordneten Tabelle oder Hierarchie exportiert wurden, werden Indizes erstellt. Wenn diese Option für eine

Hierarchie verwendet wird und Daten aus DB2 exportiert wurden, wird auch eine Typhierarchie erstellt. Diese Option kann nur mit IXF-Dateien verwendet werden.

**Anmerkung:** Wenn die Daten aus einer MVS-Host-Datenbank exportiert wurden und LONGVAR-Felder enthalten, deren Länge, berechnet anhand der Seitengröße, kürzer ist als 254, kann CREATE fehlschlagen, weil die Zeilen zu lang sind. In diesem Fall muß die Tabelle manuell erstellt und IMPORT mit INSERT aufgerufen werden. Alternativ kann auch der Befehl LOAD verwendet werden.

### DATALINK SPECIFICATION

Für jede DATALINK-Spalte ist eine Spaltenangabe in runden Klammern möglich. Jede Spaltenangabe umfaßt mindestens einen DL\_LINKTYPE-Wert, eine Präfix- und eine DL\_URL\_SUFFIX-Angabe. Die Präfixangabe kann entweder DL\_URL\_REPLACE\_PREFIX oder DL\_URL\_DEFAULT\_PREFIX sein.

Es sind so viele DATALINK-Spaltenangaben möglich, wie DATALINK-Spalten in der Tabelle definiert sind. Die Reihenfolge der Angaben richtet sich nach der Reihenfolge der DATALINK-Spalten in der Liste *einfügespalte* oder in der Tabellendefinition (wenn keine Liste *einfügespalte* angegeben wird).

### DL\_LINKTYPE

Muß, wenn angegeben, mit dem LINKTYPE-Wert der Spaltendefinition übereinstimmen. Demzufolge ist DL\_LINKTYPE URL akzeptabel, wenn die Spaltendefinition LINKTYPE URL angibt.

### DL\_URL\_DEFAULT\_PREFIX "präfix"

Muß, wenn angegeben, als Standardpräfix für alle DATALINK-Werte in der gleichen Spalte verwendet werden. In diesem Zusammenhang bezieht sich Präfix auf den Teil „Schema-Host-Anschluß“ der URL-Angabe. (Bei einem verteilten Dateisystem bezieht sich Präfix auf den Teil „Schema-Zellenname-Dateibereichsverbindung“ der URL-Angabe.)

Beispiele für Präfixe sind:

```
"http://server"
"file://server"
"file:"
"http://server:80"
"dfs://.../cellname/fs"
```

Enthalten die Daten einer Spalte kein Präfix und wird mit DL\_URL\_DEFAULT\_PREFIX ein Standardpräfix angegeben, wird dieses dem Spaltenwert (sofern nicht NULL) vorangestellt.

## IMPORT - Befehl

Beispiel: Wenn `DL_URL_DEFAULT_PREFIX` das Standardpräfix "`http://toronto`" angibt, gilt folgendes:

- Der Spalteneingabewert "`/x/y/z`" wird als "`http://toronto/x/y/z`" gespeichert.
- Der Spalteneingabewert "`http://coyote/a/b/c`" wird als "`http://coyote/a/b/c`" gespeichert.
- Der Spalteneingabewert `NULL` wird als `NULL` gespeichert.

### **DL\_URL\_REPLACE\_PREFIX "präfix"**

Diese Klausel ist zum Laden oder Importieren von Daten nützlich, die zuvor vom Dienstprogramm `EXPORT` generiert wurden, wenn der Benutzer den Host-Namen in den Daten global durch einen anderen Host-Namen ersetzen möchte. Wird, wenn angegeben, das Präfix für *alle* Spaltenwerte, die nicht `NULL` sind. Wenn ein Spaltenwert mit einem Präfix versehen ist, wird dieses durch das mit `DL_URL_REPLACE_PREFIX` angegebene Präfix ersetzt. Wenn ein Spaltenwert kein Präfix aufweist, wird das mit `DL_URL_REPLACE_PREFIX` angegebene Präfix dem Spaltenwert vorangestellt. Bei einem verteilten Dateisystem bezieht sich Präfix auf den Teil „Schema-Zellenname-Dateibereichsverbindung“ der URL-Angabe.

Beispiel: Wenn `DL_URL_REPLACE_PREFIX` das Präfix "`http://toronto`" angibt, gilt folgendes:

- Der Spalteneingabewert "`/x/y/z`" wird als "`http://toronto/x/y/z`" gespeichert.
- Der Spalteneingabewert "`http://coyote/a/b/c`" wird als "`http://toronto/a/b/c`" gespeichert. Beachten Sie, daß "`toronto`" "`coyote`" ersetzt.
- Der Spalteneingabewert `NULL` wird als `NULL` gespeichert.

### **DL\_URL\_SUFFIX "suffix"**

Wird, wenn angegeben, an jeden Spaltenwert für die Spalte, der nicht `NULL` ist, angefügt. Wird an die Pfadkomponente des URL-Teils des `DATALINK`-Werts angefügt.

### **FROM dateiname**

Gibt die Datei an, die die zu importierenden Daten enthält. Wenn der Pfad weggelassen wird, wird das aktuelle Arbeitsverzeichnis verwendet.

### **HIERARCHY**

Gibt an, daß hierarchische Daten importiert werden sollen.

### **IN tabellenbereich**

Gibt den Tabellenbereich an, in dem die Tabelle erstellt wird. Der Tabellenbereich muß vorhanden und ein Tabellenbereich des Typs `REGULAR` sein. Wenn kein anderer Tabellenbereich angegeben ist,



werden alle Teile der Tabelle in diesem Tabellenbereich gespeichert. Wenn diese Klausel nicht angegeben wird, wird die Tabelle in einem Tabellenbereich erstellt, der von der Berechtigungs-ID erstellt wird. Wenn keiner gefunden wird, wird die Tabelle in den Standardtabellenbereich `USERSPACE1` gestellt. Wenn `USERSPACE1` gelöscht wurde, schlägt die Erstellung der Tabelle fehl.

#### **INDEX IN tabellenbereich**

Gibt den Tabellenbereich an, in dem Indizes für die Tabelle erstellt werden. Diese Option ist nur zulässig, wenn der primäre Tabellenbereich, der in der `IN`-Klausel angegeben ist, ein DMS-Tabellenbereich ist. Der Tabellenbereich muß vorhanden und ein DMS-Tabellenbereich des Typs `REGULAR` sein.

**Anmerkung:** Welcher Tabellenbereich einen Index enthält, kann nur beim Erstellen der Tabelle angegeben werden.

#### **einfügespalte**

Gibt den Namen einer Spalte in der Tabelle oder der Sicht an, in die Daten eingefügt werden sollen.

#### **INSERT**

Fügt die importierten Daten in die Tabelle ein, ohne die vorhandenen Tabellendaten zu ändern.

#### **INSERT\_UPDATE**

Fügt Zeilen mit importierten Daten in die Zieltabelle ein oder aktualisiert vorhandene Zeilen (der Zieltabelle) mit entsprechenden Primärschlüsseln.

#### **INTO tabellenname**

Gibt die Datenbanktabelle an, in die die Daten importiert werden sollen. Diese Tabelle darf keine Systemtabelle, keine deklarierte temporäre Tabelle und keine Übersichtstabelle sein.

Es kann ein Aliasname für `INSERT`, `INSERT_UPDATE` oder `REPLACE` verwendet werden. Ausgenommen hiervon ist die Verwendung eines Servers mit einem älteren Versionsstand. In diesem Fall muß der vollständig qualifizierte Tabellenname oder der Tabellenname ohne Qualifikationsmerkmal verwendet werden. Ein qualifizierter Tabellenname hat das Format *schema.tabellenname*. Dabei ist *schema* der Benutzername, mit dem die Tabelle erstellt wurde.

#### **LOBS FROM lob-pfad**

Gibt einen oder mehrere Pfade an, in denen LOB-Dateien gespeichert werden. Die Namen der LOB-Datendateien werden in der Hauptdatendatei (`ASC`, `DEL` oder `IXF`) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Diese Option wird ignoriert, wenn der Änderungswert `lobsinfile` nicht angegeben wird.

## IMPORT - Befehl

### LONG IN tabellenbereich

Gibt den Tabellenbereich an, in dem die Werte von Spalten für große Objekte (LOBs) (LONG VARCHAR, LONG VARGRAPHIC, LOB-Datentypen oder einzigartige Datentypen mit einem dieser Typen als Quellentyp) gespeichert werden. Diese Option ist nur zulässig, wenn der primäre Tabellenbereich, der in der IN-Klausel angegeben ist, ein DMS-Tabellenbereich ist. Der Tabellenbereich muß vorhanden und ein DMS-Tabellenbereich des Typs LONG sein.

### MESSAGES *nachrichtendatei*

Gibt das Ziel für Warnungen und Fehlernachrichten an, die bei einer Importoperation auftreten. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm IMPORT an sie angefügt. Wenn der vollständige Pfad zu der Datei nicht angegeben wird, verwendet das Dienstprogramm das aktuelle Verzeichnis und das Standardlaufwerk als Ziel. Wenn *nachrichtendatei* weggelassen wird, werden die Nachrichten in die Standardausgabe geschrieben.

### METHOD

**L** Gibt die Anfangs- und Endnummer der Spalten an, aus denen Daten importiert werden sollen. Eine Spaltennummer ist eine relative Byteadresse in Bezug auf den Anfang einer Datenzeile. Ihre Numerierung beginnt bei 1.

**Anmerkung:** Diese Methode kann nur mit ASC-Dateien verwendet werden und ist die einzige gültige Option für diesen Dateityp.

**N** Gibt die Namen der zu importierenden Spalten an.

**Anmerkung:** Diese Methode kann nur mit IXF-Dateien verwendet werden.

**P** Gibt die Indizes (numeriert beginnend bei 1) der zu importierenden Eingabedatenfelder an.

**Anmerkung:** Diese Methode kann nur mit IXF- oder DEL-Dateien verwendet werden und ist die einzige gültige Option für den Dateityp DEL.

### MODIFIED BY *änderungswert-für-dateityp*

Gibt zusätzliche Optionen an (siehe Tabelle 5 auf Seite 63).

### NULL INDICATORS *nullanzeigerliste*

Diese Option kann nur verwendet werden, wenn der Parameter METHOD L angegeben ist, d. h. wenn es sich bei der Eingabedatei um eine ASC-Datei handelt. Die Nullanzeigerliste ist eine durch Komma getrennte Liste positiver ganzer Zahlen, die für jedes Nullan-

zeigerfeld die Spaltennummer angeben. Die Spaltennummer ist die relative Byteadresse des Nullanzeigerfelds in Bezug auf den Anfang einer Datenzeile. Die Nullanzeigerliste muß für jedes Datenfeld, das im Parameter METHOD L definiert ist, einen Wert enthalten. Eine Spaltennummer 0 gibt an, daß das entsprechende Datenfeld immer Daten enthält.

Y in der Nullanzeigerspalte gibt an, daß die Spaltendaten NULL sind. Jedes *andere* Zeichen als Y in der Nullanzeigerspalte gibt an, daß die Spaltendaten nicht NULL sind und daß die mit der Option METHOD L angegebenen Spaltendaten importiert werden.

Das Nullanzeigerzeichen kann mit der Option MODIFIED BY geändert werden (siehe die Beschreibung des Änderungswerts nullindchar in Tabelle 5 auf Seite 63).

### **OF dateityp**

Gibt das Format der Daten in der Eingabedatei an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer), das von einer Vielzahl von Datenbankmanager- und Dateimanagerprogrammen verwendet wird.
- WSF (Arbeitsblattformat), das von Programmen wie den folgenden verwendet wird:
  - Lotus 1-2-3
  - Lotus Symphony
- IXF (Integrated Exchange Format, PC-Version), d. h. es wurde aus der gleichen oder einer anderen DB2-Tabelle importiert. Eine IXF-Datei enthält auch die Tabellendefinition und Definitionen vorhandener Indizes, sofern keine Spalten in der SELECT-Anweisung angegeben sind.

Weitere Informationen zu Dateiformaten finden Sie in in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

### **REPLACE**

Löscht alle vorhandenen Daten aus der Tabelle, indem das Datenobjekt abgeschnitten wird, und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert. Diese Option kann nur verwendet werden, wenn die Tabelle vorhanden ist. Sie ist für Tabellen mit DATALINK-Spalten unzulässig. Wenn diese Option beim Versetzen von Daten zwischen Hierarchien verwendet wird, können nur die Daten für eine vollständige Hierarchie, nicht jedoch für einzelne untergeordnete Tabellen ersetzt werden.

## IMPORT - Befehl

### **REPLACE\_CREATE**

Löscht, wenn die Tabelle vorhanden ist, alle vorhandenen Daten aus der Tabelle, indem das Datenobjekt abgeschnitten wird, und fügt die importierten Daten ein, ohne die Tabellendefinition oder die Indexdefinitionen zu ändern.

Wenn die Tabelle nicht vorhanden ist, werden die Tabelle und die Indexdefinitionen sowie der Zeileninhalt erstellt.

Diese Option kann nur mit IXF-Dateien verwendet werden. Sie ist für Tabellen mit DATALINK-Spalten unzulässig. Wenn diese Option beim Versetzen von Daten zwischen Hierarchien verwendet wird, können nur die Daten für eine vollständige Hierarchie, nicht jedoch für einzelne untergeordnete Tabellen ersetzt werden.

### **RESTARTCOUNT n**

Gibt an, daß eine Importoperation bei Datensatz  $n + 1$  gestartet werden soll. Die ersten  $n$  Datensätze werden übersprungen.

### **STARTING untergeordnete-tabelle**

Ein Schlüsselwort nur für Hierarchie, das die Standardreihenfolge anfordert, beginnend mit *untergeordnete-tabelle*. Bei PC/IXF-Dateien ist die Standardreihenfolge die Reihenfolge, die in der Eingabedatei gespeichert ist. Die Standardreihenfolge ist die einzige gültige Reihenfolge für das Dateiformat PC/IXF.

### **liste-untergeordneter-tabellen**

Für typisierte Tabellen mit der Option INSERT oder INSERT\_UPDATE wird eine Liste mit untergeordneten Tabellennamen verwendet, um die untergeordneten Tabellen anzugeben, in die die Daten importiert werden sollen.

### **traversierfolgeliste**

Für typisierte Tabellen mit der Option INSERT, INSERT\_UPDATE oder REPLACE wird eine Liste mit untergeordneten Tabellennamen verwendet, um die Traversierfolge der zu importierenden untergeordneten Tabellen in der Hierarchie anzugeben.

### **UNDER untergeordnete-tabelle**

Gibt eine übergeordnete Tabelle an, wenn eine oder mehrere untergeordnete Tabellen erstellt werden.

---

**Import - API**
**C-API-Syntax**

```

/* File: sqlutil.h */
/* API: Import */
/* ... */
SQL_API_RC SQL_API_FN
sqluimpr (
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluimpt_in* pImportInfoIn,
    struct sqluimpt_out* pImportInfoOut,
    sqlint32 * pNullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */

```

**Generische API-Syntax**

```

/* File: sqlutil.h */
/* API: Import */
/* ... */
SQL_API_RC SQL_API_FN
sqlgimpr (
    unsigned short DataFileNameLen,
    unsigned short FileTypeLen,
    unsigned short MsgFileNameLen,
    char * pDataFileName,
    sqlu_media_list * pLobPathList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pMsgFileName,
    short CallerAction,
    struct sqluimpt_in* pImportInfoIn,
    struct sqluimpt_out* pImportInfoOut,
    sqlint32 * NullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */

```

### API-Parameter

#### **DataFileNameLen**

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der Eingabedatei in Byte darstellt.

#### **FileTypeLen**

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Eingabedateityps in Byte darstellt.

#### **MsgFileNameLen**

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der Nachrichtendatei in Byte darstellt.

#### **pDataFileName**

Eingabe. Eine Zeichenfolge, die den Pfad und den Namen einer externen Eingabedatei enthält, aus der die Daten importiert werden sollen.

#### **pLobPathList**

Eingabe. Eine *sqlu\_media\_list* unter Verwendung von *media\_type* `SQLU_LOCAL_MEDIA` und der Struktur *sqlu\_media\_entry*, die Pfade auf dem Client auflistet, in denen sich LOB-Dateien befinden.

Weitere Informationen finden Sie unter „SQLU-MEDIA-LIST“ im Handbuch *Administrative API Reference*.

#### **pDataDescriptor**

Eingabe. Zeiger auf eine *sqldcol*-Struktur, die Informationen zu den Spalten enthält, die zum Import aus der externen Datei ausgewählt werden. Der Wert des Feldes *dcolmeth* bestimmt, wie die restlichen in diesem Parameter bereitgestellten Informationen vom Dienstprogramm `IMPORT` ausgewertet werden. Gültige Werte für diesen Parameter (definiert in `sqlutil`) sind:

##### **SQL\_METH\_N**

Namen. Die Auswahl von Spalten in der externen Eingabedatei erfolgt nach Spaltenname.

##### **SQL\_METH\_P**

Spaltenpositionen. Die Auswahl von Spalten in der externen Eingabedatei erfolgt durch Angabe der Nummer der zu ladenen Spalten.

##### **SQL\_METH\_L**

Anfangs- und Endposition von Spalten. Die Auswahl von Spalten in der externen Eingabedatei erfolgt durch Angabe der Nummer der ersten und der letzten Position, aus der Daten geladen werden sollen. Der Datenbankmanager weist einen Importaufruf mit einem Positionspaar zurück, das aufgrund einer der folgenden Bedingungen ungültig ist:

- Entweder die Anfangs- oder die Endposition liegt nicht im Bereich von 1 bis zur größten 2 Byte großen ganzen Zahl mit Vorzeichen.
- Die Nummer der Endposition ist kleiner als die Nummer der Anfangsposition.
- Die Eingabespaltenbreite, die vom Positionspaar definiert wird, ist zum Typ und der Länge der Zielspalte nicht kompatibel.

Ein Positionspaar, bei dem beide Positionsangaben gleich Null sind, gibt an, daß eine Spalte, die Nullwerte annehmen kann, mit Nullen gefüllt werden soll.

### SQL\_METH\_D

Standardeinstellung. Wenn *pDataDescriptor* NULL oder auf SQL\_METH\_D gesetzt ist, erfolgt eine Standardauswahl von Spalten aus der externen Eingabedatei. In diesem Fall werden die Anzahl der Spalten und die Spaltenspezifikationsmatrix ignoriert. Die ersten *n* Spalten mit Daten in der externen Eingabedatei werden in der bestehenden Reihenfolge verwendet, wobei *n* für die Anzahl der Datenbankspalten steht, in die die Daten importiert werden sollen.

Weitere Informationen finden Sie unter „SQLDCOL“ im Handbuch *Administrative API Reference*.

### pActionString

Eingabe. Ein Zeiger auf eine *sqlchar*-Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die die Spalten angeben, in die die Daten importiert werden sollen.

Der Zeichenbereich besitzt folgendes Format:

```
{INSERT | INSERT_UPDATE | REPLACE | CREATE | REPLACE_CREATE}
INTO {tname[(tspaltenliste)] |
[ALL TABLES | (tname[(tspaltenliste))][,
tname[(tspaltenliste))]]}]
[IN] HIERARCHY {STARTING tname | (tname[, tname])}
[UNDER untergeordnete-tabelle | AS ROOT TABLE]
[DATA LINK SPECIFICATION datalink-spez]
```

### INSERT

Fügt die importierten Daten in die Tabelle ein, ohne die vorhandenen Tabellendaten zu ändern.

### INSERT\_UPDATE

Fügt die importierten Zeilen hinzu, wenn sich ihre Primärschlüsselwerte nicht in der Tabelle befinden, und verwendet sie zur Aktualisierung, wenn ihre Primärschlüsselwerte gefunden werden. Diese Option ist nur zulässig, wenn die Ziel-

tabelle einen Primärschlüssel enthält und die angegebene (oder implizierte) Liste mit Zielspalten, die importiert werden, alle Spalten für den Primärschlüssel enthält. Diese Option kann nicht auf Sichten angewendet werden.

### REPLACE

Löscht alle vorhandenen Daten aus der Tabelle, indem das Tabellenobjekt abgeschnitten wird, und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert. (Indizes werden gelöscht und ersetzt, wenn `indexixf` in *FileTypeMod* und *FileType SQL\_IXF* ist.) Ist die Tabelle noch nicht definiert, wird ein Fehler zurückgegeben.

**Achtung:** Wenn ein Fehler auftritt, nachdem die vorhandenen Daten gelöscht wurden, gehen die Daten verloren.

### CREATE

Erstellt die Tabellendefinition und den Zeileninhalt unter Verwendung der in der PC/IXF-Datei angegebenen Informationen, wenn die angegebene Tabelle nicht definiert ist. Wenn die Datei vorher von DB2 exportiert wurde, werden auch Indizes erstellt. Wenn die angegebene Tabelle bereits definiert ist, wird ein Fehler zurückgegeben. Diese Option ist nur für das Dateiformat PC/IXF gültig.

### REPLACE\_CREATE

Ersetzt den Tabelleninhalt durch die PC/IXF-Zeilendaten aus der PC/IXF-Datei, wenn die angegebene Tabelle definiert ist. Wenn die Tabelle noch nicht definiert ist, werden die Tabellendefinition und der Zeileninhalt unter Verwendung der Daten in der angegebenen PC/IXF-Datei erstellt. Wenn die PC/IXF-Datei vorher von DB2 exportiert wurde, werden auch Indizes erstellt. Diese Option ist nur für das Dateiformat PC/IXF gültig.

**Achtung:** Wenn ein Fehler auftritt, nachdem die vorhandenen Daten gelöscht wurden, gehen die Daten verloren.

*tname* Der Name der Tabelle, typisierten Tabelle, Sicht oder Objektsicht, in die die Daten eingefügt werden sollen. Es kann ein Aliasname für REPLACE, INSERT\_UPDATE oder INSERT angegeben werden (außer im Falle eines Servers mit einem älteren Versionsstand, in dem ein qualifizierter oder nicht qualifizierter Name angegeben werden muß). Erfolgt dies in einer Sicht, darf es keine Sicht nur mit Lesezugriff sein.



***tspaltenliste***

Eine Liste der Tabellen- oder Sichtspaltennamen, in die die Daten eingefügt werden sollen. Die Spaltennamen müssen durch Kommas getrennt werden. Wenn keine Spaltennamen angegeben werden, werden die in den Anweisungen CREATE TABLE oder ALTER TABLE definierten Spaltennamen verwendet. Wenn keine Spaltenliste für typisierte Tabellen angegeben wird, werden Daten in alle Spalten innerhalb einer jeden untergeordneten Tabelle eingefügt.

***untergeordnete-tabelle***

Gibt eine übergeordnete Tabelle an, wenn eine oder mehrere untergeordnete Tabellen mit der Option CREATE erstellt werden.

**ALL TABLES**

Ein implizites Schlüsselwort nur für die Hierarchie. Beim Importieren einer Hierarchie werden standardmäßig alle Tabellen importiert, die in *traversierfolgeliste* angegeben sind.

**HIERARCHY**

Gibt an, daß hierarchische Daten importiert werden sollen.

**STARTING**

Schlüsselwort nur für die Hierarchie. Gibt an, daß ausgehend vom Namen einer bestimmten untergeordneten Tabelle die Standardreihenfolge verwendet werden soll.

**UNDER**

Schlüsselwort nur für die Hierarchie und CREATE. Gibt an, daß die neue Hierarchie, untergeordnete Hierarchie oder untergeordnete Tabelle unter einer gegebenen untergeordneten Tabelle erstellt werden soll.

**AS ROOT TABLE**

Schlüsselwort nur für die Hierarchie und CREATE. Gibt an, daß die neue Hierarchie, untergeordnete Hierarchie oder untergeordnete Tabelle als eigenständige Hierarchie erstellt werden soll.

**DATALINK SPECIFICATION *datalink-spez***

Gibt Parameter an, die sich auf DB2 Data Links beziehen. Diese Parameter können mit der gleichen Syntax wie beim Befehl IMPORT angegeben werden (siehe „IMPORT - Befehl“ auf Seite 41).

Die Parameter *tname* und *tspaltenliste* entsprechen den Listen *tabellenname* und *spaltenname* von SQL-Anweisungen INSERT, und es gelten dafür die gleichen Einschränkungen.

Die Spalten in *tspaltenliste* und die externen Spalten (entweder angegeben oder impliziert) werden anhand ihrer Position in der Liste oder Struktur abgeglichen (Daten aus der ersten Spalte, die in der Struktur *sqliacol* angegeben ist, werden in das Tabellen- oder Sichtfeld eingefügt, das dem ersten Element von *tspaltenliste* entspricht).

Wenn eine ungleiche Anzahl von Spalten angegeben wird, wird die kleinere der angegebenen Anzahl von Spalten verarbeitet. Das kann zu einem Fehler führen (weil für einige Tabellenfelder, die keine Nullwerte annehmen dürfen, keine Werte vorhanden sind) oder eine Informationsnachricht (weil einige externe Dateispalten ignoriert werden) auslösen.

Weitere Informationen finden Sie unter „SQLCHAR“ im Handbuch *Administrative API Reference*.

### **pFileType**

Eingabe. Eine Zeichenfolge, die das Format der Daten in der externen Datei angibt. Unterstützte externe Dateiformate (definiert in `sqlutil`) sind:

#### **SQL\_ASC**

ASCII-Format mit universellen Zeilenbegrenzern.

#### **SQL\_DEL**

ASCII-Format ohne universelle Zeilenbegrenzer für den Datenaustausch mit dBase, BASIC und den IBM Personal Decision Series-Programmen und vielen anderen Datenbank- und Dateimanagern.

#### **SQL\_IXF**

PC-Version des Integrated Exchange Format. Dies ist das bevorzugte Format für den Export von Daten aus einer Tabelle, so daß sie später in dieselbe Tabelle oder eine andere Datenbankmanagertabelle importiert werden können.

#### **SQL\_WSF**

Arbeitsblattformate zum Datenaustausch mit den Programmen Lotus Symphony und 1-2-3.

Weitere Informationen zu Dateiformaten finden Sie in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

### **pFileTypeMod**

Eingabe. Ein Zeiger auf eine Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbeitungsoptionen angeben. Wenn dieser Zeiger NULL ist oder die Struk-

tur, auf die gezeigt wird, null Zeichen enthält, wird diese Aktion als Auswahl einer Standardspezifikation gewertet.

Nicht alle Optionen können mit allen unterstützten Dateitypen verwendet werden.

Weitere Informationen finden Sie unter „SQLCHAR“ im Handbuch *Administrative API Reference* und unter „Dateitypwerte (IMPORT)“ auf Seite 63.

### **pMsgFileName**

Eingabe. Eine Zeichenfolge, die den Bestimmungsort für vom Dienstprogramm zurückgegebene Fehlermeldungen, informative Nachrichten und Warnungen enthält. Dies kann der Pfad und der Name einer Betriebssystemdatei oder einer Standardeinheit sein. Wenn die Datei bereits vorhanden ist, wird an sie angefügt. Wenn sie nicht vorhanden ist, wird eine neue Datei erstellt.

### **CallerAction**

Eingabe. Eine vom Aufrufenden angeforderte Aktion. Gültige Werte (definiert in `sqlutil`) sind:

#### **SQLU\_INITIAL**

Erstaufruf. Dieser Wert muß beim ersten Aufruf der API verwendet werden.

Wenn der Erstaufruf oder ein nachfolgender Aufruf zurückgegeben wird und erfordert, daß die aufrufende Anwendung eine Aktion ausführt, bevor die angeforderte Importoperation ausgeführt wird, muß die Aufruferaktion auf eine der folgenden Optionen gesetzt werden:

#### **SQLU\_CONTINUE**

Verarbeitung fortsetzen. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebingung zu reagieren). Er gibt an, daß die vom Dienstprogramm angeforderte Benutzeraktion beendet wurde und das Dienstprogramm die Verarbeitung der Erstanforderung fortsetzen kann.

#### **SQLU\_TERMINATE**

Verarbeitung beenden. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebingung zu reagieren). Er gibt an, daß die vom Dienst-

programm angeforderte Benutzeraktion nicht ausgeführt wurde und das Dienstprogramm die Verarbeitung der Erstanforderung beenden soll.

### **pImportInfoIn**

Eingabe. Wahlfreier Zeiger auf die Struktur *sqluimpt\_in*, die zusätzliche Eingabeparameter enthält. Weitere Informationen zu dieser Struktur finden Sie in „Datenstruktur SQLUIMPT-IN“ auf Seite 60.

### **pImportInfoOut**

Ausgabe. Wahlfreier Zeiger auf die Struktur *sqluimpt\_out*, die zusätzliche Ausgabeparameter enthält. Weitere Informationen zu dieser Struktur finden Sie in „Datenstruktur SQLUIMPT-OUT“ auf Seite 61.

### **NullIndicators**

Eingabe. Nur für ASC-Dateien. Eine Matrix von ganzen Zahlen, die angeben, ob die Spaltendaten Nullwerte annehmen dürfen oder nicht. Die Anzahl der Elemente in dieser Matrix muß mit der Anzahl von Spalten in der Eingabedatei übereinstimmen. Es besteht eine 1-zu-1-Entsprechung zwischen den Elementen dieser Matrix und den Spalten, die aus der Datendatei importiert werden. Demzufolge muß die Anzahl der Elemente mit dem Feld *dcolnum* des Parameters *pDataDescriptor* übereinstimmen. Jedes Element der Matrix enthält eine Zahl, die eine Spalte in der Datendatei angibt, die als Nullanzeigerfeld verwendet werden soll, oder eine Null, um anzugeben, daß die Tabellenspalte keine Nullwerte enthalten darf. Wenn das Element nicht Null ist, muß die angegebene Spalte in der Datendatei Y oder N enthalten. Y gibt an, daß die Tabellenspaltendaten NULL sind, N gibt an, daß die Tabellenspaltendaten nicht NULL sind.

### **pReserved**

Zur zukünftigen Verwendung reserviert.

### **pSqlca**

Ausgabe. Ein Zeiger auf die Struktur *sqlca*. Weitere Informationen zu dieser Struktur finden Sie im Abschnitt über SQLCA des Handbuchs *Administrative API Reference*.

## REXX-API-Syntax

```

IMPORT FROM datafile OF filetype
[MODIFIED BY :filetmod]
[METHOD {L|N|P} USING :dcoldata]
[COMMITCOUNT :commitcnt] [RESTARTCOUNT :restartcnt]
MESSAGES msgfile
{INSERT|REPLACE|CREATE|INSERT_UPDATE|REPLACE_CREATE}
INTO tname [(:columns)]
[OUTPUT INTO :output]

CONTINUE IMPORT

STOP IMPORT

```

## REXX-API-Parameter

### datafile

Name der Datei, aus der die Daten importiert werden sollen.

### filetype

Das Format der Daten in der externen Importdatei. Folgende Dateiformate werden unterstützt:

- DEL** ASCII-Format ohne universelle Zeilenbegrenzer
- ASC** ASCII-Format mit universellen Zeilenbegrenzern
- WSF** Arbeitsblattformat
- IXF** PC-Version des Integrated Exchange Format

### filetmod

Eine Host-Variable, die zusätzliche Verarbeitungsoptionen enthält (siehe „Dateitypwerte (IMPORT)“ auf Seite 63).

### L|N|P

Ein Zeichen, das die Methode angibt, die zum Auswählen von Spalten in der externen Eingabedatei verwendet werden soll. Gültige Werte:

- L** Anfangs- und Endposition von Spalten
- N** Name
- P** Spaltenpositionen

### dcoldata

Eine zusammengesetzte REXX-Host-Variable, die Informationen zu den Spalten enthält, die in der externen Eingabedatei zum Importieren ausgewählt sind. Der Inhalt der Struktur hängt von der angegebenen *Methode* ab. Im folgenden steht XXX für den Namen der Host-Variablen:

- Anfangs- und Endposition von Spalten

**XXX.0** Anzahl der Elemente im restlichen Teil der Variablen.

**XXX.1** Eine Zahl, die die Anfangsposition dieser Spalte in der Eingabedatei angibt. Diese Spalte wird zur ersten Spalte in der Datenbanktabelle.

**XXX.2** Eine Zahl, die die Endposition der Spalte angibt.

**XXX.3** Eine Zahl, die die Anfangsposition dieser Spalte in der Eingabedatei angibt. Diese Spalte wird zur zweiten Spalte in der Datenbanktabelle.

**XXX.4** Eine Zahl, die die Endposition der Spalte angibt.

**XXX.5** und so weiter.

- **Namen**

**XXX.0** Anzahl der Spaltennamen, die in der Host-Variablen enthalten sind.

**XXX.1** Erster Name.

**XXX.2** Zweiter Name.

**XXX.3** und so weiter.

- **Spaltenpositionen**

**XXX.0** Anzahl der Spaltenpositionen, die in der Host-Variablen enthalten sind.

**XXX.1** Eine Spaltenposition in der externen Eingabedatei.

**XXX.2** Eine Spaltenposition in der externen Eingabedatei.

**XXX.3** und so weiter.

**tname** Name der Zieltabelle oder -sicht. Daten können nicht in eine Sicht importiert werden, für die nur Lesezugriff besteht.

**columns**

Eine REXX-Host-Variable, die die Namen der Spalten in der Tabelle oder Sicht enthält, in die die Daten eingefügt werden sollen. Im folgenden steht XXX für den Namen der Host-Variablen:

**XXX.0** Anzahl von Spalten.

**XXX.1** Erster Spaltenname.

**XXX.2** Zweiter Spaltenname.

**XXX.3** und so weiter.

**msgfile**

Datei-, Pfad- oder Einheitenname für Fehlernachrichten und Warnungen.

**commitcnt**

Führt eine COMMIT-Operation aus, nachdem jeweils *commitcnt* Datensätze importiert wurden.

**restartcnt**

Gibt an, daß eine Importoperation bei Datensatz *restartcnt* + 1 gestartet werden soll. Die ersten *restartcnt* Datensätze werden übersprungen.

**output**

Eine zusammengesetzte REXX-Host-Variable, an die die Daten aus der Importoperation übergeben werden. Im folgenden steht XXX für den Namen der Host-Variablen:

- XXX.1** Anzahl der Datensätze, die bei der Importoperation aus der externen Eingabedatei gelesen wurden.
- XXX.2** Anzahl der Datensätze, die vor Beginn der Einfügung oder Aktualisierung übersprungen wurden.
- XXX.3** Anzahl der Zeilen, die in die Zieltabelle eingefügt wurden.
- XXX.4** Anzahl der Zeilen in der Zieltabelle, die mit Daten aus den importierten Datensätzen aktualisiert wurden.
- XXX.5** Anzahl der Datensätze, die nicht importiert werden konnten.
- XXX.6** Anzahl der Datensätze, die erfolgreich importiert und in der Datenbank festgeschrieben wurden, einschließlich aller eingefügten, aktualisierten, übersprungenen und zurückgewiesenen Zeilen.

### Datenstruktur SQLUIMPT-IN

Diese Struktur wird verwendet, um Informationen an die IMPORT-API zu übergeben (siehe „Import - API“ auf Seite 49).

Tabelle 3. Felder in der Struktur SQLUIMPT-IN

Feldname	Datentyp	Beschreibung
SIZEOFSTRUCT	INTEGER	Größe der Struktur in Byte.
COMMITCNT	INTEGER	Die Anzahl der Datensätze, die importiert werden sollen, bevor sie in der Datenbank festgeschrieben werden. Jedes Mal, wenn <i>commitcnt</i> Datensätze importiert wurden, wird eine COMMIT-Operation ausgeführt.
RESTARTCNT	INTEGER	Die Anzahl der Datensätze, die übersprungen werden, bevor mit der Einfügung oder Aktualisierung von Datensätzen begonnen wird. Dieser Parameter sollte verwendet werden, wenn ein vorhergehender Versuch, Datensätze zu importieren, fehlschlägt, nachdem einige Datensätze in der Datenbank festgeschrieben wurden. Der angegebene Wert stellt einen Ausgangspunkt für die nächste Importoperation dar.

## Sprachsyntax

### C-Struktur

```
/* File: sqlutil.h */
/* Structure: SQLUIMPT-IN */
/* ... */
SQL_STRUCTURE sqluimpt_in
{
    sqluint32    sizeofStruct;
    sqluint32    commitcnt;
    sqluint32    restartcnt;
};
/* ... */
```

### COBOL-Struktur

```
* File: sqlutil.cbl
01 SQL-UIMPT-IN.
   05 SQL-SIZE-OF-UIMPT-IN    PIC 9(9) COMP-5 VALUE 12.
   05 SQL-COMMITCNT          PIC 9(9) COMP-5 VALUE 0.
   05 SQL-RESTARTCNT         PIC 9(9) COMP-5 VALUE 0.
*
```



## Datenstruktur SQLUIMPT-OUT

Diese Struktur wird verwendet, um Informationen von der IMPORT-API zu übergeben (siehe „Import - API“ auf Seite 49).

*Tabelle 4. Felder in der Struktur SQLUIMPT-OUT*

Feldname	Datentyp	Beschreibung
SIZEOFSTRUCT	INTEGER	Größe der Struktur in Byte.
ROWSREAD	INTEGER	Anzahl der Datensätze, die bei der Importoperation aus der Datei gelesen wurden.
ROWSSKIPPED	INTEGER	Anzahl der Datensätze, die vor Beginn der Einfügung oder Aktualisierung übersprungen wurden.
ROWSINSERTED	INTEGER	Anzahl der Zeilen, die in die Zieltabelle eingefügt wurden.
ROWSUPDATED	INTEGER	Anzahl der Zeilen in der Zieltabelle, die mit Daten aus den importierten Datensätzen aktualisiert wurden (Datensätzen, deren Primärschlüsselwert bereits in der Tabelle vorhanden ist).
ROWSREJECTED	INTEGER	Anzahl der Datensätze, die nicht importiert werden konnten.
ROWSCOMMITTED	INTEGER	Anzahl der Datensätze, die erfolgreich importiert und in der Datenbank festgeschrieben wurden.

## Sprachsyntax

### C-Struktur

```

/* File: sqlutil.h */
/* Structure: SQLUIMPT-OUT */
/* ... */
SQL_STRUCTURE sqluimpt_out
{
    sqluint32    sizeofStruct;
    sqluint32    rowsRead;
    sqluint32    rowsSkipped;
    sqluint32    rowsInserted;
    sqluint32    rowsUpdated;
    sqluint32    rowsRejected;
    sqluint32    rowsCommitted;
};
/* ... */

```

## Datenstruktur SQLUIMPT-OUT

### COBOL-Struktur

```
* File: sqlutil.cbl
01 SQL-UIMPT-OUT.
   05 SQL-SIZE-OF-UIMPT-OUT PIC 9(9) COMP-5 VALUE 28.
   05 SQL-ROWSREAD PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSSKIPPED PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSINSERTED PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSUPDATED PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSREJECTED PIC 9(9) COMP-5 VALUE 0.
   05 SQL-ROWSCOMMITTED PIC 9(9) COMP-5 VALUE 0.
*
```

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT)

Wert	Beschreibung
<b>Alle Dateiformate</b>	
compound=x	<p>x ist eine Zahl zwischen 1 und 100 (jeweils einschließlich). Verwendet keine ganzheitliche zusammengesetzte SQL-Anweisung zum Einfügen der Daten, und es werden jeweils x Anweisungen versucht.</p> <p>Wenn dieser Wert angegeben wird und das Transaktionsprotokoll nicht groß genug ist, schlägt die Importoperation fehl. Das Transaktionsprotokoll muß groß genug sein, um entweder die durch COMMITCOUNT angegebene Anzahl Zeilen oder die Anzahl der Zeilen in der Datendatei aufnehmen zu können (wenn COMMITCOUNT nicht angegeben wurde). Es empfiehlt sich daher, die Option COMMITCOUNT anzugeben, um einen Überlauf des Transaktionsprotokolls zu verhindern.</p> <p>Dieser Wert ist mit dem Modus INSERT_UPDATE, mit hierarchischen Tabellen und den folgenden Werten nicht kompatibel: usedefaults, identitymissing, identityignore, generatedmissing und generatedignore.</p>
generatedignore	Dieser Wert informiert das Dienstprogramm IMPORT darüber, daß für alle generierten Spalten in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, daß alle Werte für die generierten Spalten durch das Dienstprogramm generiert werden. Dieser Wert kann nicht zusammen mit dem Wert generatedmissing verwendet werden.
generatedmissing	Wird dieser Wert angegeben, geht das Dienstprogramm davon aus, daß die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die generierten Spalten enthält, und generiert daher für jede Zeile einen Wert. Dieser Wert kann nicht zusammen mit dem Wert generatedignore verwendet werden.

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
identityignore	Dieser Wert informiert das Dienstprogramm IMPORT darüber, daß für die Identitätsspalte in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, daß alle Identitätswerte durch das Dienstprogramm generiert werden. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dies bedeutet, daß bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Wert kann nicht zusammen mit dem Wert identitymissing verwendet werden.
identitymissing	Wird dieser Wert angegeben, geht das Dienstprogramm davon aus, daß die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Identitätsspalte enthält, und generiert daher für jede Zeile einen Wert. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dieser Wert kann nicht zusammen mit dem Wert identityignore verwendet werden.
lobsinfile	<i>lob-pfad</i> gibt den Pfad zu den Dateien an, die LOB-Werte enthalten.
no_type_id	Nur beim Import in eine einzelne untergeordnete Tabelle gültig. Wird typischerweise für den Export von Daten aus einer regulären Tabelle und dem nachfolgenden Aufruf einer Importoperation (mit diesem Wert) zur Umsetzung der Daten in eine einzelne untergeordnete Tabelle verwendet.
nodefaults	Wenn nicht ausdrücklich eine Quellenspalte für eine Zieltabellenspalte angegeben ist und die Tabellenspalte keine Nullwerte enthalten darf, werden keine Standardwerte geladen. Ohne diese Option geschieht folgendes, wenn eine Quellenspalte für eine der Zieltabellenspalten nicht ausdrücklich angegeben ist: <ul style="list-style-type: none"><li>• Wenn ein Standardwert für eine Spalte angegeben werden kann, wird der Standardwert geladen.</li><li>• Wenn die Spalte Nullwerte enthalten darf und für diese Spalte kein Standardwert angegeben werden kann, wird ein Nullwert geladen.</li><li>• Wenn die Spalte keine Nullwerte enthalten darf und kein Standardwert angegeben werden kann, wird ein Fehler zurückgegeben, und das Dienstprogramm beendet die Verarbeitung.</li></ul>

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens ein Zeilenexemplar keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> <li>• Für DEL-Dateien: „,“ wird für die Spalte angegeben.</li> <li>• Für ASC-Dateien: Der NULL-Anzeiger wird für die Spalte auf Ja gesetzt.</li> <li>• Für DEL/ASC/WSF-Dateien: Eine Zeile, die nicht genug Spalten enthält oder für die ursprüngliche Spezifikation nicht lang genug ist.</li> </ul> <p>Ohne diese Option geschieht folgendes, wenn eine Quellenspalte keine Daten für ein Zeilenexemplar enthält:</p> <ul style="list-style-type: none"> <li>• Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen.</li> <li>• Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.</li> </ul>
<b>ASCII-Dateiformate (ASC/DEL)</b>	
dateformat="x"	<p>x ist das Format des Datums in der Quellendatei.<sup>a</sup> Gültige Datums-elemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)  M - Monat (eine oder zwei Ziffern von 1 bis 12)  MM - Monat (zwei Ziffern von 1 bis 12;  dieses Element und das Element M schließen sich gegenseitig aus)  D - Tag (eine oder zwei Ziffern von 1 bis 31)  DD - Tag (zwei Ziffern von 1 bis 31;  dieses Element und das Element D schließen sich gegenseitig aus)  DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 1 zugeordnet. Einige Beispiele für Datumsformate:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
implieddecimal	Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Zum Beispiel wird der Wert 12345 in eine Spalte, für die DECIMAL(8,2) definiert ist, als 123.45 und <i>nicht</i> als 12345.00 geladen.
noeofchar	Das wahlfreie Dateiendezeichen x'1A' wird nicht als Ende der Datei erkannt. Die Verarbeitung wird wie bei einem normalen Zeichen fortgesetzt.
timeformat="x"	<p>x ist das Zeitformat in der Quelldatei.<sup>a</sup> Gültige Zeitelemente sind:</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</p> <p>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</p> <p>M - Minute (eine oder zwei Ziffern von 0 bis 59)</p> <p>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Element M schließen sich gegenseitig aus)</p> <p>S - Sekunde (eine oder zwei Ziffern von 0 bis 59)</p> <p>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</p> <p>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>TT - Meridiananzeiger (AM oder PM)</p> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate:</p> <p>"HH:MM:SS"</p> <p>"HH.MM TT"</p> <p>"SSSSS"</p>

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
timestampformat="x"	<p>x ist das Format der Zeitmarke in der Quelldatei.<sup>a</sup> Gültige Zeitmarkenelemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)</p> <p>M - Monat (eine oder zwei Ziffern von 1 bis 12)</p> <p>MM - Monat (zwei Ziffern von 1 bis 12; dieses Element und das Monatelement M schließen sich gegenseitig aus)</p> <p>D - Tag (eine oder zwei Ziffern von 1 bis 31)</p> <p>DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und das Element D schließen sich gegenseitig aus)</p> <p>DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus)</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</p> <p>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</p> <p>M - Minute (eine oder zwei Ziffern von 0 bis 59)</p> <p>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus)</p> <p>S - Sekunde (eine oder zwei Ziffern von 0 bis 59)</p> <p>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</p> <p>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>UUUUUU - Mikrosekunde (6 Ziffern von 000000 bis 999999)</p> <p>TT - Meridiananzeiger (AM oder PM)</p>

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
Fortsetzung	<p>Für nicht angegebene Elemente YYYY, M, MM, D, DD oder DDD wird der Standardwert 1 zugeordnet. Für alle anderen nicht angegebenen Elemente wird der Standardwert 0 zugeordnet. Beispiel für ein Zeitmarkenformat:</p> <pre>"YYYY/MM/DD HH:MM:SS.UUUUUU"</pre> <p>Das folgende Beispiel veranschaulicht, wie Daten in eine Tabelle namens SCHEDULE importiert werden, die benutzerdefinierte Datums- und Zeitformate enthalten:</p> <pre>db2 import from delfile2 of del   modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
<b>ASC (ASCII-Format mit universellen Zeilenbegrenzern)</b>	
nochecklengths	<p>Wenn nochecklengths angegeben wird, wird versucht, jede Zeile zu importieren, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich importiert werden, wenn die Codepage-Umsetzung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, daß die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.</p>
nullindchar=x	<p><i>x</i> ist ein einzelnes Zeichen. Ändert das Zeichen, das einen NULL-Wert angibt, in <i>x</i>. Der Standardwert für <i>x</i> ist <math>\gamma</math>.<sup>b</sup></p> <p>Bei diesem Wert wird bei EBCDIC-Datendateien zwischen Groß-/Kleinschreibung unterschieden, sofern es sich beim Zeichen nicht um ein lateinisches Zeichen handelt. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</p>
reclen=x	<p><i>x</i> ist eine ganze Zahl mit dem Maximalwert 32 767. <i>x</i> Zeichen werden für jede Zeile gelesen, und es wird kein Zeilenvorschubzeichen verwendet, um das Ende der Zeile anzugeben.</p>



Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
striptblanks	<p>Schneidet abschließende Leerzeichen ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden Leerstellen beibehalten.</p> <p>Im folgenden Beispiel bewirkt striptblanks, daß das Dienstprogramm IMPORT abschließende Leerzeichen abschneidet:</p> <pre style="margin-left: 40px;">db2 import from myfile.asc of asc    modified by striptblanks    method 1 (1 10, 12 15) messages msgs.txt    insert into staff</pre> <p>Diese Option kann nicht zusammen mit striptnulls angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Kompatibilität zu früheren Versionen unterstützt wird.</p>
striptnulls	<p>Schneidet abschließende NULL-Zeichen (0x00-Zeichen) ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden NULL-Zeichen beibehalten.</p> <p>Diese Option kann nicht zusammen mit striptblanks angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Kompatibilität zu früheren Versionen unterstützt wird.</p>
<b>DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)</b>	
chardelx	<p>x ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.<sup>bc</sup></p> <p>Auch das einfache Anführungszeichen (') kann als Zeichenfolgebegrenzer angegeben werden. Im folgenden Beispiel bewirkt chardel'', daß das Dienstprogramm IMPORT jedes einfache Anführungszeichen (') als Zeichenfolgebegrenzer wertet:</p> <pre style="margin-left: 40px;">db2 "import from myfile.del of del    modified by chardel''    method p (1, 4) insert into staff (id, years)"</pre>

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
coldelx	<p><i>x</i> ist ein einzelner Zeichenspaltenbegrenzer. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.<sup>bc</sup></p> <p>Im folgenden Beispiel bewirkt <code>coldel;</code>, daß das Dienstprogramm IMPORT jedes Semikolon (;) als Spaltenbegrenzungszeichen wertet:</p> <pre>db2 import from myfile.del of del       modified by coldel;       messages msgs.txt insert into staff</pre>
datesiso	Datumsformat. Bewirkt, daß alle Datumswerte im ISO-Format importiert werden.
decplusblank	Zeichen für Pluszeichen. Bewirkt, daß positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, Dezimalwerten ein Pluszeichen voranzustellen.
decptx	<p><i>x</i> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punktes als Dezimalzeichen verwendet.<sup>bc</sup></p> <p>Im folgenden Beispiel bewirkt <code>decpt;</code>, daß das Dienstprogramm IMPORT jedes Semikolon (;) als Dezimalzeichen wertet:</p> <pre>db2 "import from myfile.del of del       modified by chardel'       decpt; messages msgs.txt insert into staff"</pre>

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
delprioritychar	<p>Die aktuelle Standardpriorität für Begrenzer ist: Satzbegrenzer, Zeichenbegrenzer, Spaltenbegrenzer. Durch diesen Wert werden vorhandene Anwendungen geschützt, die auf die früher geltende Priorität angewiesen sind, indem die Begrenzerpriorität wie folgt geändert wird: Zeichenbegrenzer, Satzbegrenzer, Spaltenbegrenzer. Syntax:</p> <pre style="margin-left: 40px;">db2 import ... modified by delprioritychar ...</pre> <p>Beispiel anhand der folgenden DEL-Datendatei:</p> <pre style="margin-left: 40px;">"Smith, Joshua",4000,34.98&lt;zeilenbegrenzer&gt; "Vincent,&lt;zeilenbegrenzer&gt;, is a manager", ... ... 4005,44.37&lt;zeilenbegrenzer&gt;</pre> <p>Wenn der Wert delprioritychar angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite &lt;zeilenbegrenzer&gt; wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte &lt;zeilenbegrenzer&gt; als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Wert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen &lt;zeilenbegrenzer&gt;.</p>
dlldelx	<p>x ist ein DATALINK-Begrenzer (Einzelzeichen). Der Standardwert ist ein Semikolon (;). Das angegebene Zeichen wird anstelle eines Semikolons als Trennzeichen zwischen Feldern für einen DATALINK-Wert verwendet. Es ist erforderlich, weil ein DATALINK-Wert mehrere Teilwerte enthalten kann. <sup>bc</sup></p> <p><b>Anmerkung:</b> x darf nicht mit dem Zeichen übereinstimmen, das als Zeilen-, Spalten- oder Zeichenfolgebegrenzer angegeben wurde.</p>
keepblanks	<p>Erhält die führenden und abschließenden Leerzeichen in allen Feldern des Typs CHAR, VARCHAR, LONG VARCHAR oder CLOB. Ohne diese Option werden alle führenden und abschließenden Leerzeichen, die nicht innerhalb von Zeichenbegrenzern stehen, entfernt, und in die Tabelle wird für jedes leere Feld ein Nullwert eingefügt.</p>
nodoubledel	<p>Unterdrückt die Erkennung von doppelten Zeichenbegrenzern. Weitere Informationen finden Sie unter „Einschränkungen für Begrenzer“ auf Seite 20.</p>
<b>IXF-Dateiformat</b>	

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
forcein	<p>Weist das Dienstprogramm an, Daten trotz Codepage-Abweichungen zu akzeptieren und die Umsetzung zwischen Codepages zu unterdrücken.</p> <p>Zielfelder mit fester Länge werden überprüft, um sicherzustellen, daß sie groß genug für die Daten sind. Wenn der Wert <code>nochecklengths</code> angegeben ist, wird keine Überprüfung durchgeführt und nach Möglichkeit jede Zeile geladen.</p>
indexif	<p>Weist das Dienstprogramm an, alle Indizes zu löschen, die derzeit für die vorhandene Tabelle definiert sind, und aus den Indexdefinitionen in der PC/IXF-Datei neue Indizes zu erstellen. Diese Option kann nur verwendet werden, wenn der Inhalt der Tabelle ersetzt wird. Sie kann nicht mit einer Sicht verwendet werden, oder wenn eine <i>einfügespalte</i> angegeben ist.</p>
indexschema= <i>schema</i>	<p>Verwendet bei der Indexerstellung das angegebene <i>schema</i> für den Indexnamen. Wenn <i>schema</i> nicht angegeben wird (aber das Schlüsselwort <code>indexschema</code> angegeben ist), wird die Verbindungs-Benutzer-ID verwendet. Wenn das Schlüsselwort nicht angegeben ist, wird das Schema in der IXF-Datei verwendet.</p>
nochecklengths	<p>Wenn <code>nochecklengths</code> angegeben wird, wird versucht, jede Zeile zu importieren, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich importiert werden, wenn die Codepage-Umsetzung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, daß die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.</p>

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
	<p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"> <li>Das Dienstprogramm IMPORT setzt keine Warnung ab, wenn versucht wird, nicht unterstützte Dateitypen mit der Option MODIFIED BY zu verwenden. Wenn dies versucht wird, schlägt die Importoperation fehl, und es wird ein Fehlercode zurückgegeben.</li> <li><sup>a</sup> Die Zeichenfolge für das Datumsformat muß in doppelte Anführungszeichen gesetzt werden. Feldtrennzeichen dürfen die folgenden Zeichen nicht enthalten: a-z, A-Z und 0-9. Das Feldtrennzeichen darf nicht mit dem Zeichenbegrenzer oder dem Feldebegrenzer im DEL-Dateiformat identisch sein. Die Verwendung eines Feldtrennzeichens ist wahlfrei, sofern die Anfangs- und Endposition eines Elements eindeutig ist. Dies ist beispielsweise dann nicht der Fall, wenn (abhängig vom Wert) Elemente wie D, H, M oder S verwendet werden, da die Einträge in ihrer Länge variabel sind.</li> </ol> <p>Bei Zeitmarkenformaten muß darauf geachtet werden, eine Mehrdeutigkeit zwischen den Deskriptoren für Monat und Minute zu vermeiden, da beide den Buchstaben M verwenden. Ein Monatsfeld muß neben anderen Datumsfeldern stehen. Ein Minutenfeld muß neben anderen Zeitfeldern stehen. Die folgenden Zeitmarkenformate sind beispielsweise mehrdeutig:</p> <ul style="list-style-type: none"> <li>"M" (Hierdurch könnte sowohl eine Monats- als auch eine Minutenangabe dargestellt sein.)</li> <li>"M:M" (In diesem Fall ist nicht eindeutig, welches Element für den Monat und welches für die Minutenangabe steht.)</li> <li>"M:YYYY:M" (Hier werden beide Elemente M als Monat interpretiert.)</li> <li>"S:M:YYYY" (In diesem Fall steht das Element M sowohl neben einem Zeit- als auch neben einem Datumswert.)</li> </ul> <p>Im Fall einer Mehrdeutigkeit meldet das Dienstprogramm eine Fehlernachricht, und die Operation schlägt fehl.</p> <p>Die folgenden Zeitmarkenformate sind beispielsweise eindeutig:</p> <ul style="list-style-type: none"> <li>"M:YYYY" (Monat)</li> <li>"S:M" (Minute)</li> <li>"M:YYYY:S:M" (Monat...Minute)</li> <li>"M:H:YYYY:M:D" (Minute...Monat)</li> </ul> <p><b>Anmerkung:</b> Einigen Zeichen (z. B. doppelten Anführungszeichen und umgekehrten Schrägstrichen) muß ein Escape-Zeichen vorangestellt sein (beispielsweise \).</p>

## Dateitypwerte (IMPORT)

Tabelle 5. Gültige Dateitypwerte (IMPORT) (Forts.)

Wert	Beschreibung
	<p>1. <sup>b</sup> Das Zeichen muß in der Codepage der Quelldaten angegeben werden.</p> <p>Der Zeichencodepunkt (statt des Zeichensymbols) kann mit der Syntax xJJ bzw. 0xJJ angegeben werden, wobei JJ die hexadezimale Darstellung des Codepunkts ist.</p> <p>Beispiel: Das Zeichen # kann wie folgt als Spaltenbegrenzungszeichen angegeben werden</p> <pre>... modified by coldel# ... ... modified by coldel0x23 ... ... modified by coldelX23 ...</pre>
	<p>2. <sup>c</sup> Unter „Einschränkungen für Begrenzer“ auf Seite 20 werden Einschränkungen aufgelistet, die für zur Übersetzung von Begrenzern verwendbare Zeichen gelten.</p>

---

### Hinweise zu Zeichensatz und NLS

Manchmal kann es aufgrund ungleicher Codepages zu Situationen kommen, in denen die Zeichendaten möglicherweise verlängert oder gekürzt werden. Solche Situationen können bei EUC- und DBCS-Zeichensätzen für Japanisch oder traditionelles Chinesisch auftreten, die verschiedene Längencodierungen für dasselbe Zeichen verwenden können. Normalerweise wird die Länge der Eingabedaten vor dem Einlesen von Daten mit der Länge der Zielspalte verglichen. Wenn die Länge der Eingabedaten die Länge der Zielspalte überschreitet, werden Nullwerte in die Spalte eingefügt, wenn für die Spalte Nullwerte zulässig sind. Andernfalls wird die Anforderung zurückgewiesen. Wenn der Wert `nochecklengths` (siehe „Dateitypwerte (IMPORT)“ auf Seite 63) angegeben ist, wird vorab kein Vergleich durchgeführt, und die Daten werden nach Möglichkeit importiert. Wenn die Daten nach der Umsetzung zu lang sind, wird die entsprechende Zeile zurückgewiesen. Ansonsten werden die Daten importiert.

## Beispiele für IMPORT-Sitzungen

### CLP-Beispiele

#### Beispiel 1

Das folgende Beispiel zeigt, wie Sie die Daten aus myfile.ixf in die Tabelle STAFF importieren können:

```
db2 import from myfile.ixf of ixf messages msg.txt insert into staff
```

SQL3150N Der H-Satz in der PC/IXF-Datei enthält das Produkt "DB2 01.00", das Datum "19970220" und die Uhrzeit "140848".

SQL3153N Der T-Satz in der PC/IXF-Datei hat den Namen "myfile", das Qualifikationsmerkmal " " und die Quelle " ".

SQL3109N Das Dienstprogramm beginnt mit dem Laden von Daten aus der Datei "myfile".

SQL3110N Die Verarbeitung des Dienstprogramms ist abgeschlossen. Es wurden "58" Zeile(n) aus der Eingabedatei gelesen.

SQL3221W ...COMMIT WORK beginnt. Zähler für Eingabesätze: "58".

SQL3222W ...COMMIT für alle Änderungen der Datenbank wurde erfolgreich ausgeführt.

SQL314 N "58" Zeile(n) aus der Eingabedatei wurden verarbeitet. "58" Zeile(n) wurden ohne Fehler der Tabelle hinzugefügt. "0" Zeile(n) wurden zurückgewiesen.

#### Beispiel 2

Das folgende Beispiel zeigt, wie die Tabelle MOVIE TABLE aus der Eingabedatei delfile1 importiert werden kann, die Daten im DEL-Format enthält:

```
db2 import from delfile1 of del
  modified by dl|del|
  insert into movietable (actorname, description, url_making_of, url_movie)
  datalink specification (dl_url_default_prefix "http://narang"),
  (dl_url_replace_prefix "http://bomdel" dl_url_suffix ".mpeg")
```

#### Anmerkungen:

##### 1. Die Tabelle enthält vier Spalten:

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (mit LINKTYPE URL)
url_movie	DATALINK (mit LINKTYPE URL)

##### 2. Die DATALINK-Daten in der Eingabedatei verwenden den vertikalen Balken (|) als Teilfeldbegrenzer.

##### 3. Wenn ein Spaltenwert für url\_making\_of keine Präfixzeichenfolge aufweist, wird "http://narang" verwendet.

## Beispiele für IMPORT-Sitzungen

4. Jeder Spaltenwert ungleich NULL für url\_movie erhält "http://bomdel" als Präfix. Vorhandene Werte werden ersetzt.
5. An jeden Spaltenwert ungleich NULL für url\_movie wird ".mpeg" an den Pfad angefügt. Beispiel: Wenn ein Spaltenwert für url\_movie "http://server1/x/y/z" ist, wird er als "http://bomdel/x/y/z.mpeg" gespeichert; ist der Wert "/x/y/z", wird er als "http://bomdel/x/y/z.mpeg" gespeichert.

### Beispiel 3 (Importieren in eine Tabelle mit einer Identitätsspalte)

TABLE1 enthält 4 Spalten:

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 entspricht TABLE1, wobei jedoch C2 eine als GENERATED ALWAYS definierte Identitätsspalte ist.

Die Datensätze in DATAFILE1 (DEL-Format) lauten:

```
"Liszt"  
"Holst",,187.43, H  
"Grieg",100, 66.34, G  
"Satie",101, 818.23, I
```



Die Datensätze in DATAFILE2 (DEL-Format) lauten:

```
"Liszt", 74.49, A  
"Holst", 0.01, H  
"Grieg", 66.34, G  
"Satie", 818.23, I
```

Der folgende Befehl generiert Identitätswerte für die Zeilen 1 und 2, da in der Datei DATAFILE1 für diese Zeilen keine Identitätswerte zur Verfügung gestellt werden. Den Zeilen 3 und 4 wird jedoch der benutzerdefinierte Identitätswert 100 bzw. 101 zugeordnet.

```
db2 import from datafile1.del of del replace into table1
```

Um die Datei DATAFILE1 so in die Tabelle TABLE1 zu importieren, daß für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 import from datafile1.del of del method P(1, 3, 4)  
replace into table1 (c1, c3, c4)  
db2 import from datafile1.del of del modified by identityignore  
replace into table1
```

Um die Datei DATAFILE2 so in die Tabelle TABLE1 zu importieren, daß für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 import from datafile2.del of del replace into table1 (c1, c3, c4)  
db2 import from datafile2.del of del modified by identitymissing  
replace into table1
```

Wird die Datei DATAFILE1 in die Tabelle TABLE2 importiert, ohne daß einer der identitätsbezogenen Dateitypwerte verwendet wird, werden die Zeilen 1 und 2 eingefügt, die Zeilen 3 und 4 jedoch zurückgewiesen, da diese Zeilen eigene Werte, die keine Nullwerte sind, liefern und die Identitätsspalte als GENERATED ALWAYS definiert wurde.

### API-Beispiele

Siehe „API-Beispiele“ auf Seite 23.

---

### Optimieren der IMPORT-Leistung

Die Angabe von Zieltabellenspaltennamen oder einer bestimmten Importmethode verlangsamt den Import in eine ferne Datenbank.

### Rahmenbedingungen und Einschränkungen

Für das Dienstprogramm IMPORT gelten die folgenden Rahmenbedingungen:

- Dieses Dienstprogramm bietet keine Unterstützung für Kurznamen.
- Ist die vorhandene Tabelle eine übergeordnete Tabelle mit einem Primärschlüssel, auf den ein Fremdschlüssel einer abhängigen Tabelle verweist, können keine Daten ersetzt werden. Es ist nur möglich, die neuen Daten anzufügen.
- Es ist nicht möglich, eine IMPORT REPLACE-Operation in eine zugrundeliegende Tabelle einer Übersichtstabelle auszuführen, die im Modus sofortiger Aktualisierung definiert wurde.
- Sie können keine Daten in eine Systemtabelle, eine Übersichtstabelle oder in eine Tabelle mit Spalten strukturierten Typs importieren.
- Sie können keine Daten in deklarierte temporäre Tabellen importieren.
- Mit dem Dienstprogramm IMPORT können keine Sichten erstellt werden.
- Referentielle Integritätsbedingungen und Fremdschlüsseldefinitionen werden beim Erstellen von Tabellen aus PC/IXF-Dateien nicht beibehalten. (Primärschlüsseldefinitionen *werden* beibehalten, wenn die Daten zuvor mit SELECT \* exportiert wurden.)
- Da vom Dienstprogramm IMPORT eigene SQL-Anweisungen generiert werden, kann die maximale Anweisungsgröße von 64 KB in einigen Fällen überschritten werden.

Für das Dienstprogramm IMPORT gilt die folgende Einschränkung:

Überschreitet der Umfang der Ausgabenachrichten, die durch eine Importoperation für eine ferne Datenbank generiert wurden, eine Größe von 60 KB, behält das Dienstprogramm die ersten 30 KB und die letzten 30 KB bei.

---

### Fehlerbehebung

Bei DB2-Operationen wie dem Exportieren, Importieren, Laden, Binden oder Wiederherstellen von Daten können Sie angeben, daß Nachrichtendateien erstellt werden, die Fehlermeldungen, Warnungen und informative Nachrichten zu diesen Operationen enthalten. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an.

Diese Nachrichtendateien sind gewöhnliche ASCII-Dateien. Jede Nachricht in einer Nachrichtendatei beginnt in einer neuen Zeile und enthält Informationen, die von der DB2-Funktion zum Abrufen von Nachrichten bereitgestellt werden. Sie können sie mit dem bei Ihrem Betriebssystem üblichen Verfahren drucken oder mit einem beliebigen ASCII-Editor anzeigen.

---

## Kapitel 3. LOAD

In diesem Kapitel wird das DB2 UDB-Dienstprogramm LOAD beschrieben, mit dem Daten aus Dateien, benannten Pipes oder Einheiten in eine DB2-Tabelle versetzt werden. Diese Datenquellen können sich entweder auf dem Knoten befinden, auf dem sich auch die Datenbank befindet, oder auf einem Client mit Fernverbindung. Die zu ladende Tabelle muß vorhanden sein. Wenn die Tabelle, die die neuen Daten empfängt, bereits Daten enthält, können Sie diese ersetzen oder die neuen Daten an die vorhandenen Daten anfügen.

In diesem Kapitel werden die folgenden Themen behandelt:

- „LOAD - Übersicht“ auf Seite 80
- „Parallelität und Laden“ auf Seite 88
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD“ auf Seite 89
- „Verwenden von LOAD“ auf Seite 90
- „Verwenden von LOAD mit Identitätsspalten“ auf Seite 91
- „Verwenden von LOAD mit generierten Spalten“ auf Seite 93
- „Überprüfung auf ungültige Integritätsbedingungen“ auf Seite 96
- „Neustart einer unterbrochenen Ladeoperation“ auf Seite 98
- „Verwenden der Datei mit den Angaben zur Speicherposition der Ladekopie“ auf Seite 98
- „LOAD - Befehl“ auf Seite 101
- „LOAD QUERY - Befehl“ auf Seite 116
- „LOAD - API“ auf Seite 118
- „Datenstruktur SQLLOAD-IN“ auf Seite 128
- „Datenstruktur SQLLOAD-OUT“ auf Seite 133
- „db2LoadQuery - API Load Query“ auf Seite 135
- „Dateitypwerte (LOAD)“ auf Seite 140
- „Ausnahmetabelle“ auf Seite 157
- „Speicherauszugsdatei“ auf Seite 158
- „Temporäre Dateien (LOAD)“ auf Seite 158
- „Protokollsätze zum Dienstprogramm LOAD“ auf Seite 159
- „Zeichensatz und NLS“ auf Seite 160
- „Beispiele für LOAD-Sitzungen“ auf Seite 160

- „Statusangaben für anstehende Aktionen nach einer Ladeoperation“ auf Seite 173
- „Optimieren der Leistung von LOAD“ auf Seite 174
- „Rahmenbedingungen und Einschränkungen“ auf Seite 180
- „Fehlerbehebung“ auf Seite 180.

Weitere Informationen zum Laden von DB2 Data Links Manager-Daten finden Sie unter „Versetzen von DB2 Data Links Manager-Daten mit LOAD“ auf Seite 207.

---

## LOAD - Übersicht

Mit dem Dienstprogramm LOAD können große Datenmengen effizient in neu erstellte Tabellen oder in Tabellen, die bereits Daten enthalten, versetzt werden. Es können alle Datentypen verarbeitet werden, einschließlich großer Objekte (LOBs) und benutzerdefinierter Typen (UDTs). Das Dienstprogramm LOAD arbeitet schneller als das Dienstprogramm IMPORT, da es formatierte Seiten direkt in die Datenbank schreibt, während das Dienstprogramm IMPORT die Daten mit SQL-Anweisungen INSERT einfügt. Das Dienstprogramm LOAD startet keine Auslöser und führt außer der Prüfung auf eindeutige Indizes keine Prüfungen auf referentielle Integrität oder Integritätsbedingungen in Tabellen aus. Einen ausführlichen Vergleich der Dienstprogramme LOAD und IMPORT finden Sie in „Anhang B. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD“ auf Seite 235.

Der LOAD-Prozeß umfaßt drei Einzelphasen (siehe Abb. 1 auf Seite 81):

- **LOAD.** Dabei werden die Daten in die Tabelle geschrieben.  
Während der LOAD-Phase werden die Daten in die Tabelle geladen. Bei Bedarf können Indexschlüssel und Tabellenstatistiken gesammelt werden. Sicherungspunkte oder Konsistenzzustände werden in Intervallen eingerichtet, die Sie mit dem Parameter SAVECOUNT des Befehls LOAD angegeben haben. Es werden Nachrichten generiert, durch die Sie erfahren, wie viele Eingabezeilen zum Zeitpunkt des Sicherungspunkts erfolgreich geladen wurden. Für mit FILE LINK CONTROL definierte DATALINK-Spalten werden Verbindungsoperationen für Spaltenwerte ungleich NULL ausgeführt. Wenn ein Fehler auftritt, können Sie die Ladeoperation erneut starten. Mit der Option RESTART wird die Ladeoperation automatisch vom letzten erfolgreichen Konsistenzzustand ausgehend erneut gestartet. Mit der Option TERMINATE wird die fehlgeschlagene Ladeoperation rückgängig gemacht.

## Die drei Phasen einer LOAD-Operation



Abbildung 1. Die drei Phasen des LOAD-Prozesses: LOAD, BUILD und DELETE. Zugeordnete Tabellenbereiche befinden sich vom Beginn der LOAD-Phase bis zum Ende der BUILD-Phase im Status „Laden anstehend“. Vom Ende der BUILD-Phase bis zum Ende der DELETE-Phase befinden sie sich im Status „Löschen anstehend“.

- **BUILD.** Dabei werden die Indizes erstellt.  
Während der BUILD-Phase werden auf der Grundlage der während der LOAD-Phase gesammelten Indexschlüssel Indizes erstellt. Die Indexschlüssel werden während der LOAD-Phase sortiert, und Indexstatistiken werden gesammelt (wenn bei der Option INDEXES der Wert STATISTICS YES angegeben wurde). Die erfaßten Statistikdaten sind denen des Befehls RUNSTATS ähnlich (siehe Handbuch *Command Reference*). Wenn in der BUILD-Phase ein Fehler auftritt, können Sie die Ladeoperation mit der Option RESTART automatisch vom letzten erfolgreichen Konsistenzzustand ausgehend erneut starten.
- **DELETE.** Dabei werden die Zeilen, die zu Verstößen gegen eindeutige Schlüssel oder einer DATALINK-Verletzung geführt haben, aus der Tabelle gelöscht. Verstöße gegen eindeutige Schlüssel werden in die Ausnahmetabelle eingetragen, sofern eine solche Tabelle angegeben wurde (siehe „Ausnahmetabelle“ auf Seite 157). Nachrichten zu zurückgewiesenen Zeilen werden in die Nachrichtendatei aufgenommen. Führen Sie im Anschluß an den LOAD-Prozeß folgende Aktionen aus: Prüfen Sie die Nachrichten, korrigieren Sie eventuelle Fehler, und fügen Sie die korrigierten Zeilen in die Tabelle ein.

Versuchen Sie nicht, temporäre Dateien, die vom Dienstprogramm LOAD erstellt wurden, zu löschen oder zu ändern. Manche temporäre Dateien sind für die DELETE-Phase kritisch. Wenn in der DELETE-Phase ein Fehler auftritt, wird die Ladeoperation mit der Option RESTART automatisch vom letzten erfolgreichen Konsistenzzustand ausgehend erneut gestartet.

**Anmerkung:** Jedes Löschereignis wird protokolliert. Wenn eine größere Zahl von Datensätzen die Eindeutigkeitsbedingung verletzt, kommt es möglicherweise während der DELETE-Phase zu einem Protokollüberlauf.

## LOAD - Übersicht

Folgende Informationen werden benötigt, wenn Daten geladen werden:

- Der Pfad und der Name der Eingabedatei, benannten Pipe oder Einheit.
- Der Name oder Aliasname der Zieltabelle.
- Das Format der Daten in der Eingabedatei. Mögliche Formate sind DEL, ASC oder PC/IXF. Siehe „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.
- Angabe, ob die Eingabedaten an die Tabelle angefügt werden oder die vorhandenen Daten in der Tabelle ersetzen sollen.
- Der Name einer Nachrichtendatei, wenn das Dienstprogramm über die Anwendungsprogrammierschnittstelle (API) **sqluload** aufgerufen wird.

Sie können auch folgendes angeben:

- Die zu ladenden Daten befinden sich auf dem Client - sofern das Dienstprogramm LOAD von einem Client mit Fernverbindung aufgerufen wird.
- Die Methode, die zum Laden der Daten verwendet werden soll: Anfangs- und Endposition von Spalten, Spaltenname oder Spaltenpositionen.
- Die Häufigkeit, mit der das Dienstprogramm Konsistenzzustände herstellen soll. Verwenden Sie zur Angabe dieses Wertes den Parameter SAVECOUNT. Wenn dieser Parameter angegeben wird, beginnt eine LOAD RESTART-Operation vom letzten Konsistenzzustand ausgehend und nicht vom Anfang.
- Die Namen der Tabellenspalten, in die die Daten eingefügt werden sollen.
- Die Pfade und die Namen der Eingabedateien, in denen LOBs gespeichert sind. Der Wert `lobsinfile` teilt dem Dienstprogramm LOAD mit, daß alle LOB-Daten aus Dateien geladen werden (siehe „Dateitypwerte (LOAD)“ auf Seite 140).
- Angabe, ob Spaltenwerte, die geladen werden, ein impliziertes Dezimalzeichen haben oder nicht. Der Wert `implieddecimal` teilt dem Dienstprogramm LOAD mit, daß Dezimalzeichen auf die Daten angewendet werden sollen, wenn sie in die Tabelle geschrieben werden (siehe „Dateitypwerte (LOAD)“ auf Seite 140). Zum Beispiel wird der Wert 12345 in eine Spalte, für die DECIMAL(8,2) definiert ist, als 123.45 und nicht als 12345.00 geladen.
- Angabe, ob das Dienstprogramm die Menge des freien Speicherbereichs nach dem Laden einer Tabelle ändern soll. Zusätzlicher freier Speicherbereich ermöglicht ein Anwachsen der Tabelle durch INSERT und UPDATE nach Abschluß einer Ladeoperation. Verkleinerter freier Speicherbereich hält zusammengehörige Zeilen näher zusammen und kann zu einem verbesserten Durchsatz führen.

- Der Wert `totalfreespace` ermöglicht das Anfügen leerer Datenseiten an das Ende der geladenen Tabelle. Die mit diesem Parameter angegebene Zahl ist der Prozentsatz der Gesamtseiten in der Tabelle, der als freier Speicherbereich an das Tabellenende angefügt werden soll. Wenn Sie beispielsweise für diesen Parameter 20 angeben und die Tabelle 100 Datenseiten umfaßt, werden 20 zusätzliche leere Seiten angefügt. Die Gesamtzahl der Datenseiten in der Tabelle ist somit 120.
- Mit dem Wert `pagefreespace` können Sie die Menge freien Speicherbereichs steuern, der auf jeder geladenen Datenseite zulässig ist. Die mit diesem Parameter angegebene Zahl ist der Prozentsatz jeder Datenseite, der als Speicherbereich freigelassen werden soll. Die erste Zeile auf einer Seite wird ohne Einschränkung hinzugefügt. Bei sehr großen Zeilen und einer großen Zahl für diesen Parameter ist daher eventuell weniger freier Speicherbereich auf jeder Seite übrig, als durch den für diesen Parameter verwendeten Wert angegeben wird.
- Mit dem Wert `indexfreespace` können Sie die Menge freien Speicherbereichs steuern, der auf jeder geladenen Indexseite zulässig ist. Die mit diesem Parameter angegebene Zahl ist der Prozentsatz jeder Indexseite, der als Speicherbereich freigelassen werden soll. Der erste Indexeintrag auf einer Seite wird ohne Einschränkung hinzugefügt. Zusätzliche Indexeinträge werden auf der Indexseite plaziert, vorausgesetzt, der Schwellenwert für freien Speicherbereich (in Prozent) kann beibehalten werden. Der Standardwert ist der zum Zeitpunkt der Indexerstellung (CREATE INDEX) verwendete Wert. Der Wert `indexfreespace` hat Vorrang vor dem in der Anweisung CREATE INDEX angegebenen Wert für `PCTFREE`.

Wenn Sie den Wert `pagefreespace` angeben und einen Index für die Tabelle verwenden, können Sie auch `indexfreespace` angeben. Beim Festlegen, wieviel Speicherbereich jeweils freigelassen werden soll, ist zu beachten, daß jede in die Tabelle eingefügte Zeile wahrscheinlich größer ist als der zugehörige Schlüssel, der in den Index einzufügen ist. Außerdem können die Seiten der Tabellenbereiche für die Tabelle und für den Index unterschiedlich groß sein.

- Angabe, ob während des LOAD-Prozesses Statistikdaten gesammelt werden. Diese Option wird nur unterstützt, wenn die Ladeoperation im Modus REPLACE ausgeführt wird.

Wenn Daten an eine Tabelle angefügt werden, werden keine Statistikdaten gesammelt. Rufen Sie zum Sammeln von aktuellen Statistikdaten zu einer angefügten Tabelle nach Beendigung des Ladeprozesses das Dienstprogramm RUNSTATS auf.

## LOAD - Übersicht

Wenn Statistikdaten zu einer Tabelle mit einem eindeutigen Index gesammelt und während der DELETE-Phase doppelte Schlüssel gelöscht werden, erfolgt keine Aktualisierung der Statistikdaten zur Berücksichtigung der gelöschten Datensätze. Wenn Sie eine signifikante Anzahl von doppelten Datensätzen erwarten, dürfen Sie während der Ladeoperation keine Statistikdaten sammeln. Rufen Sie statt dessen nach Beendigung des Ladeprozesses das Dienstprogramm RUNSTATS auf.

- Angabe, ob eine Kopie der vorgenommenen Änderungen erhalten bleiben soll. Diese wird benötigt, um eine aktualisierende Wiederherstellung der Datenbank zu ermöglichen. Diese Option wird nicht unterstützt, wenn für die Datenbank die aktualisierende Wiederherstellung über Protokolle inaktiviert ist, d. h. wenn die Datenbankkonfigurationsparameter *logretain* und *userexit* inaktiviert sind. Wenn keine Kopie erstellt wird und die aktualisierende Wiederherstellung über Protokolle aktiviert ist, verbleibt der Tabellenbereich nach Beendigung der Ladeoperation im Status „Sicherung anstehend“ (siehe „Statusangaben für anstehende Aktionen nach einer Ladeoperation“ auf Seite 173).

Für vollständig wiederherstellbare Datenbanken ist eine Protokollierung erforderlich. Das Dienstprogramm LOAD schließt die mit dem Laden von Daten verbundene Protokollierung fast völlig aus. Anstelle der Protokollierung haben Sie die Möglichkeit, eine Kopie des geladenen Tabellenteils zu erstellen. Weitere Informationen dazu, wie DB2 die Ladekopien verwaltet, finden Sie in „Verwenden der Datei mit den Angaben zur Speicherposition der Ladekopie“ auf Seite 98. Bei einer Datenbankumgebung, die Datenbankwiederherstellung nach einem Fehler zuläßt, haben Sie folgende Möglichkeiten:

- Explizites Anfordern der Erstellung einer Kopie des geladenen Tabellenteils
- Nach Beendigung der Ladeoperation Erstellen einer Sicherung der Tabellenbereiche, in denen sich die Tabelle befindet

Wenn Sie eine Tabelle laden, die bereits Daten enthält, und die Datenbank nicht wiederherstellbar ist, müssen Sie sicherstellen, daß Sie über eine Sicherungskopie der Datenbank oder über die Tabellenbereiche für die geladene Tabelle verfügen, bevor Sie das Dienstprogramm LOAD aufrufen, damit Sie Fehler beheben können.

Wenn Sie eine Folge von mehreren Ladeoperationen für eine wiederherstellbare Datenbank ausführen möchten, erfolgt die Operationsfolge schneller, wenn Sie jede Ladeoperation als nicht wiederherstellbar definieren und am Ende der Ladefolge eine Sicherung erstellen, als wenn Sie jede der Ladeoperationen mit der Option COPY YES aufrufen.



Sie können die Option NONRECOVERABLE verwenden, um festzulegen, daß eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und daß es nicht möglich sein wird, sie durch eine nachfolgende aktualisierende Wiederherstellung wiederherzustellen. Das Dienstprogramm zur aktualisierenden Wiederherstellung überspringt die Transaktion und kennzeichnet die Tabelle, in die die Daten geladen wurden, als „ungültig“. Vom Dienstprogramm werden auch alle nachfolgenden Transaktionen für diese Tabelle ignoriert. Nach Beendigung der aktualisierenden Wiederherstellung kann eine solche Tabelle nur gelöscht werden (siehe Abb. 2). Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status „Sicherung anstehend“ gesetzt, und während der Ladeoperation muß keine Kopie der geladenen Daten erstellt werden.

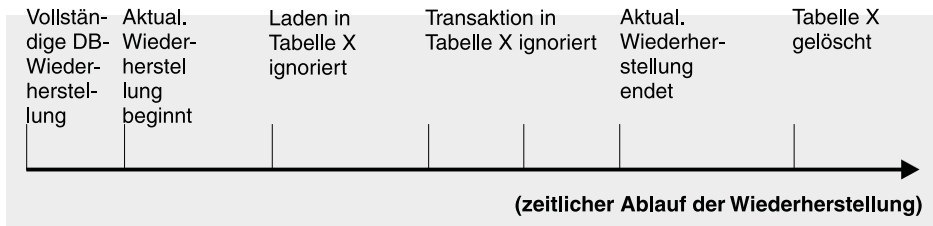


Abbildung 2. Nicht wiederherstellbare Verarbeitung bei einer aktualisierenden Wiederherstellung

Weitere Informationen finden Sie im Kapitel zur Datenbankwiederherstellung im Handbuch *Systemverwaltung*.

- Den vollständig qualifizierten Pfad, der beim Erstellen von temporären Dateien während einer Ladeoperation verwendet werden soll. Der Name wird mit dem Parameter `TEMPFILES PATH` des Befehls `LOAD` angegeben. Der Standardwert ist der Datenbankpfad. Der Pfad befindet sich auf der Server-Maschine, und es wird darauf ausschließlich vom DB2-Exemplar zugegriffen. Daher muß jede Angabe eines Pfadnamens in diesem Parameter die Verzeichnisstruktur des Servers und nicht des Clients berücksichtigen. Außerdem benötigt der DB2-Exemplareigner Lese- und Schreibzugriff für den Pfad. Das gilt auch, wenn Sie der Exemplareigner sind. Wenn Sie nicht der Exemplareigner sind, müssen Sie eine Speicherposition angeben, für die der Exemplareigner Schreibzugriff besitzt. Weitere Informationen zu temporären Dateien finden Sie unter „Temporäre Dateien (LOAD)“ auf Seite 158.

## Änderungen bei Syntax und Funktionsweise des Dienstprogramms LOAD

### Änderungen der früheren Funktionsweise des Dienstprogramms LOAD (eingeführt in Version 6 und 7)

Die Versionen 6 und 7 des Dienstprogramms LOAD bieten eine uneingeschränkte Abwärtskompatibilität zu früheren Releases, d. h. sie akzeptieren die Syntax aus früheren Releases und funktionieren ordnungsgemäß. Im folgenden werden die Syntaxänderungen und die Änderungen der Funktionsweise des Dienstprogramms LOAD zusammengefaßt, die in Version 6 eingeführt wurden:

- Beim erneuten Starten des Dienstprogramms LOAD wird nicht mehr ein Wert RESTARTCOUNT verwendet. Der Parameter ist jetzt reserviert. Wenn eine zuvor unterbrochene Ladeoperation erneut gestartet wird, wird die Ladeoperation automatisch ausgehend vom letzten Konsistenzzustand in der LOAD-, BUILD- oder DELETE-Phase fortgesetzt.
- Das Sortieren der Indexschlüssel während der Indexerstellung nutzt einen neuen Sortieralgorithmus, der von der Datenbanksteuerkomponente der Version 6 von DB2 UDB verwendet wird. Die für den Sortiervorgang bestimmte Speicherkapazität wird durch den Wert des Datenbankkonfigurationsparameters für den Sortierspeicher (*sortheap*) sowie den Wert des Konfigurationsparameters für den Zwischenspeicher für Sortierlisten (*sheapthres*) des Datenbankmanagers gesteuert. Dies bedeutet, daß die früheren Optionen des Dienstprogramms LOAD für SORT BUFFER und die über die Klausel USING angegebenen temporären Sortierverzeichnisse nun obsolet sind. Wenn diese Optionen in Version 6 angegeben werden, wird eine Informationsnachricht mit einer Warnung zurückgegeben, die Ladeoperation jedoch normal fortgesetzt.

Sortierüberläufe, die während der Ladeoperation bei der Indexerstellung auftreten, werden nun in einem temporären Tabellenbereich ausgeführt. Ein Überlauf der Sortiervorgänge findet nicht direkt auf die Platte statt, sondern vielmehr in den Pufferpool, der dem temporären Tabellenbereich zugeordnet ist. Die für die Indexerstellung benötigte Zeit kann verbessert werden, wenn dem temporären Tabellenbereich ein großer Pufferpool zugeordnet wird. Um denselben Typ der E/A-Parallelität zu erzielen, der bei LOAD SORT-Operationen vor Version 6 (durch Angabe mehrerer temporärer Sortierverzeichnisse) verfügbar war, empfiehlt es sich, temporäre Tabellenbereiche mit mehreren Behältern zu deklarieren, die sich jeweils auf unterschiedlichen Platteneinheiten befinden. Es empfiehlt sich außerdem, temporäre Tabellenbereiche als SMS (System Managed Space) zu deklarieren, damit diese zur Aufnahme von umfangreichen Datenmengen vergrößert werden können, ohne jedoch Plattenressourcen zu belegen, wenn diese nicht verwendet werden.

## Änderungen bei Syntax und Funktionsweise des Dienstprogramms LOAD

- Die Option REMOTE FILE wurde umbenannt. (Das Dienstprogramm akzeptiert jedoch weiterhin den Parameter REMOTE FILE bei der Angabe des Pfads von temporären Dateien.) Dies ist eine rein syntaktische Änderung, die Bedeutung und Zweck dieses Parameters besser wiedergeben soll. Der Parameter TEMPFILES PATH verweist auf ein Verzeichnis anstelle einer Datei (siehe „Temporäre Dateien (LOAD)“ auf Seite 158).
- Das Dienstprogramm LOAD unterstützt jetzt mehrere Indexierungsmodi: „Alle Indizes wiederherstellen“, „Indizes mit neuen Daten erweitern“, „Indexaktualisierung verzögern“ (nach Abschluß einer Ladeoperation) sowie „Automatisch auswählen“. Der letzte Modus wählt bei Laufzeit zwischen der Wiederherstellung aller Indizes und der Erweiterung der Indizes mit neuen Daten aus. Der Modus „Alle Indizes wiederherstellen“ spiegelt die Funktionsweise von Releases vor Version 6 wider. Standardmodus für Version 6 ist der Modus „Automatisch auswählen“.
- In Version 6 kann eine Ladeoperation mit der Option TERMINATE zurückgesetzt werden. Zuvor wurde durch diese Option ein Tabellenbereich in den Status „Wiederherstellung anstehend“ gesetzt. Bitte beachten Sie jedoch, daß nach einer fehlgeschlagenen Operation LOAD REPLACE die Tabellendaten durch eine Anforderung TERMINATE *nicht* wiederhergestellt werden.

Version 7 des Dienstprogramms LOAD kann Daten laden, die sich auf einem Client mit Fernverbindung, in vollständig qualifizierten Pipes oder in benannten Pipes befinden. (Separate Dateien mit LOB-Werten sollten sich auf dem Server befinden, wenn der Dateitypwert `lobsinfile` angegeben wird.)

### Parallelität und Laden

Das Dienstprogramm LOAD kann die Vorteile einer Hardwarekonfiguration nutzen, in der mehrere Prozessoren oder mehrere Speichereinheiten verwendet werden, wie z. B. in einer symmetrischen Mehrprozessorumgebung (SMP - Symmetric Multiprocessor). Es gibt mehrere Möglichkeiten, wie eine parallele Verarbeitung umfangreicher Datenmengen mit dem Dienstprogramm LOAD durchgeführt werden kann. Eine Möglichkeit besteht in der Verwendung mehrerer Speichereinheiten, die eine E/A-Parallelität während der Ladeoperation ermöglicht (siehe Abb. 3). Eine weitere Möglichkeit ist der Einsatz mehrerer Prozessoren in einer SMP-Umgebung, der eine partitionsinterne Parallelität ermöglicht (siehe Abb. 4). Beide Möglichkeiten können kombiniert verwendet werden, um ein noch schnelleres Laden der Daten zu erreichen. Weitere Informationen finden Sie unter „Optimieren der Leistung von LOAD“ auf Seite 174.

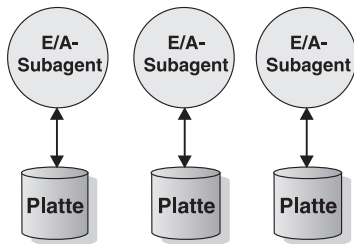


Abbildung 3. E/A-Parallelität beim Laden von Daten

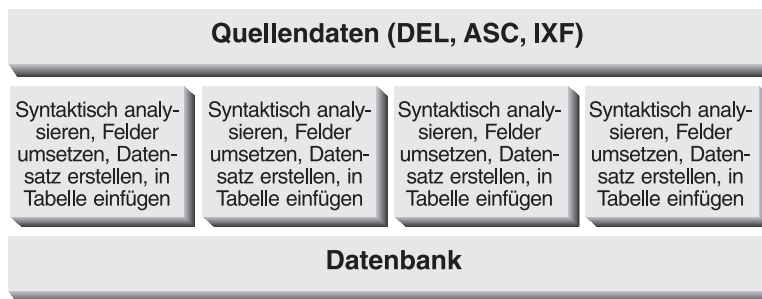


Abbildung 4. Partitionsinterne Parallelität beim Laden von Daten

### Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie Operationen von Dienstprogrammen und zur Datenbankmanagerpflege auf höherer Ebene zu gruppieren. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie die entsprechende Berechtigung besitzen, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen eine der folgenden Berechtigungen, damit Sie Daten in eine Tabelle laden können:

- Berechtigung SYSADM
- Berechtigung DBADM
- Berechtigung LOAD für die Datenbank und
  - Zugriffsrecht INSERT für die Tabelle, wenn das Dienstprogramm LOAD im Modus INSERT, im Modus TERMINATE (zur Beendigung einer vorherigen LOAD INSERT-Operation) oder im Modus RESTART (zum erneuten Starten einer vorherigen LOAD INSERT-Operation) aufgerufen wird.
  - Zugriffsrecht INSERT und DELETE für die Tabelle, wenn das Dienstprogramm LOAD im Modus REPLACE, im Modus TERMINATE (zur Beendigung einer vorherigen LOAD REPLACE-Operation) oder im Modus RESTART (zum erneuten Starten einer vorherigen LOAD REPLACE-Operation) aufgerufen wird.
  - Zugriffsrecht INSERT für die Ausnahmetabelle, wenn eine solche Tabelle im Rahmen der Ladeoperation verwendet wird.

Da der Exemplareigner Eigner aller LOAD-Prozesse (und im allgemeinen auch aller DB2-Server-Prozesse) ist und alle diese Prozesse die ID des Exemplareigners verwenden, um auf benötigte Dateien zuzugreifen, muß der Exemplareigner über einen Lesezugriff auf Eingabedatendateien verfügen. Diese Eingabedatendateien müssen für den Exemplareigner lesbar sein, unabhängig davon, wer den Befehl aufruft.

Wenn DB2 für Windows NT im Betriebssystem Windows NT als Dienst definiert wurde, muß der Dienst ein Benutzerkonto mit den erforderlichen Berechtigungen zum Lesen und Schreiben von Dateien besitzen, um LAN-Ressourcen (Laufwerke, Verzeichnisse und Dateien) verwenden zu können.

### Verwenden von LOAD

#### Vor der Verwendung von LOAD

Bevor Sie das Dienstprogramm LOAD ausführen, müssen Sie mit der Datenbank, in die die Daten geladen werden, verbunden bzw. in der Lage sein, implizit eine Verbindung zur Datenbank herzustellen. Da das Dienstprogramm die Anweisung COMMIT absetzt, sollten Sie vor dem Aufrufen von LOAD alle Transaktionen beenden und alle Sperren aufheben, indem Sie die Anweisung COMMIT oder ROLLBACK ausführen.

Da Daten in der gleichen Reihenfolge geladen werden, wie sie in der Eingabedatei vorliegen, müssen sie, wenn eine bestimmte Reihenfolge gewünscht wird, vor dem Durchführen der Ladeoperation sortiert werden.

Wenn Cluster erforderlich sind, müssen die Daten vor dem Laden nach dem Clusterindex sortiert werden.

#### Aufrufen von LOAD

Das Dienstprogramm LOAD kann wie folgt aufgerufen werden:

- Über den Befehlszeilenprozessor (CLP).

Es folgt ein Beispiel für den Befehl LOAD, der über den CLP abgesetzt wird:

```
db2 load from stafftab.ixf of ixf messages staff.msgs
      insert into userid.staff copy yes use tsm data buffer 4000
```

In diesem Beispiel gilt:

- Alle Warnungen oder Fehlermeldungen werden in die Datei `staff.msgs` gestellt.
- Eine Kopie der vorgenommenen Änderungen wird in Tivoli Storage Manager (TSM, vormals ADSM) gespeichert. Weitere Informationen zu TSM finden Sie im Abschnitt zu Tivoli Storage Manager im Kapitel zur Datenbankwiederherstellung im Handbuch *Systemverwaltung*.
- Bei der Ladeoperation sind viertausend Seiten Pufferspeicherbereich zu verwenden.

Es folgt ein weiteres Beispiel für den Befehl LOAD, der über den CLP abgesetzt wird:

```
db2 load from stafftab.ixf of ixf messages staff.msgs
      tempfiles path /u/myuser replace into staff
```

In diesem Beispiel gilt:

- Die Tabellendaten werden ersetzt.
- Der Parameter `TEMPFILES PATH` wird verwendet, um `/u/myuser` als Server-Pfad anzugeben, in den temporäre Dateien geschrieben werden.

**Anmerkung:** In diesen Beispielen werden relative Pfadnamen für die LOAD-Eingabedatei verwendet. Relative Pfadnamen sind nur für Aufrufe eines Clients auf demselben Knoten wie die Datenbank zulässig. Es empfiehlt sich die Verwendung von vollständig qualifizierten Pfadnamen.

- Über das Notizbuch **Laden** in der Steuerzentrale. So öffnen Sie das Notizbuch **Laden**:
  1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie den Ordner **Tabellen** sehen.
  2. Klicken Sie den Ordner **Tabellen** an. Alle vorhandenen Tabellen werden auf der rechten Seite des Fensters, im Inhaltsteilfenster, angezeigt.
  3. Klicken Sie im Inhaltsteilfenster die gewünschte Tabelle mit Maustaste 2 an, und wählen Sie **Laden** im Kontextmenü aus. Das Notizbuch **Laden** wird geöffnet.

Basisinformationen zur Steuerzentrale finden Sie im Handbuch *Systemverwaltung*. Zusatzinformationen bietet die Online-Hilfefunktion innerhalb der Steuerzentrale.

- Über die Anwendungsprogrammierschnittstelle (API) **sqluload**. Informationen zu dieser API finden Sie unter „LOAD - API“ auf Seite 118. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, finden Sie im Handbuch *Application Building Guide*.

---

### Verwenden von LOAD mit Identitätsspalten

Mit dem Dienstprogramm LOAD können Sie Daten in eine Tabelle laden, die eine Identitätsspalte enthält. Sofern keine identitätsbezogenen Dateitypwerte verwendet werden, liegen der Ausführung des Dienstprogramms die folgenden Regeln zugrunde:

- Ist die Identitätsspalte als GENERATED ALWAYS definiert, wird für eine Tabellenzeile immer dann ein Identitätswert generiert, wenn die entsprechende Zeile in der Eingabedatei keinen Wert für die Identitätsspalte enthält oder explizit einen Nullwert vorgibt. Ist für die Identitätsspalte ein Wert angegeben, der kein Nullwert ist, wird die Zeile zurückgewiesen (SQL3550W).
- Ist die Identitätsspalte als GENERATED BY DEFAULT definiert, verwendet das Dienstprogramm LOAD benutzerdefinierte Werte, sofern diese vorhanden sind. Wenn diese Daten fehlen oder explizit Nullwerte sind, wird ein Wert generiert.

Das Dienstprogramm LOAD führt keine zusätzliche Gültigkeitsprüfung der benutzerdefinierten Identitätswerte durch, die über die übliche Gültigkeitsprüfung für Werte mit dem Datentyp der Identitätsspalte (also SMALLINT, INT, BIGINT oder DECIMAL) hinausgeht. Falls Werte doppelt vorhanden sind, wird keine entsprechende Meldung ausgegeben.

## Verwenden von LOAD mit Identitätsspalten

Das Dienstprogramm LOAD unterstützt drei (sich gegenseitig ausschließende) Dateitypwerte, die die Verwendung dieses Dienstprogramms für Tabellen mit einer Identitätsspalte vereinfachen:

- Der Wert `identitymissing` vereinfacht das Laden einer Tabelle mit einer Identitätsspalte, wenn die Eingabedatendatei für die Identitätsspalte keine Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 varchar(30),
                    c2 int generated by default as identity,
                    c3 decimal(7,2),
                    c4 char(1))
```

Ein Benutzer möchte Daten aus einer Datei (`load.del`) in die Tabelle `TABLE1` laden. Diese Datei wurde aus einer Tabelle exportiert, die keine Identitätsspalte enthält. Es folgt ein Beispiel für eine solche Datei.

```
Robert, 45.2, J
Mike, 76.9, K
Leo, 23.4, I
```

Eine Methode für das Laden dieser Datei wäre das explizite Auflisten der zu ladenden Spalten durch den folgenden Befehl LOAD:

```
db2 load from load.del of del replace into table1 (c1, c3, c4)
```

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Laden der Datei ist die folgende Verwendung des Dateityperts `identitymissing`:

```
db2 load from load.del of del modified by identitymissing replace into table1
```

- Der Wert `identityignore` stellt in gewisser Hinsicht das Gegenteil des Werts `identitymissing` dar: Er weist das Dienstprogramm LOAD an, daß die in der Eingabedatendatei vorhandenen Werte für die Identitätsspalte ignoriert werden sollen und daß für jede Zeile ein Identitätswert generiert werden soll. Beispiel: Ein Benutzer möchte die wie oben definierte Tabelle `TABLE1` aus einer Datendatei (`load.del`) mit den folgenden Daten laden:

```
Robert, 1, 45.2, J
Mike, 2, 76.9, K
Leo, 3, 23.4, I
```

Wenn die benutzerdefinierten Werte 1, 2 und 3 nicht für die Identitätsspalte verwendet werden sollen, könnte der Benutzer den folgenden Befehl LOAD absetzen:

```
db2 load from load.del of del method P(1, 3, 4) replace
into table1 (c1, c3, c4)
```



Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Wert `identityignore` vereinfacht die Syntax folgendermaßen:

```
db2 load from load.del of del modified by identityignore replace into table1
```

- Der Wert `identityoverride` wird verwendet, um benutzerdefinierte Werte in eine Tabelle mit einer als `GENERATED ALWAYS` definierten Identitätsspalte zu laden. Dies kann bei der Migration von Daten aus einem anderen Datenbanksystem, bei der die Tabelle als `GENERATED ALWAYS` definiert sein muß, oder beim Laden einer Tabelle aus Daten nützlich sein, die mit der Option `DROPPED TABLE RECOVERY` des Befehls `ROLLFORWARD DATABASE` wiederhergestellt wurden. Bei Verwendung dieses Werts werden alle Zeilen ohne Daten (oder mit Nulldaten) für die Identitätsspalte zurückgewiesen (SQL3116W).

**Anmerkung:** Die Verwendung dieses Werts kann das Eindeutigkeitsmerkmal von Spalten verletzen, die als `GENERATED ALWAYS` definiert sind.

---

## Verwenden von LOAD mit generierten Spalten

Mit dem Dienstprogramm `LOAD` können Sie Daten in eine Tabelle laden, die generierte Spalten enthält, bei denen es sich nicht um Identitätsspalten handelt. Gegenwärtig unterstützt es jedoch innerhalb des Dienstprogramms die Generierung von Werten für Spalten, die keine Identitätsspalten sind, nicht. Daher verbleibt beim Laden in eine Tabelle, die solche Spalten enthält, die Tabelle immer im Status „Überprüfung anstehend“. Um den Status „Überprüfung anstehend“ für die Tabelle zu beenden und die Generierung von Werten zu erzwingen, setzen Sie den folgenden Befehl ab:

```
SET INTEGRITY FOR tabellenname IMMEDIATE CHECKED FORCE GENERATED;
```

Um die Werte in ihrer geladenen Form zu akzeptieren, setzen Sie den folgenden Befehl ab:

```
SET INTEGRITY FOR tabellenname GENERATED COLUMN IMMEDIATE UNCHECKED;
```

Sofern keine Dateitypwerte verwendet werden, die sich auf generierte Spalten beziehen, liegen der Ausführung des Dienstprogramms `LOAD` die folgenden Regeln zugrunde:

- Bei generierten Spalten, die Nullwerte enthalten können, wird ein Nullwert akzeptiert und in die Tabelle geladen, wenn ein Nullwert zur Verfügung gestellt wird. Bei generierten Spalten, die keine Nullwerte enthalten dürfen, wird der Standardwert für den Datentyp der generierten Spalte geladen, wenn in der Datendatei für die Spalte ein Nullwert zur Verfügung gestellt wird. Sowohl bei generierten Spalten, die Nullwerte enthalten dürfen, als

## Verwenden von LOAD mit generierten Spalten

auch bei generierten Spalten, die keine Nullwerte enthalten dürfen, wird die Zeile zurückgewiesen, wenn die Datendatei Werte enthält, die keine Nullwerte sind (SQL3550W).

- Nach dem Abschluß der Ladeoperation werden geladene Werte durch generierte Werte ersetzt, sofern der Status „Überprüfung anstehend“ für die Tabelle durch die Anweisung SET INTEGRITY mit der Option FORCE GENERATED beendet wird. Wird die Option FORCE GENERATED nicht verwendet, bleiben die geladenen Werte erhalten, sofern folgendes zutrifft:
  - Die Werte verletzen die Integritätsbedingungen von generierten Spalten nicht.
  - Die Option GENERATED COLUMN IMMEDIATE UNCHECKED wird verwendet.

Wenn die Option GENERATED COLUMN IMMEDIATE UNCHECKED *nicht* verwendet wird, werden Zeilen, die generierte Spalten mit ungültigen Integritätsbedingungen enthalten, in eine Ausnahmetabelle gestellt, sofern eine solche zur Verfügung steht.

Das Dienstprogramm LOAD unterstützt drei (sich gegenseitig ausschließende) Dateitypwerte, die die Verwendung dieses Dienstprogramms für Tabellen mit generierten Spalten vereinfachen:

- Der Wert generatedmissing vereinfacht das Laden einer Tabelle mit generierten Spalten, wenn die Eingabedatendatei für alle in der Tabelle vorhandenen generierten Spalten keine Werte (auch keine Nullwerte) enthält. Angenommen, es wurde beispielsweise eine Tabelle mit der folgenden SQL-Anweisung definiert:

```
create table table1 (c1 int,  
                    c2 int,  
                    g1 int generated always as (c1 + c2),  
                    g2 int generated always as (2 * c1),  
                    c3 char(1))
```

Ein Benutzer möchte Daten aus einer Datei (load.del) in die Tabelle TABLE1 laden. Diese Datei wurde aus einer Tabelle exportiert, die keine generierten Spalten enthält. Es folgt ein Beispiel für eine solche Datei.

```
1, 5, J  
2, 6, K  
3, 7, I
```

Eine Methode für das Laden dieser Datei wäre das explizite Auflisten der zu ladenden Spalten durch den folgenden Befehl LOAD:

```
db2 load from load.del of del replace into table1 (c1, c2, c3)
```

## Verwenden von LOAD mit generierten Spalten

Bei einer Tabelle mit vielen Spalten ist die Verwendung dieser Syntax jedoch eventuell umständlich und fehlerträchtig. Eine alternative Methode für das Laden der Datei ist die folgende Verwendung des Dateityperts `generatedmissing`:

```
db2 load from load.del of del modified by generatedmissing replace
into table1
```

- Der Wert `generatedignore` stellt in gewisser Hinsicht das Gegenteil des Werts `generatedmissing` dar: Er weist das Dienstprogramm LOAD an, daß die in der Eingabedatendatei vorhandenen Werte für alle in der Zieltabelle vorhandenen generierten Spalten ignoriert und in jede Spalte Nullwerte geladen werden sollen. Beispiel: Ein Benutzer möchte die wie oben definierte Tabelle TABLE1 aus einer Datendatei (load.del) mit den folgenden Daten laden:

```
1, 5, 10, 15, J
2, 6, 11, 16, K
3, 7, 12, 17, I
```

Die benutzerdefinierten Werte 10, 11, und 12 (für g1) sowie 15, 16 und 17 (für g2), die keine Nullwerte sind, bewirken, daß die Zeile zurückgewiesen wird (SQL3550W). Um dies zu verhindern, könnte der Benutzer den folgenden Befehl LOAD absetzen:

```
db2 load from load.del of del method P(1, 2, 5) replace
into table1 (c1, c2, c3)
```

Aber auch diese Methode kann möglicherweise umständlich und fehlerträchtig sein, wenn die Tabelle zu viele Spalten enthält. Der Wert `generatedignore` vereinfacht die Syntax folgendermaßen:

```
db2 load from load.del of del modified by generatedignore replace into table1
```

- Der Wert `generatedoverride` wird verwendet, um benutzerdefinierte Werte in eine Tabelle mit generierten Spalten zu laden. Dies kann bei der Migration von Daten aus einem anderen Datenbanksystem oder beim Laden einer Tabelle aus Daten nützlich sein, die mit der Option DROPPED TABLE RECOVERY des Befehls ROLLFORWARD DATABASE wiederhergestellt wurden. Bei Verwendung dieses Werts werden alle Zeilen ohne Daten (oder mit Nulldaten) für generierte Spalten, die keine Nullwerte enthalten können, zurückgewiesen (SQL3116W).

## Überprüfung auf Integritätsbedingungen

---

### Überprüfung auf ungültige Integritätsbedingungen

Nach einer Ladeoperation kann sich die geladene Tabelle im Status „Überprüfung anstehend“ befinden, wenn für sie Prüfungen auf Integritätsbedingungen oder referentielle Integritätsbedingungen definiert sind. Die Markierung STATUS des Eintrags für die geladene Tabelle in SYSCAT.TABLES gibt den Status „Überprüfung anstehend“ der Tabelle an. Damit die geladene Tabelle verwendbar ist, muß für STATUS der Wert N eingetragen sein, der den normalen Status der Tabelle bezeichnet.

Verwenden Sie die Anweisung SET INTEGRITY, um den Status „Überprüfung anstehend“ zu entfernen (siehe *SQL Reference*). Mit der Anweisung SET INTEGRITY wird eine Tabelle auf ungültige Integritätsbedingungen überprüft, und die Tabelle wird aus dem Status „Überprüfung anstehend“ herausgenommen. Wenn alle Ladeoperationen im Modus INSERT ausgeführt werden, werden die Integritätsbedingungen mit der Anweisung SET INTEGRITY standardmäßig inkrementell verarbeitet (d. h. es wird nur der angefügte Teil der Tabelle auf ungültige Integritätsbedingungen überprüft). Beispiel:

```
db2 load from infile1.ixf of ixf insert into table1
db2 set integrity for table1 immediate checked
```

Nur der angefügte Teil von TABLE1 wird auf ungültige Integritätsbedingungen überprüft. Die Überprüfung nur des angefügten Teils auf ungültige Integritätsbedingungen ist schneller als die Überprüfung der vollständigen Tabelle, vor allem bei einer umfangreichen Tabelle, an die kleine Datenmengen angefügt werden.

Es können eine oder mehrere Tabellen in einem einzigen Aufruf dieser Anweisung überprüft werden. Wenn eine abhängige Tabelle überprüft werden soll, darf sich die übergeordnete Tabelle nicht im Status „Überprüfung anstehend“ befinden. In Falle eines Zyklus für referentielle Integrität müssen alle von diesem Zyklus betroffenen Tabellen in einen einzigen Aufruf der Anweisung SET INTEGRITY aufgenommen werden. Es kann von Vorteil sein, die übergeordnete Tabelle auf ungültige Integritätsbedingungen zu überprüfen, während eine abhängige Tabelle geladen wird. Dies kann nur geschehen, wenn sich die beiden Tabellen nicht im selben Tabellenbereich befinden.

Beim Absetzen der Anweisung SET INTEGRITY können Sie die Option INCREMENTAL angeben, um die inkrementelle Verarbeitung explizit anzufordern. In den meisten Fällen wird diese Option nicht benötigt, weil das Standardverhalten die inkrementelle Verarbeitung ist. Wenn die inkrementelle Verarbeitung nicht möglich ist, wird automatisch die volle Verarbeitung verwendet. Wenn die Option INCREMENTAL angegeben ist, die inkrementelle Verarbeitung jedoch nicht möglich ist, wird in folgenden Fällen ein Fehler gemeldet:

- Neue Integritätsbedingungen wurden der Tabelle oder der übergeordneten Tabelle hinzugefügt, während sich beide Tabellen im Status „Überprüfung anstehend“ befinden.
- Eine LOAD REPLACE-Operation wurde ausgeführt, oder die Option NOT LOGGED INITIALLY WITH EMPTY TABLE wurde aktiviert, nachdem die letzte Überprüfung auf Integritätsbedingungen für die Tabelle stattgefunden hat.
- Für eine übergeordnete Tabelle wurde eine LOAD REPLACE-Operation ausgeführt, oder eine solche Tabelle wurde nicht inkrementell auf Integrität überprüft.
- Vor der Migration befand sich die Tabelle im Status „Überprüfung anstehend“. Wenn für die Tabelle zum ersten Mal nach der Migration eine Überprüfung auf Integritätsbedingungen vorgenommen wird, ist eine vollständige Verarbeitung erforderlich.
- Der Tabellenbereich, der die Tabelle oder ihre übergeordnete Tabelle enthält, wurde bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt.

Wenn die Tabelle T einen oder mehrere Werte W in der Spalte CONST\_CHECKED des Katalogs SYSCAT.TABLES aufweist (siehe Handbuch *SQL Reference*, Anweisung SET INTEGRITY, für eine Beschreibung des Status W), überprüft das System die gesamte Tabelle auf ungültige Integritätsbedingungen, wenn die Option INCREMENTAL nicht angegeben ist. Wenn die Option angegeben ist, ist sie zulässig, die Spalte CONST\_CHECKED von SYSCAT.TABLES wird jedoch mit U gekennzeichnet, um anzugeben, daß nicht alle Daten vom System überprüft wurden.

## Überprüfung auf Integritätsbedingungen

Verwenden Sie die Option zum Laden der Ausnahmetabelle, um Informationen zu Zeilen mit ungültigen Integritätsbedingungen zu erfassen (siehe „Ausnahmetabelle“ auf Seite 157).

Die Anweisung SET INTEGRITY aktiviert keine DELETE-Auslöser als Ergebnis des Löschens von Zeilen, die gegen Integritätsbedingungen verstoßen. Die Auslöser sind jedoch aktiv, sobald für die Tabelle der Status „Überprüfung anstehend“ aufgehoben wird. Demzufolge werden, wenn Sie Daten korrigieren und Zeilen aus der Ausnahmetabelle in die geladene Tabelle einfügen, auch die INSERT-Auslöser aktiviert, die möglicherweise für die Tabelle definiert sind. Die Implikationen davon sollten berücksichtigt werden. Eine Möglichkeit wäre z. B., den INSERT-Auslöser zu löschen, Zeilen aus der Ausnahmetabelle einzufügen und anschließend den INSERT-Auslöser erneut zu erstellen.

---

### Neustart einer unterbrochenen Ladeoperation

Wenn das Dienstprogramm LOAD aufgrund eines Benutzerfehlers nicht gestartet werden kann, etwa aufgrund einer nicht vorhandenen Datendatei oder ungültiger Spaltennamen, wird es beendet, und der Tabellenbereich verbleibt in einem normalen Status.

Tritt beim Laden von Daten ein Fehler auf, können Sie die Ladeoperation mit der Option RESTART vom letzten Konsistenzzustand ausgehend erneut starten oder die gesamte Tabelle mit der Option REPLACE erneut laden. Geben Sie die gleichen Parameter an wie beim vorherigen Aufruf, damit das Dienstprogramm die erforderlichen temporären Dateien findet.

---

### Verwenden der Datei mit den Angaben zur Speicherposition der Ladekopie

Anhand der Registrierungsvariablen DB2LOADREC wird die Datei mit den Angaben zur Speicherposition der Ladekopie identifiziert. Diese Datei wird während der aktualisierenden Wiederherstellung zum Lokalisieren der Ladekopie verwendet. Sie enthält folgende Informationen:

- Datenträgertyp
- Anzahl der zu verwendenden Datenträgereinheiten
- Speicherposition der Ladekopie, die bei einer Ladeoperation für eine Tabelle generiert wurde
- Dateiname der Ladekopie (sofern zutreffend)

## Verwenden der Datei mit den Angaben zur Speicherposition der Ladekopie

Falls die Speicherpositionsdatei nicht existiert oder darin kein übereinstimmender Eintrag gefunden wurde, werden die Informationen aus dem Protokollsatz verwendet.

Die Informationen in der Datei können überschrieben werden, bevor die aktualisierende Wiederherstellung stattfindet.

In einer Umgebung mit partitionierten Datenbanken muß die Datei der Ladekopie auf jedem Datenbankpartitions-Server vorhanden und der Dateiname (einschließlich des Pfads) derselbe sein.

Es folgt ein Beispiel für eine solche Datei. Die ersten fünf Parameter müssen gültige Werte aufweisen und werden zum Identifizieren der Ladekopie verwendet. Die gesamte Struktur wird für jede aufgezeichnete Ladekopie wiederholt.

```
TIMestamp      19950725182542  * Zur Ladezeit generierte Zeitmarke
SCHema         PAYROLL          * Schema der geladenen Tabelle
TABlename     EMPLOYEES        * Tabellename
DATabasename  DBT                    * Datenbankname
DB2instance   TORONTO          * DB2INSTANCE
BUFFernumber  NULL                    * Anzahl der Puffer für die Wiederherstellung
SESSionnumber NULL                    * Anzahl der Sitzungen für die Wiederherstellung
TYPeofmedia   L                        * Datenträgertyp - L für lokale Einheit
                                     A für Tivoli Storage Manager
                                     (TSM, vormals ADSM)
                                     0 für andere Lieferanten
LOCationnumber 3                * Anzahl der Speicherpositionen
ENTry         /u/toronto/dbt.payroll.employes.001
ENT          /u/toronto/dbt.payroll.employes.002
ENT          /dev/rmt0
TIM          19950725192054
SCH         PAYROLL
TAB         DEPT
DAT         DBT
DB2        TORONTO
SES        NULL
BUF        NULL
TYP        A
TIM        19940325192054
SCH         PAYROLL
TAB         DEPT
DAT         DBT
DB2        TORONTO
SES        NULL
BUF        NULL
TYP        0
SHRlib      /@sys/lib/backup_vendor.a
```

## Verwenden der Datei mit den Angaben zur Speicherposition der Ladekopie

### Anmerkungen:

1. Für jedes Schlüsselwort sind die ersten drei Zeichen wichtig. Alle Schlüsselwörter sind in der angegebenen Reihenfolge erforderlich. Leerzeilen werden nicht akzeptiert.
2. Die Zeitmarke hat das Format *jjjmmmtssmmss* (Jahr Monat Tag Stunde Minute Sekunde).
3. Alle Felder sind verbindlich, mit Ausnahme von BUF und SES, die den Wert NULL haben können.
4. Es gibt folgende Datenträgertypen: lokale Einheit (L für Band, Platte oder Diskette), Tivoli Storage Manager (A, vormals ADSM) oder Datenträger anderer Lieferanten (O). Lautet der Typ „L“, ist die Anzahl der Speicherpositionen, gefolgt von den Einträgen zur Speicherposition, erforderlich. Lautet der Typ „A“, ist keine weitere Eingabe erforderlich. Lautet der Typ „O“, ist der Name der gemeinsam benutzten Bibliothek erforderlich. Weitere Informationen zur Verwendung von Tivoli Storage Manager (TSM) oder der Produkte anderer Lieferanten als Sicherungsdatenträger finden Sie im Abschnitt zu Tivoli Storage Manager im Kapitel zur Datenbankwiederherstellung im Handbuch *Systemverwaltung*.
5. Der Parameter SHRlib zeigt auf eine Bibliothek, die zum Speichern der LOAD COPY-Daten dient.

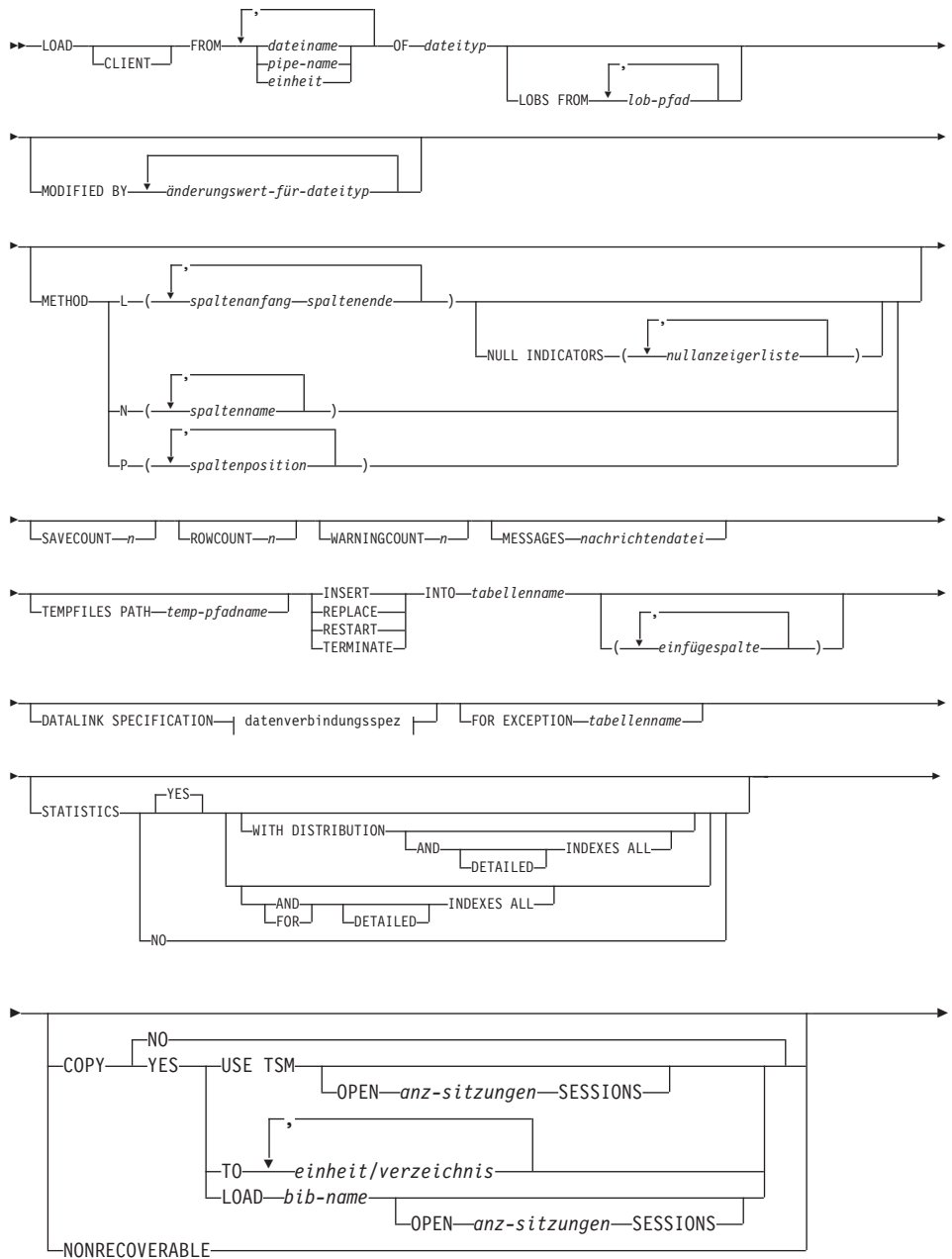
Wenn Sie LOAD COPY NO ausführen und nach Ausführung der Ladeoperation keine Sicherungskopie der Datenbank oder der betroffenen Tabellenbereiche erstellen, können Sie die Datenbank oder Tabellenbereiche nicht bis zu einem bestimmten Zeitpunkt wiederherstellen, der zeitlich nach der Ladeoperation liegt. Das heißt, Sie können keine aktualisierende Wiederherstellung ausführen, um die Datenbank bzw. die Tabellenbereiche auf einem Stand nach der Ausführung der Ladeoperation wiederherzustellen. Sie können die Datenbank bzw. die Tabellenbereiche lediglich bis zu einem Zeitpunkt *vor* der Ladeoperation wiederherstellen.

Für den Fall, daß Sie eine bestimmte Ladekopie verwenden wollen, werden die LOAD-Zeitmarken in der Datei des Wiederherstellungsprotokolls für die Datenbank aufgezeichnet. In einer partitionierte Datenbankumgebung ist die Datei des Wiederherstellungsprotokolls für jede Datenbankpartition lokal.

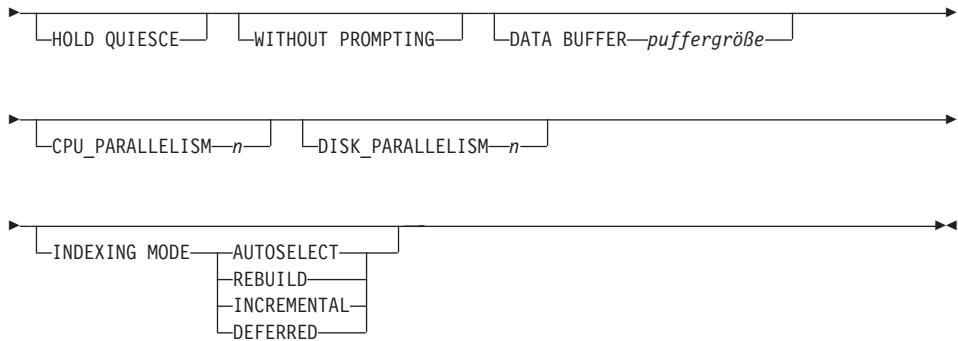


LOAD - Befehl

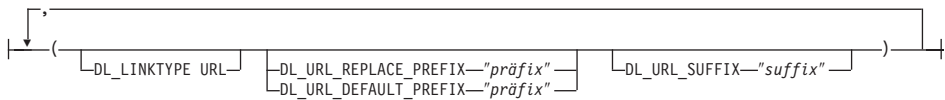
Befehlssyntax



## LOAD - Befehl



### datenverbindungsspez:



## Befehlsparameter

### CLIENT

Gibt an, daß sich die zu ladenden Daten auf einem Client mit Fernverbindung befinden. Diese Option wird ignoriert, wenn die Ladeoperation nicht durch einen fernen Client aufgerufen wird.

### Anmerkungen:

1. Das Laden von Daten, die sich auf einem Client mit Fernverbindung befinden, wird unter den folgenden Bedingungen nicht unterstützt:
  - Die Datenbank, mit der der Client verbunden ist, befindet sich in einer Umgebung mit DB2 Enterprise - Extended Edition .
  - Die Datenbank, mit der der Client verbunden ist, wird für eine bereits katalogisierte Datenbank katalogisiert.
2. Die Werte DUMPFILERE und LOBSINFILERE (siehe Tabelle 8 auf Seite 140) verweisen selbst bei Angabe des Schlüsselworts CLIENT auf Dateien, die sich auf dem Server befinden.
3. Die Codepage wird während einer fernen Ladeoperation nicht umgesetzt. Wenn sich die Codepage der Daten von der Codepage des Servers unterscheidet, muß die Codepage der Daten mit dem Wert CODEPAGE angegeben werden (siehe Tabelle 8 auf Seite 140).
4. Beim Laden ferner Client-Daten aus Dateien gilt eine Höchstgrenze von 2 GB pro Datei. Dies gilt jedoch nicht für benannte Pipes.

Im folgenden Beispiel soll eine Datendatei (/u/user/data.del), die sich auf einem Client mit Fernverbindung befindet, in die Tabelle MYTABLE der Server-Datenbank geladen werden:

```
db2 load client from /u/user/data.del of del
modified by codepage=850 insert into mytable
```

### **COPY NO**

Gibt an, daß der Tabellenbereich, in dem sich die Tabelle befindet, in den Status „Sicherung anstehend“ versetzt wird, wenn die aktualisierende Wiederherstellung aktiviert ist (das heißt, *logretain* oder *userexit* ist aktiv). Die Daten in allen Tabellen des Tabellenbereichs können erst dann aktualisiert oder gelöscht werden, nachdem eine Sicherung des Tabellenbereichs oder eine vollständige Sicherung der Datenbank durchgeführt wurde. Der Zugriff auf die Daten in allen Tabellen ist jedoch durch die Anweisung SELECT möglich.

### **COPY YES**

Gibt an, daß eine Kopie der geladenen Daten gesichert wird. Diese Option ist unzulässig, wenn die aktualisierende Wiederherstellung inaktiviert ist (sowohl *logretain* als auch *userexit* sind inaktiv). Die Option wird für Tabellen mit DATALINK-Spalten nicht unterstützt.

### **USE TSM**

Gibt an, daß die Kopie unter Verwendung von Tivoli Storage Manager (TSM) gespeichert wird.

### **OPEN anz-sitzungen SESSIONS**

Die Anzahl der E/A-Sitzungen, die mit Tivoli Storage Manager (TSM) oder einem Produkt eines anderen Lieferanten verwendet werden sollen. Der Standardwert ist 1.

### **TO einheit/verzeichnis**

Gibt die Einheit oder das Verzeichnis an, auf der bzw. in dem das Kopierabbild erstellt wird. Ein Band wird unter OS/2 nicht unterstützt. Das Kopieren auf Bänder wird für DB2-Server unter SCO UnixWare 7 nicht unterstützt.

### **LOAD bib-name**

Der Name der gemeinsam benutzten Bibliothek (DLL unter OS/2 oder Windows), die die Sicherungs- und Wiederherstellungs-E/A-Funktionen des Lieferanten enthält, die verwendet werden sollen. Kann den vollständigen Pfad enthalten. Wenn der vollständige Pfad nicht angegeben wird, wird standardmäßig der Pfad verwendet, in dem sich die Benutzer-Exit-Programme befinden.

### **CPU\_PARALLELISM n**

Gibt die Anzahl der Prozesse oder Threads an, die vom Dienstprogramm LOAD gestartet werden, um Datensätze beim Erstellen von

Tabellenobjekten syntaktisch zu analysieren, umzusetzen und zu formatieren. Dieser Parameter soll die Nutzung der partitionsinternen Parallelität ermöglichen. Er ist besonders nützlich beim Laden von vorsortierten Daten, weil die Satzreihenfolge der Quelldaten beibehalten wird. Wenn der Wert dieses Parameters Null ist oder nicht angegeben wurde, verwendet das Dienstprogramm LOAD zur Laufzeit einen sinnvollen Standardwert. Dieser basiert in der Regel auf der Anzahl der verfügbaren Zentraleinheiten.

### Anmerkungen:

1. Wenn dieser Parameter bei Tabellen verwendet wird, die LOB- oder LONG VARCHAR-Felder enthalten, wird sein Wert Eins, unabhängig von der Anzahl der System-CPU's oder dem vom Benutzer angegebenen Wert.
2. Die Angabe eines kleinen Werts für den Parameter SAVECOUNT bewirkt, daß das Ladeprogramm viel mehr E/A-Vorgänge ausführen muß, um sowohl Daten als auch Tabellenmetadaten zu löschen. Wenn CPU\_PARALLELISM größer als Eins ist, erfolgen die Löschvorgänge asynchron, so daß das Ladeprogramm die CPU nutzen kann. Wenn CPU\_PARALLELISM auf Eins gesetzt ist, wartet das Ladeprogramm bei Konsistenzzuständen auf E/A. Eine Ladeoperation wird, wenn CPU\_PARALLELISM auf Zwei und SAVECOUNT auf 10 000 gesetzt ist, schneller ausgeführt als die gleiche Operation, wenn CPU\_PARALLELISM auf Eins gesetzt ist, auch wenn nur eine CPU vorhanden ist.

### DATA BUFFER puffergröße

Gibt die Anzahl der 4 KB-Seiten (unabhängig vom Grad der Parallelität) an, die als Pufferbereich zur Übertragung von Daten im Dienstprogramm verwendet werden sollen. Wenn der angegebene Wert kleiner als der algorithmische Mindestwert ist, werden die erforderlichen Mindestressourcen verwendet, und es wird keine Warnung zurückgegeben.

Dieser Speicher wird direkt aus dem Zwischenspeicher der Dienstprogramme zugeordnet, dessen Größe mit dem Datenbankkonfigurationsparameter *util\_heap\_sz* geändert werden kann.

Wenn kein Wert angegeben wird, wird vom Dienstprogramm zur Laufzeit ein sinnvoller Standardwert berechnet. Der Standardwert basiert auf einem Prozentsatz des freien Speicherbereichs, der bei der Exemplarerstellung des Ladeprogramms im Zwischenspeicher der Dienstprogramme verfügbar ist, sowie auf einigen Merkmalen der Tabelle.

### DATALINK SPECIFICATION

Für jede DATALINK-Spalte ist eine Spaltenangabe in runden Klammern möglich. Jede Spaltenangabe umfaßt mindestens einen

DL\_LINKTYPE-Wert, eine Präfix- und eine DL\_URL\_SUFFIX-Angabe. Die Präfixangabe kann entweder DL\_URL\_REPLACE\_PREFIX oder DL\_URL\_DEFAULT\_PREFIX sein.

Es sind so viele DATALINK-Spaltenangaben möglich, wie DATALINK-Spalten in der Tabelle definiert sind. Die Reihenfolge der Angaben richtet sich nach der Reihenfolge der DATALINK-Spalten in der Liste *einfügespalte* oder in der Tabellendefinition (wenn keine Liste *einfügespalte* angegeben wird).

#### **DISK\_PARALLELISM n**

Gibt die Anzahl der Prozesse oder Threads an, die vom Dienstprogramm LOAD gestartet werden, um Daten in die Tabellenbereichsbehälter zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichsbehälter und der Merkmale der Tabelle aus.

#### **DL\_LINKTYPE**

Muß, wenn angegeben, mit dem LINKTYPE-Wert der Spaltendefinition übereinstimmen. Demzufolge ist DL\_LINKTYPE URL akzeptabel, wenn die Spaltendefinition LINKTYPE URL angibt.

#### **DL\_URL\_DEFAULT\_PREFIX "präfix"**

Muß, wenn angegeben, als Standardpräfix für alle DATALINK-Werte in der gleichen Spalte verwendet werden. In diesem Zusammenhang bezieht sich Präfix auf den Teil „Schema-Host-Anschluß“ der URL-Angabe. (Bei einem verteilten Dateisystem bezieht sich Präfix auf den Teil „Schema-Zellenname-Dateibereichsverbindung“ der URL-Angabe.)

Beispiele für Präfixe sind:

```
"http://server"
"file://server"
"file:"
"http://server:80"
"dfs://.../cellname/fs"
```

Enthalten die Spaltendaten kein Präfix und wird mit DL\_URL\_DEFAULT\_PREFIX ein Standardpräfix angegeben, wird dieses dem Spaltenwert (sofern nicht NULL) vorangestellt.

Beispiel: Wenn DL\_URL\_DEFAULT\_PREFIX das Standardpräfix "http://toronto" angibt, gilt folgendes:

- Der Spalteneingabewert "/x/y/z" wird als "http://toronto/x/y/z" gespeichert.
- Der Spalteneingabewert "http://coyote/a/b/c" wird als "http://coyote/a/b/c" gespeichert.
- Der Spalteneingabewert NULL wird als NULL gespeichert.

### **DL\_URL\_REPLACE\_PREFIX "präfix"**

Diese Klausel ist zum Laden oder Importieren von Daten nützlich, die zuvor vom Dienstprogramm EXPORT generiert wurden, wenn der Benutzer den Host-Namen in den Daten global durch einen anderen Host-Namen ersetzen möchte. Wird, wenn angegeben, das Präfix für *alle* Spaltenwerte, die nicht NULL sind. Wenn ein Spaltenwert mit einem Präfix versehen ist, wird dieses durch das mit DL\_URL\_REPLACE\_PREFIX angegebene Präfix ersetzt. Wenn ein Spaltenwert kein Präfix aufweist, wird das mit DL\_URL\_REPLACE\_PREFIX angegebene Präfix dem Spaltenwert vorangestellt. Bei einem verteilten Dateisystem bezieht sich Präfix auf den Teil „Schema-Zellenname-Dateibereichsverbindung“ der URL-Angabe.

Beispiel: Wenn DL\_URL\_REPLACE\_PREFIX das Präfix "http://toronto" angibt, gilt folgendes:

- Der Spalteneingabewert "/x/y/z" wird als "http://toronto/x/y/z" gespeichert.
- Der Spalteneingabewert "http://coyote/a/b/c" wird als "http://toronto/a/b/c" gespeichert. Beachten Sie, daß "toronto" "coyote" ersetzt.
- Der Spalteneingabewert NULL wird als NULL gespeichert.

### **DL\_URL\_SUFFIX "suffix"**

Wird, wenn angegeben, an jeden Spaltenwert für die Spalte, der nicht NULL ist, angefügt. Wird an die Pfadkomponente des Datenspeicherpositionsteils des DATALINK-Werts angefügt.

### **FOR EXCEPTION tabellenname**

Gibt die Ausnahmetabelle an, in die fehlerhafte Zeilen kopiert werden. Jede Zeile, die gegen einen eindeutigen Index oder einen Primärschlüsselindex verstößt, wird kopiert. DATALINK-Ausnahmen werden ebenfalls in der Ausnahmetabelle erfaßt. Wenn ein Tabellenname ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

Informationen, die in die Ausnahmetabelle geschrieben werden, werden *nicht* in die Speicherauszugsdatei geschrieben (eine Beschreibung des Änderungswerts `dumpfile` finden Sie in Tabelle 8 auf Seite 140). In einer Umgebung mit partitionierten Datenbanken muß eine Ausnahmetabelle für die Knoten definiert werden, auf denen die zu ladende Tabelle definiert ist. Die Speicherauszugsdatei andererseits enthält Zeilen, die nicht geladen werden können, weil sie ungültig sind oder Syntaxfehler aufweisen. Weitere Informationen finden Sie unter „Ausnahmetabelle“ auf Seite 157.

### **FROM dateiname/pipe-name/einheit**

Gibt die Datei, Pipe oder Einheit an, die die zu ladenden Daten enthält. Sofern nicht die Option CLIENT angegeben wird, muß sich diese

Datei, Pipe oder Einheit auf dem Knoten befinden, auf dem sich auch die Datenbank befindet. Wenn mehrere Namen angegeben werden, werden sie nacheinander verarbeitet. Wenn das letzte angegebene Element ein Bandlaufwerk ist, wird der Benutzer aufgefordert, ein weiteres Band einzulegen. Gültige Antwortoptionen sind:

- c** Fortsetzen. Verwendung der Einheit fortsetzen, die die Warnung generiert hat (zum Beispiel, wenn ein neues Band eingelegt wurde).
- d** Einheit beenden. Verwendung der Einheit beenden, die die Warnung generiert hat (zum Beispiel, wenn keine Bänder mehr vorhanden sind).
- t** Beenden. Alle Einheiten beenden.

#### **Anmerkungen:**

1. Ein Band wird unter OS/2 nicht unterstützt.
2. Es empfiehlt sich, den vollständig qualifizierten Dateinamen zu verwenden. Wenn es sich um einen fernen Server handelt, muß der vollständig qualifizierte Dateiname verwendet werden. Wenn sich die Datenbank auf dem gleichen Knoten befindet wie der Aufrufer, können relative Pfade verwendet werden.
3. Das Laden von Daten aus mehreren IXF-Dateien wird unterstützt, wenn es sich physisch um separate Dateien, logisch jedoch um eine einzige Datei handelt. Es wird *nicht* unterstützt, wenn die Dateien sowohl logisch als auch physisch separate Dateien sind.
4. Wenn bei Angabe von *pipe-name* unter OS/2 weniger als die erwartete Menge von Daten geladen wird, müssen Systemressourcen bereinigt werden (Neustart des Systems empfohlen) und muß der Befehl LOAD erneut abgesetzt werden.
5. Werden Daten geladen, die sich auf einer Client-Maschine befinden, müssen die Daten entweder in Form einer vollständig qualifizierten Datei oder einer benannten Pipe vorliegen.

#### **HOLD QUIESCE**

Gibt an, daß das Dienstprogramm die Tabelle nach der Ladeoperation im exklusiven Wartemodus belassen soll. Setzen Sie folgenden Befehl ab, um den Wartemodus der Tabellenbereiche zu beenden:

```
db2 quiesce tablespaces for table <tabellenname> reset
```

**Anmerkung:** Stellen Sie sicher, daß keine *Scheinwartemodi* (Phantom Quiesce) erstellt werden (siehe Handbuch *Command Reference*).

#### **INDEXING MODE**

Gibt an, ob das Dienstprogramm LOAD Indizes erneut erstellen oder sie inkrementell erweitern soll. Gültige Werte:

### AUTOSELECT

Das Dienstprogramm LOAD wählt automatisch den Modus REBUILD oder INCREMENTAL.

### REBUILD

Alle Indizes werden erneut erstellt. Das Dienstprogramm benötigt genug Ressourcen, um alle Indexschlüsselteile für alte und angefügte Tabellendaten zu sortieren.

### INCREMENTAL

Indizes werden mit neuen Daten erweitert. Dieser Ansatz verbraucht freien Indexspeicherbereich. Er erfordert nur genug Sortierspeicherplatz, um Indexschlüssel für die eingefügten Datensätze anzufügen. Diese Methode wird nur in Fällen unterstützt, in denen das Indexobjekt bei Beginn einer Ladeoperation gültig ist und darauf zugegriffen werden kann (es ist zum Beispiel unmittelbar nach einer Ladeoperation, bei der der Modus DEFERRED angegeben wurde, nicht gültig). Wenn dieser Modus angegeben ist, jedoch aufgrund des Status des Index nicht unterstützt wird, wird eine Warnung zurückgegeben, und die Ladeoperation wird im Modus REBUILD fortgesetzt. Ähnliches gilt, wenn eine LOAD RESTART-Operation während der LOAD BUILD-Phase gestartet wird. In diesem Fall wird der Modus INCREMENTAL nicht unterstützt.

Die inkrementelle Indizierung wird nicht unterstützt, wenn alle folgenden Bedingungen zutreffen:

- Die Option LOAD COPY wurde angegeben (*logretain* oder *userexit* ist aktiv).
- Die Tabelle befindet sich in einem DMS-Tabellenbereich.
- Das Indexobjekt befindet sich in einem Tabellenbereich, der mit anderen Tabellenobjekten gemeinsam benutzt wird, die zu der geladenen Tabelle gehören.

Es empfiehlt sich, die Indizes in einen separaten Tabellenbereich zu stellen, um diese Einschränkung zu umgehen.

### DEFERRED

Das Dienstprogramm LOAD erstellt keine Indizes, wenn dieser Modus angegeben wird. Die Indizes werden als aktualisierungsbedürftig gekennzeichnet. Der erste Zugriff auf solche Indizes, der nicht im Zusammenhang mit der Ladeoperation steht, kann zu einer erneuten Erstellung führen (weitere Informationen finden Sie im Handbuch *Systemverwaltung*), oder die Indizes können erneut erstellt werden, wenn die Datenbank erneut gestartet wird. Dieser Ansatz erfordert genug Sortierspeicherplatz für alle Schlüsselteile für den größten Index. Die Gesamtzeit, die für den Aufbau des Index benötigt wird, ist



länger als im Modus REBUILD erforderlich. Unter Leistungsgesichtspunkten ist es daher bei der Durchführung mehrerer Ladeoperationen mit verzögerter Indexierung ratsam, bei der letzten Ladeoperation eine erneute Erstellung des Indexes durchführen zu lassen, statt die Indizes beim ersten, nicht mit der Ladeoperation im Zusammenhang stehenden Zugriff erneut erstellen zu lassen.

Die verzögerte Indexierung wird nur für Tabellen mit nicht eindeutigen Indizes unterstützt, so daß doppelte Schlüssel, die während der LOAD-Phase eingefügt werden, nach der Ladeoperation nicht erhalten bleiben.

**Anmerkung:** Bei Tabellen mit DATALINK-Spalten wird die verzögerte Indexierung nicht unterstützt.

## INSERT

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Fügt die geladenen Daten in die Tabelle ein, ohne die vorhandenen Tabellendaten zu ändern.

### einfügespalte

Gibt die Tabellenspalte an, in die die Daten eingefügt werden sollen.

Das Dienstprogramm LOAD kann keine Spalten syntaktisch analysieren, deren Namen Leerzeichen enthalten. Beispiel:

```
db2 load from delfile1 of del modified by noeofchar noheader
method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
insert into table1 (BLOB1, S2, I3, Int 4, I5, I6, DT7, I8, TM9)
```

Dieser Befehl schlägt aufgrund der Spalte Int 4 fehl. Die Lösung besteht darin, solche Spaltennamen in doppelte Anführungszeichen einzuschließen:

```
db2 load from delfile1 of del modified by noeofchar noheader
method P (1, 2, 3, 4, 5, 6, 7, 8, 9)
insert into table1 (BLOB1, S2, I3, "Int 4", I5, I6, DT7, I8, TM9)
```

## INTO tabellenname

Gibt die Datenbanktabelle an, in die die Daten geladen werden sollen. Diese Tabelle darf keine Systemtabelle oder deklarierte temporäre Tabelle sein. Es kann ein Aliasname, der vollständig qualifizierte Name oder ein Name ohne Qualifikationsmerkmal angegeben werden. Ein qualifizierter Tabellenname hat das Format *schema.tabellenname*. Wenn ein Tabellenname ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

## LOBS FROM lob-pfad

Der Pfad zu den Datendateien mit LOB-Werten, die geladen werden

## LOAD - Befehl

sollen. Der Pfad muß mit einem Schrägstrich (/) enden. Falls die Option CLIENT angegeben wird, muß es sich um einen vollständig qualifizierten Pfad handeln. Die Namen der LOB-Datendateien werden in der Hauptdatendatei (ASC, DEL oder IXF) in der Spalte gespeichert, die in die LOB-Spalte geladen wird. Diese Option wird ignoriert, wenn lobsinfile nicht in der Zeichenfolge *änderungswert-für-dateityp* angegeben wird (siehe Tabelle 8 auf Seite 140).

### MESSAGES *nachrichtendatei*

Gibt das Ziel für Warnungen und Fehlernachrichten an, die während der Ladeoperation auftreten. Wenn keine Nachrichtendatei angegeben wird, werden die Nachrichten in die Standardausgabe geschrieben. Wenn der vollständige Pfad zu der Datei nicht angegeben wird, verwendet das Dienstprogramm LOAD das aktuelle Verzeichnis und das Standardlaufwerk als Ziel. Wenn der Name einer Datei, die bereits vorhanden ist, angegeben wird, fügt das Dienstprogramm die Informationen an.

Die Nachrichtendaten wird für gewöhnlich am Ende der Ladeoperation mit Nachrichten gefüllt und ist insofern nicht geeignet, um den Fortschritt der Operation zu überwachen. Informationen zur Echtzeitüberwachung der Ladeoperation finden Sie unter „LOAD QUERY - Befehl“ auf Seite 116.

### METHOD

**L** Gibt die Anfangs- und Endnummer der Spalten an, aus denen Daten geladen werden sollen. Eine Spaltennummer ist eine relative Byteadresse in Bezug auf den Anfang einer Datenzeile. Ihre Numerierung beginnt bei 1.

**Anmerkung:** Diese Methode kann nur mit ASC-Dateien verwendet werden und ist die einzige gültige Option für diesen Dateityp.

**N** Gibt die Namen der Spalten in der zu ladenden Datendatei an. Die Schreibweise dieser Spaltennamen muß mit der Schreibweise der entsprechenden Namen in den Systemkatalogen übereinstimmen. Für jede Tabellenspalte, die keine Nullwerte enthalten kann, muß ein entsprechender Eintrag in der Liste METHOD N vorhanden sein. Beispiel: Für die Datenfelder F1, F2, F3, F4, F5 und F6 sowie die Tabellenspalten C1 INT, C2 INT NOT NULL, C3 INT NOT NULL und C4 INT ist method N (F2, F1, F4, F3) eine gültige Anforderung, wohingegen method N (F2, F1) nicht gültig ist.

**Anmerkung:** Diese Methode kann nur mit IXF-Dateien verwendet werden.

**P** Gibt die Indizes (numeriert beginnend bei 1) der zu ladenden Eingabedatenfelder an. Für jede Tabellenspalte, die keine Nullwerte enthalten kann, muß ein entsprechender Eintrag in der Liste METHOD P vorhanden sein. Beispiel: Für die Datenfelder F1, F2, F3, F4, F5 und F6 sowie die Tabellenspalten C1 INT, C2 INT NOT NULL, C3 INT NOT NULL und C4 INT ist method P (2, 1, 4, 3) eine gültige Anforderung, wohingegen method P (2, 1) nicht gültig ist.

**Anmerkung:** Diese Methode kann nur mit IXF- oder DEL-Dateien verwendet werden und ist die einzige gültige Option für den Dateityp DEL.

#### **MODIFIED BY änderungswert-für-dateityp**

Gibt zusätzliche Optionen an (siehe Tabelle 8 auf Seite 140).

#### **NONRECOVERABLE**

Gibt an, daß eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und daß sie nicht durch eine nachfolgende aktualisierende Wiederherstellung wiedergestellt werden kann. Das Dienstprogramm zur aktualisierenden Wiederherstellung überspringt die Transaktion und kennzeichnet die Tabelle, in die die Daten geladen wurden, als "ungültig". Vom Dienstprogramm werden auch alle nachfolgenden Transaktionen für diese Tabelle ignoriert. Nach Abschluß der Operation für die aktualisierende Wiederherstellung kann eine solche Tabelle nur nach einem COMMIT-Punkt, der auf den Abschluß der nicht wiederherstellbaren Ladeoperation folgt, gelöscht oder aus einer Sicherung (vollständige Sicherung oder Sicherung des Tabellenbereichs) wiederhergestellt werden.

Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status „Sicherung anstehend“ gesetzt, und während der Ladeoperation muß keine Kopie der geladenen Daten erstellt werden.

Diese Option sollte nicht verwendet werden, wenn DATALINK-Spalten mit dem Attribut FILE LINK CONTROL in der Tabelle vorhanden sind oder ihr hinzugefügt werden.

#### **NULL INDICATORS nullanzeigerliste**

Diese Option kann nur verwendet werden, wenn der Parameter METHOD L angegeben ist, d. h. wenn es sich bei der Eingabedatei um eine ASC-Datei handelt. Die Nullanzeigerliste ist eine durch Kommata gegliederte Liste positiver ganzer Zahlen, die für jedes Nullanzeigerfeld die Spaltennummer angeben. Die Spaltennummer gibt die relative Byteadresse des Nullanzeigerfelds in Bezug auf den Anfang einer Datenzeile an. Die Nullanzeigerliste muß für jedes Datenfeld,

## LOAD - Befehl

das im Parameter METHOD L definiert ist, einen Wert enthalten. Eine Spaltennummer 0 gibt an, daß das entsprechende Datenfeld immer Daten enthält.

Y in der Nullanzeigerspalte gibt an, daß die Spaltendaten NULL sind. Jedes *andere* Zeichen als Y in der Nullanzeigerspalte gibt an, daß die Spaltendaten nicht NULL sind und daß die mit der Option METHOD L angegebenen Spaltendaten geladen werden.

Das Nullanzeigerzeichen kann mit der Option MODIFIED BY geändert werden (siehe die Beschreibung des Änderungswerts nullindchar in Tabelle 8 auf Seite 140).

### OF dateityp

Gibt das Format der Daten in der Eingabedatei an:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer)
- IXF (Integrated Exchange Format, PC-Version), aus der gleichen oder einer anderen DB2-Tabelle exportiert.

Weitere Informationen zu Dateiformaten finden Sie in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

### REPLACE

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Löscht alle vorhandenen Daten aus der Tabelle und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert. Wenn diese Option beim Versetzen von Daten zwischen Hierarchien verwendet wird, können nur die Daten für eine vollständige Hierarchie, nicht jedoch für einzelne untergeordnete Tabellen ersetzt werden.

Diese Option wird für Tabellen mit DATALINK-Spalten nicht unterstützt.

### RESTART

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Startet eine zuvor unterbrochene Ladeoperation erneut. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand in der LOAD-, BUILD- oder DELETE-Phase ausgehend fortgesetzt.

### RESTARTCOUNT

Reserviert.

### ROWCOUNT n

Gibt die Anzahl von *n* physischen Sätzen in der Datei an, die geladen werden sollen. Ermöglicht es einem Benutzer, nur die ersten *n* Zeilen in einer Datei zu laden.

**SAVECOUNT *n***

Gibt an, daß das Dienstprogramm LOAD alle *n* Zeilen Konsistenzzustände einrichten soll. Dieser Wert wird in einen Seitenzähler umgesetzt und auf Intervalle in der Größe des Speicherbereichs gerundet. Da bei jedem Konsistenzzustand eine Nachricht abgesetzt wird, sollte diese Option ausgewählt werden, wenn die Ladeoperation mit einem LOAD QUERY-Befehl überwacht wird (siehe „LOAD QUERY - Befehl“ auf Seite 116. Wenn der Wert von *n* nicht groß genug ist, wirkt sich die Synchronisation der Aktivitäten, die bei jedem Konsistenzzustand ausgeführt werden, nachteilig auf die Leistung aus.

Der Standardwert ist Null, d. h., daß Konsistenzzustände nur definiert werden, sofern dies notwendig ist.

**SORT BUFFER *puffergröße***

Reserviert.

**STATISTICS NO**

Gibt an, daß keine Statistikdaten gesammelt werden sollen und daß die Statistikdaten in den Katalogen nicht geändert werden dürfen. Dies ist der Standardwert.

**STATISTICS YES**

Gibt an, daß Statistikdaten für die Tabelle und für alle vorhandenen Indizes gesammelt werden sollen. Diese Option wird nur unterstützt, wenn die Ladeoperation im Modus REPLACE ausgeführt wird.

**WITH DISTRIBUTION**

Gibt an, daß Verteilungsstatistikdaten gesammelt werden sollen.

**AND INDEXES ALL**

Gibt an, daß sowohl Tabellen- als auch Indexstatistikdaten gesammelt werden sollen.

**FOR INDEXES ALL**

Gibt an, daß nur Indexstatistikdaten gesammelt werden sollen.

**DETAILED**

Gibt an, daß erweiterte Indexstatistikdaten gesammelt werden sollen.

**TEMPFILES PATH *temp-pfadname***

Gibt den Namen des Pfads an, der beim Erstellen von temporären Dateien während einer Ladeoperation verwendet werden soll. Er muß bezüglich des Server-Knotens vollständig qualifiziert sein.

Temporäre Dateien belegen Speicherplatz im Dateisystem. Manchmal ist dieser Platzbedarf beträchtlich. Im folgenden finden Sie Richtwerte dazu, wieviel Dateisystemplatz für alle temporären Dateien zugeordnet werden sollte:

## LOAD - Befehl

- 4 Byte für jede doppelte oder zurückgewiesene Zeile, die DATALINK-Werte enthält
- 136 Byte für jede Nachricht, die vom Dienstprogramm LOAD generiert wird
- 15 KB zusätzlich, wenn die Datendatei Langfelddaten oder LOBs enthält. Diese Menge kann erheblich anwachsen, wenn die Option INSERT angegeben wird und sich eine große Menge von Langfeld- oder LOB-Daten bereits in der Tabelle befindet.

Weitere Informationen zu temporären Dateien finden Sie in „Temporäre Dateien (LOAD)“ auf Seite 158.

### TERMINATE

Einer von vier Modi, in denen das Dienstprogramm LOAD ausgeführt werden kann. Beendet eine zuvor unterbrochene Ladeoperation und macht die Operation bis zu dem Zeitpunkt, zu dem sie gestartet wurde, rückgängig, auch wenn Konsistenzzustände übergangen wurden. Die Statuswerte von betroffenen Tabellenbereichen werden in „Normal“ zurückgeändert, und alle Tabellenobjekte werden konsistent gemacht (Indexobjekte können als ungültig gekennzeichnet werden, so daß beim nächsten Zugriff automatisch eine Indexwiederherstellung erfolgt). Wenn die beendete Ladeoperation eine LOAD REPLACE-Operation ist, wird die Tabelle nach der LOAD TERMINATE-Operation abgeschnitten, so daß eine leere Tabelle übrigbleibt. Wenn die beendete Ladeoperation eine LOAD INSERT-Operation ist, bleiben in der Tabelle nach der LOAD TERMINATE-Operation alle ursprünglich vorhandenen Datensätze erhalten.

Mit der Option LOAD TERMINATE wird der Status „Sicherung anstehend“ für Tabellenbereiche nicht aufgehoben.

**Anmerkung:** Diese Option wird für Tabellen mit DATALINK-Spalten nicht unterstützt.

### USING verzeichnis

Reserviert.

### WARNINGCOUNT n

Stoppt die Ladeoperation nach  $n$  Warnungen. Definieren Sie diesen Parameter, wenn keine Warnungen erwartet werden, jedoch eine Prüfung darauf, ob die richtige Datei und Tabelle verwendet werden, gewünscht wird. Wenn  $n$  Null ist oder diese Option nicht angegeben wird, wird die Ladeoperation unabhängig von der Anzahl der abgesetzten Warnungen fortgesetzt. Wenn die Ladeoperation gestoppt wird, da die Schwelle für Warnungen erreicht wurde, kann eine weitere Ladeoperation im RESTART-Modus gestartet werden. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand ausge-

hend fortgesetzt. Alternativ kann eine weitere Ladeoperation im REPLACE-Modus eingeleitet werden, die am Anfang der Eingabedatei beginnt.

**WITHOUT PROMPTING**

Gibt an, daß die Liste mit Datendateien alle Dateien enthält, die geladen werden sollen, und daß die aufgelisteten Einheiten oder Verzeichnisse für die gesamte Ladeoperation genügen. Wenn keine weitere Eingabedatei gefunden wird oder die Kopierziele gefüllt sind, bevor die Ladeoperation beendet ist, schlägt die Ladeoperation fehl, und die Tabelle verbleibt im Status „Laden anstehend“.

Wenn diese Option nicht angegeben wird und beim Bandlaufwerk für das Kopierabbild das Bandende erreicht wird oder das letzte aufgelistete Element ein Bandlaufwerk ist, wird der Benutzer aufgefordert, ein neues Band in das betreffende Laufwerk einzulegen. Ein Band wird unter OS/2 nicht unterstützt.

## LOAD QUERY - Befehl

---

### LOAD QUERY - Befehl

Überprüft den Status einer Ladeoperation während der Verarbeitung. Sie benötigen eine Verbindung zu derselben Datenbank und eine separate Sitzung des Befehlszeilenprozessors, um diesen Befehl eingeben zu können. Er kann von lokalen und fernen Benutzern verwendet werden.

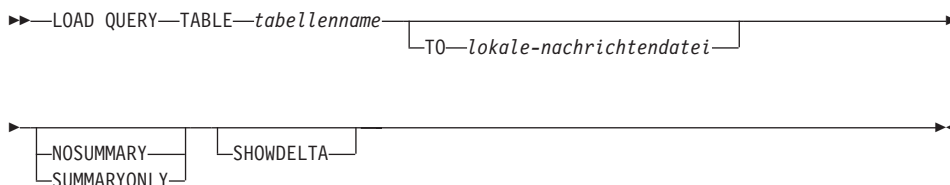
#### Berechtigung

Keine

#### Erforderliche Verbindung

Datenbank

#### Befehlssyntax



#### Befehlsparameter

##### NOSUMMARY

Gibt an, daß keine LOAD-Ergebnistexte (gelesene Zeilen, übersprungene Zeilen, geladene Zeilen, zurückgewiesene Zeilen, gelöschte Zeilen, festgeschriebene Zeilen und Anzahl der Warnungen) gemeldet werden.

##### SHOWDELTA

Gibt an, daß nur neue Informationen (bezüglich LOAD-Ereignissen, die seit dem letzten Aufruf des Befehls `LOAD QUERY` eingetreten sind) gemeldet werden.

##### SUMMARYONLY

Gibt an, daß nur Ergebnistexte gemeldet werden.

##### TABLE *tabellenname*

Gibt den Namen der Tabelle an, in die die Daten geladen werden. Wenn ein Tabellename ohne Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (`CURRENT SCHEMA`) qualifiziert.

##### TO *lokale-nachrichtendatei*

Gibt das Ziel für Warnungen und Fehlernachrichten an, die während der Ladeoperation auftreten. Diese Datei darf nicht mit der für den



Befehl LOAD angegebenen *nachrichtendatei* identisch sein. Wenn die Datei bereits vorhanden ist, werden die Informationen vom Dienstprogramm LOAD an sie angefügt.

### Beispiele

Ein Benutzer, der eine große Datenmenge in die Tabelle STAFF lädt, sollte den Status der Ladeoperation überprüfen. Der Benutzer kann folgendes angeben:

```
db2 connect to <datenbank>
db2 load query table staff to /u/mydir/staff.tempsg
```

Die Ausgabedatei /u/mydir/staff.tempsg könnte folgendermaßen aussehen:

```
SQL3500W Die Phase "LOAD" wird gestartet (Zeit:
"02-13-1997 19:40:29.645353").
```

```
SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler
für Eingabesätze: "0".
```

```
SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.
```

```
SQL3109N Das Dienstprogramm beginnt mit dem Laden von Daten aus
der Datei "/u/mydir/data/staffbig.ixf".
```

```
SQL3150N Der H-Satz in der PC/IXF-Datei enthält das Produkt "DB2 01.00",
das Datum "19970111" und die Uhrzeit "194554".
```

```
SQL3153N Der T-Satz in der PC/IXF-Datei hat den Namen "data/staffbig.ixf",
das Qualifikationsmerkmal " " und die Quelle " ".
```

```
SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler
für Eingabesätze: "111152".
```

```
SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.
```

```
SQL3519W Synchronisationspunkt am Beginn des Ladevorgangs. Zähler
für Eingabesätze: "222304".
```

```
SQL3520W Synchronisationspunkt für Ladevorgang erfolgreich.
```

### Siehe auch

„LOAD - Befehl“ auf Seite 101.

### C-API-Syntax

```
/* File: sqlutil.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
sqluload (
    sqlu_media_list * pDataFileList,
    sqlu_media_list * pLobPathList,
    struct sqldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pLocalMsgFileName,
    char * pRemoteMsgFileName,
    short CallerAction,
    struct sqluload_in * pLoadInfoIn,
    struct sqluload_out * pLoadInfoOut,
    sqlu_media_list * pWorkDirectoryList,
    sqlu_media_list * pCopyTargetList,
    sqlint32 * pNullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */
```

## Generische API-Syntax

```

/* File: sqlutil.h */
/* API: Load */
/* ... */
SQL_API_RC SQL_API_FN
sqlgload (
    unsigned short FileTypeLen,
    unsigned short LocalMsgFileNameLen,
    unsigned short RemoteMsgFileNameLen,
    sqlu_media_list * pDataFileList,
    sqlu_media_list * pLobPathList,
    struct sqlldcol * pDataDescriptor,
    struct sqlchar * pActionString,
    char * pFileType,
    struct sqlchar * pFileTypeMod,
    char * pLocalMsgFileName,
    char * pRemoteMsgFileName,
    short CallerAction,
    struct sqluload_in * pLoadInfoIn,
    struct sqluload_out * pLoadInfoOut,
    sqlu_media_list * pWorkDirectoryList,
    sqlu_media_list * pCopyTargetList,
    sqlint32 * pNullIndicators,
    void * pReserved,
    struct sqlca * pSqlca);
/* ... */

```

## API-Parameter

### FileTypeLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Dateityps in Byte darstellt.

### LocalMsgFileNameLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Namens der lokalen Nachrichtendatei in Byte darstellt.

### RemoteMsgFileNameLen

Eingabe. Eine 2 Byte große ganze Zahl ohne Vorzeichen, die die Länge des Pfadnamens für temporäre Dateien in Byte darstellt.

### pDataFileList

Eingabe. Ein Zeiger auf eine Struktur *sqlu\_media\_list*, die verwendet wird, um eine Liste mit Quelldateien, Einheiten, Lieferanten oder Pipes bereitzustellen. Ein Band wird unter OS/2 nicht unterstützt.

Die in dieser Struktur bereitgestellten Daten hängen vom Wert des Feldes *media\_type* ab. Gültige Werte (definiert in *sqlutil*) sind:

### SQLU\_SERVER\_LOCATION

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, stellt der Aufrufer Daten über die Strukturen *sqlu\_location\_entry* bereit.

Das Feld *sessions* gibt die Anzahl der bereitgestellten Strukturen *sqlu\_location\_entry* an. Es wird für Dateien, Einheiten und benannte Pipes verwendet.

### SQLU\_CLIENT\_LOCATION

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, stellt der Aufrufer Daten über die Strukturen *sqlu\_location\_entry* bereit. Das Feld *sessions* gibt die Anzahl der bereitgestellten Strukturen *sqlu\_location\_entry* an. Es wird für vollständig qualifizierte Dateien und benannte Pipes verwendet. Es ist zu beachten, daß dieses Feld *media\_type* nur dann gültig ist, wenn die API über einen Client mit Fernverbindung aufgerufen wird.

### SQLU\_TSM\_MEDIA

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, wird die Struktur *sqlu\_vendor* verwendet, wobei *filename* die eindeutige Kennung für die zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der zu startenden TSM-Sitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

### SQLU\_OTHER\_MEDIA

Wenn das Feld *media\_type* auf diesen Wert gesetzt ist, wird die Struktur *sqlu\_vendor* verwendet, wobei *shr\_lib* der Name der gemeinsam benutzten Bibliothek und *filename* die eindeutige Kennung für die zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der übrigen einzuleitenden Lieferantensitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

Wenn ein Dateiname angegeben wird, muß er stets vollständig qualifiziert sein. Weitere Informationen finden Sie unter „SQLU-MEDIA-LIST“ im Handbuch *Administrative API Reference*.

### pLobPathList

Eingabe. Ein Zeiger auf eine Struktur *sqlu\_media\_list*. Für die Dateitypen IXF, ASC und DEL eine Liste mit vollständig qualifizierten Pfaden oder Einheiten, die die Speicherposition der einzelnen zu ladenden LOB-Dateien angeben. Die Dateinamen befinden sich in den IXF-, ASC- oder DEL-Dateien und werden an die bereitgestellten Pfade angefügt. Ein Band wird unter OS/2 nicht unterstützt.

Die in dieser Struktur bereitgestellten Daten hängen vom Wert des Feldes *media\_type* ab. Gültige Werte (definiert in *sqlutil*) sind:

**SQLU\_LOCAL\_MEDIA**

Bei diesem Wert stellt der Aufrufer Daten über die Strukturen *sqlu\_media\_entry* bereit. Das Feld *sessions* gibt die Anzahl der bereitgestellten Strukturen *sqlu\_media\_entry* an.

**SQLU\_TSM\_MEDIA**

Bei diesem Wert wird die Struktur *sqlu\_vendor* verwendet, wobei *filename* die eindeutige Kennung für die zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der zu startenden TSM-Sitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folge-nummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

**SQLU\_OTHER\_MEDIA**

Bei diesem Wert wird die Struktur *sqlu\_vendor* verwendet, wobei *shr\_lib* der Name der gemeinsam benutzten Bibliothek und *filename* die eindeutige Kennung der zu ladenden Daten ist. Es darf nur einen *sqlu\_vendor*-Eintrag geben, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der übrigen einzuleitenden Lieferantensitzungen an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folge-nummern, jedoch mit denselben Daten in dem einen *sqlu\_vendor*-Eintrag.

Weitere Informationen finden Sie unter „SQLU-MEDIA-LIST“ im Handbuch *Administrative API Reference*.

**pDataDescriptor**

Eingabe. Zeiger auf eine Struktur *sqldcol*, die Informationen zu den Spalten enthält, die zum Laden aus der externen Datei ausgewählt werden.

Wenn der Parameter *pFileType* auf *SQL\_ASC* gesetzt ist, muß das Feld *dcolmeth* dieser Struktur auf *SQL\_METH\_L* gesetzt werden. Der Benutzer gibt die Anfangs- und Endposition für jede zu ladende Spalte an.

Wenn der Dateityp *SQL\_DEL* ist, kann *dcolmeth* entweder *SQL\_METH\_P* oder *SQL\_METH\_D* sein. Bei *SQL\_METH\_P* muß der Benutzer die Quellspaltenposition angeben. Bei *SQL\_METH\_D* wird die erste Spalte in der Datei in die erste Spalte der Tabelle geladen usw.

Wenn der Dateityp *SQL\_IXF* ist, kann *dcolmeth* entweder *SQL\_METH\_P*, *SQL\_METH\_D* oder *SQL\_METH\_N* sein. Hier gelten die Regeln für *DEL*-Dateien, außer daß *SQL\_METH\_N* angibt, daß Dateispaltennamen in der Struktur *sqldcol* bereitgestellt werden sollen.

Weitere Informationen finden Sie unter „SQLDCOL“ im Handbuch *Administrative API Reference*.

### pActionString

Eingabe. Ein Zeiger auf eine *sqlchar*-Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine Maßnahme angeben, die sich auf die Tabelle auswirkt.

Der Zeichenbereich besitzt folgendes Format:

```
"INSERT|REPLACE|RESTART|TERMINATE  
INTO tname [(spaltenliste)]  
[DATA LINK SPECIFICATION datalink-spez]  
[FOR EXCEPTION a_tname]"
```

### INSERT

Fügt die importierten Daten zur Tabelle hinzu, ohne die vorhandenen Tabellendaten zu ändern.

### REPLACE

Löscht alle vorhandenen Daten aus der Tabelle und fügt die importierten Daten ein. Die Tabellendefinition und die Indexdefinitionen werden nicht verändert.

### RESTART

Startet eine zuvor unterbrochene Ladeoperation erneut. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand in der LOAD-, BUILD- oder DELETE-Phase ausgehend fortgesetzt.

### TERMINATE

Beendet eine zuvor unterbrochene Ladeoperation und macht die Operation bis zu dem Zeitpunkt, zu dem sie gestartet wurde, rückgängig, auch wenn Konsistenzzustände übergangen wurden. Die Statusangaben von betroffenen Tabellenbereichen werden in „Normal“ zurückgeändert, und alle Tabellenobjekte werden konsistent gemacht (Indexobjekte können als ungültig gekennzeichnet werden, so daß beim nächsten Zugriff automatisch eine Indexwiederherstellung erfolgt). Wenn sich die Tabellenbereiche, in denen die Tabelle enthalten ist, nicht im Status „Laden anstehend“ befinden, wirkt sich diese Option nicht auf den Status der Tabellenbereiche aus.

Mit der Option LOAD TERMINATE wird der Status „Sicherung anstehend“ für Tabellenbereiche nicht aufgehoben.

*tname* Der Name der Tabelle, in die die Daten geladen werden sollen. Diese Tabelle darf keine Systemtabelle oder deklarierte temporäre Tabelle sein. Es kann ein Aliasname, der vollständig qualifizierte Name oder ein Name ohne Qualifikationsmerkmal angegeben werden. Ein qualifizierter Tabellenname hat das Format *schema.tabellenname*. Wenn ein Name ohne

Qualifikationsmerkmal angegeben wird, wird die Tabelle mit dem aktuellen Schema (CURRENT SCHEMA) qualifiziert.

*(spaltenliste)*

Eine Liste der Tabellenspaltennamen, in die die Daten eingefügt werden sollen. Die Spaltennamen müssen durch Kommas getrennt werden. Wenn ein Name Leerzeichen oder Kleinbuchstaben enthält, muß er in Anführungszeichen gesetzt werden.

**DATALINK SPECIFICATION** *datalink-spez*

Gibt Parameter an, die sich auf DB2 Data Links beziehen. Diese Parameter können mit der gleichen Syntax wie beim Befehl LOAD angegeben werden (siehe „LOAD - Befehl“ auf Seite 101).

**FOR EXCEPTION** *a\_tname*

Gibt die Ausnahmetabelle an, in die fehlerhafte Zeilen kopiert werden. Jede Zeile, die gegen einen eindeutigen Index oder einen Primärschlüsselindex verstößt, wird kopiert. DATALINK-Ausnahmen werden ebenfalls in der Ausnahmetabelle erfaßt.

**pFileType**

Eingabe. Eine Zeichenfolge, die das Format der Daten in der externen Datei angibt. Unterstützte externe Dateiformate (definiert in `sqlutil`) sind:

**SQL\_ASC**

ASCII-Format mit universellen Zeilenbegrenzern.

**SQL\_DEL**

ASCII-Format ohne universelle Zeilenbegrenzer für den Datenaustausch mit dBase, BASIC und den IBM Personal Decision Series-Programmen und vielen anderen Datenbank- und Dateimanagern.

**SQL\_IXF**

PC-Version des Integrated Exchange Format. Dies ist das bevorzugte Format für den Export von Daten aus einer Tabelle, so daß sie später in dieselbe Tabelle oder eine andere Datenbankmanagertabelle geladen werden können.

Weitere Informationen zu Dateiformaten finden Sie in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

**pFileTypeMod**

Eingabe. Ein Zeiger auf eine Struktur, die ein 2-Byte-Langfeld enthält, gefolgt von einer Matrix mit Zeichen, die eine oder mehrere Verarbei-

tungsoptionen angeben. Wenn dieser Zeiger NULL ist oder die Struktur, auf die gezeigt wird, null Zeichen enthält, wird diese Aktion als Auswahl einer Standardspezifikation gewertet.

Nicht alle Optionen können mit allen unterstützten Dateitypen verwendet werden.

Weitere Informationen finden Sie unter „SQLCHAR“ im Handbuch *Administrative API Reference* und unter „Dateitypwerte (LOAD)“ auf Seite 140.

### **pLocalMsgFileName**

Eingabe. Eine Zeichenfolge, die den Namen einer logischen Datei enthält, in die Ausgabenachrichten geschrieben werden.

### **pRemoteMsgFileName**

Eingabe. Eine Zeichenfolge, die den Pfadnamen enthält, der auf dem Server für temporäre Dateien verwendet werden soll. Temporäre Dateien werden erstellt, um Nachrichten, Konsistenzzustände und Informationen zu DELETE-Phasen zu speichern. Weitere Informationen zu temporären Dateien finden Sie in „Temporäre Dateien (LOAD)“ auf Seite 158.

### **CallerAction**

Eingabe. Eine vom Aufrufenden angeforderte Aktion. Gültige Werte (definiert in `sqlutil`) sind:

#### **SQLU\_INITIAL**

Erstaufruf. Dieser Wert (oder `SQLU_NOINTERRUPT`) muß beim ersten Aufruf der API verwendet werden.

#### **SQLU\_NOINTERRUPT**

Erstaufruf. Verarbeitung nicht aussetzen. Dieser Wert (oder `SQLU_INITIAL`) muß beim ersten Aufruf der API verwendet werden.

Wenn der Erstaufruf oder ein nachfolgender Aufruf zurückkehrt und erfordert, daß die aufrufende Anwendung eine Aktion ausführt, bevor die angeforderte Ladeoperation ausgeführt wird, muß die Aufruferaktion auf eine der folgenden Optionen gesetzt werden:

#### **SQLU\_CONTINUE**

Verarbeitung fortsetzen. Dieser Wert kann nur bei nachfolgenden Aufrufen der API verwendet werden, nachdem der Erstaufruf zurückgegeben wurde und das Dienstprogramm eine Benutzereingabe anfordert (z. B. um auf eine Bandendebingung zu reagieren). Er gibt an, daß die vom Dienstprogramm angeforderte Benutzeraktion beendet wurde und das Dienstprogramm die Verarbeitung der Erstanforderung fortsetzen kann.



**SQLU\_TERMINATE**

Verarbeitung beenden. Bewirkt, daß das Dienstprogramm LOAD vorzeitig beendet wird, wobei die zu ladenden Tabellenbereiche im Status LOAD\_PENDING verbleiben. Diese Option sollte angegeben werden, wenn keine weitere Verarbeitung der Daten erfolgen soll.

**SQLU\_ABORT**

Verarbeitung beenden. Bewirkt, daß das Dienstprogramm LOAD vorzeitig beendet wird, wobei die zu ladenden Tabellenbereiche im Status LOAD\_PENDING verbleiben. Diese Option sollte angegeben werden, wenn keine weitere Verarbeitung der Daten erfolgen soll.

**SQLU\_RESTART**

Verarbeitung erneut starten.

**SQLU\_DEVICE\_TERMINATE**

Eine einzelne Einheit beenden. Diese Option sollte angegeben werden, wenn das Dienstprogramm das Lesen von Daten von der Einheit stoppen, jedoch eine weitere Verarbeitung der Daten erfolgen soll.

**pLoadInfoIn**

Eingabe. Wahlfreier Zeiger auf die Struktur *sqluload\_in*, die zusätzliche Eingabeparameter enthält. Weitere Informationen zu dieser Struktur finden Sie in „Datenstruktur SQLULOAD-IN“ auf Seite 128.

**pLoadInfoOut**

Ausgabe. Wahlfreier Zeiger auf die Struktur *sqluload\_out*, die zusätzliche Ausgabeparameter enthält. Weitere Informationen zu dieser Struktur finden Sie in „Datenstruktur SQLULOAD-OUT“ auf Seite 133.

**pWorkDirectoryList**

Reserviert.

**pCopyTargetList**

Eingabe. Ein Zeiger auf eine Struktur *sqlu\_media\_list*, die (wenn ein Kopierabbild erstellt werden soll) verwendet wird, um eine Liste mit Zielpfaden, Einheiten oder eine gemeinsam benutzte Bibliothek bereitzustellen, in die das Kopierabbild geschrieben werden soll.

Die in dieser Struktur bereitgestellten Werte hängen vom Wert des Feldes *media\_type* ab. Gültige Werte für dieses Feld (definiert in `sqlutil`) sind:

**SQLU\_LOCAL\_MEDIA**

Wenn die Kopie auf einen lokalen Datenträger geschrieben werden soll, muß *media\_type* auf diesen Wert gesetzt werden, und in den Strukturen *sqlu\_media\_entry* müssen Informationen

zu den Zielen bereitgestellt werden. Das Feld *sessions* gibt die Anzahl der bereitgestellten Strukturen *sqlu\_media\_entry* an.

### SQLU\_TSM\_MEDIA

Verwenden Sie diesen Wert, wenn die Kopie in Tivoli Storage Manager (TSM) geschrieben werden soll. Es sind keine weitere Informationen erforderlich.

### SQLU\_OTHER\_MEDIA

Wenn ein Produkt eines Lieferanten verwendet werden soll, müssen Sie diesen Wert verwenden und weitere Informationen über eine Struktur *sqlu\_vendor* bereitstellen. Setzen Sie das Feld *shr\_lib* dieser Struktur auf den Namen der gemeinsam benutzten Bibliothek des Produkts des Lieferanten. Geben Sie nur einen *sqlu\_vendor*-Eintrag an, unabhängig vom Wert von *sessions*. Das Feld *sessions* gibt die Anzahl der bereitgestellten Strukturen *sqlu\_media\_entry* an. Das Dienstprogramm LOAD startet die Sitzungen mit unterschiedlichen Folgenummern, jedoch mit denselben Daten, die in dem einen *sqlu\_vendor*-Eintrag bereitgestellt werden.

Weitere Informationen finden Sie unter „SQLU-MEDIA-LIST“ im Handbuch *Administrative API Reference*.

### pNullIndicators

Eingabe. Nur für ASC-Dateien. Eine Matrix von ganzen Zahlen, die angeben, ob die Spaltendaten Nullwerte annehmen dürfen oder nicht. Es besteht eine 1-zu-1-Entsprechung zwischen den Elementen dieser Matrix und den Spalten, die aus der Datendatei geladen werden. Das heißt, die Anzahl der Elemente muß mit dem Feld *dcolnum* des Parameters *pDataDescriptor* übereinstimmen. Jedes Element der Matrix enthält eine Zahl, die eine Speicherposition in der Datendatei angibt, die als Nullanzeigerfeld verwendet werden soll, oder eine Null, um anzugeben, daß die Tabellenspalte keine Nullwerte enthalten darf. Wenn das Element nicht Null ist, muß die angegebene Speicherposition in der Datendatei Y oder N enthalten. Y gibt an, daß die Tabellenspaltendaten NULL sind, N gibt an, daß die Tabellenspaltendaten nicht NULL sind.

### pReserved

Zur zukünftigen Verwendung reserviert.

### pSqlca

Ausgabe. Ein Zeiger auf die Struktur *sqlca*. Weitere Informationen zu dieser Struktur finden Sie im Handbuch *Administrative API Reference* und im Handbuch *SQL Reference*.

**REXX-API-Syntax**

Diese API kann aus REXX über die Schnittstelle SQLDB2 aufgerufen werden. Weitere Informationen finden Sie im Handbuch *Application Development Guide*. Eine Beschreibung der Syntax finden Sie in „LOAD - Befehl“ auf Seite 101.

### Datenstruktur SQLULOAD-IN

Diese Struktur wird verwendet, um Informationen während eines Aufrufs an die LOAD-API zu übergeben (siehe „LOAD - API“ auf Seite 118).

Tabelle 6. Felder in der Struktur SQLULOAD-IN

Feldname	Datentyp	Beschreibung
SIZEOFSTRUCT	sqluint32	Größe der Struktur in Byte.
SAVECNT	sqluint32	<p>Die Anzahl der Datensätze, die geladen werden sollen, bevor ein Konsistenzzustand definiert wird. Dieser Wert wird in einen Seitenzähler umgesetzt und auf Intervalle in der Größe des Speicherbereichs gerundet. Da bei jedem Konsistenzzustand eine Nachricht abgesetzt wird, sollte diese Option ausgewählt werden, wenn die Ladeoperation mit einem LOAD QUERY-Befehl überwacht wird (siehe „db2LoadQuery - API Load Query“ auf Seite 135. Wenn der Wert von <i>savecnt</i> nicht groß genug ist, wirkt sich die Synchronisation der Aktivitäten, die bei jedem Konsistenzzustand ausgeführt werden, nachteilig auf die Leistung aus.</p> <p>Der Standardwert ist 0, d. h., daß Konsistenzzustände nur definiert werden, sofern dies notwendig ist.</p>
RESTARTCNT	sqluint32G	Reserviert.
ROWCNT	sqluint32	Die Anzahl der physischen Sätze, die geladen werden sollen. Ermöglicht es einem Benutzer, nur die ersten <i>rowcnt</i> Zeilen in einer Datei zu laden.
WARNINGCNT	sqluint32	<p>Stoppt die Ladeoperation nach <i>warningcnt</i> Warnungen. Definieren Sie diesen Parameter, wenn keine Warnungen erwartet werden, jedoch eine Prüfung darauf, ob die richtige Datei und Tabelle verwendet werden, gewünscht wird. Wenn <i>warningcnt</i> den Wert 0 hat oder diese Option nicht angegeben wird, wird die Ladeoperation unabhängig von der Anzahl der abgesetzten Warnungen fortgesetzt.</p> <p>Wenn die Ladeoperation gestoppt wird, da die Schwelle für Warnungen überschritten wurde, kann eine weitere Ladeoperation im RESTART-Modus gestartet werden. Die Ladeoperation wird automatisch vom letzten Konsistenzzustand ausgehend fortgesetzt. Alternativ kann eine weitere Ladeoperation im REPLACE-Modus eingeleitet werden, die am Anfang der Eingabedatei beginnt.</p>

Tabelle 6. Felder in der Struktur SQLLOAD-IN (Forts.)

Feldname	Datentyp	Beschreibung
DATA_BUFFER_SIZE	sqluint32	<p>Die Anzahl der 4 KB-Seiten (unabhängig vom Grad der Parallelität), die als Pufferbereich zur Übertragung von Daten im Dienstprogramm verwendet werden sollen. Wenn der angegebene Wert kleiner als der algorithmische Mindestwert ist, wird der erforderliche Mindestwert verwendet, und es wird keine Warnung zurückgegeben.</p> <p>Dieser Speicher wird direkt aus dem Zwischenspeicher der Dienstprogramme zugeordnet, dessen Größe mit dem Datenbankkonfigurationsparameter <i>util_heap_sz</i> geändert werden kann.</p> <p>Wenn kein Wert angegeben wird, wird vom Dienstprogramm zur Laufzeit ein sinnvoller Standardwert berechnet. Der Standardwert basiert auf einem Prozentsatz des freien Speicherbereichs, der bei der Exemplarerstellung des Ladeprogramms im Zwischenspeicher der Dienstprogramme verfügbar ist, sowie auf einigen Merkmalen der Tabelle.</p>
SORT_BUFFER_SIZE	sqluint32	Reserviert.
HOLD QUIESCE	UNSIGNED SHORT	Eine Markierung, deren Wert auf TRUE gesetzt wird, wenn das Dienstprogramm die Tabelle nach dem Laden im Status „Im Wartemodus: EXCLUSIVE“ belassen soll, bzw. auf FALSE, wenn das nicht erwünscht ist.
RESTARTPHASE	CHAR(1)	Reserviert.
STATSOPT	CHAR(1)	Granularität der zu sammelnden Statistikdaten. Die Werte sind unten ersichtlich.
CPU_PARALLELISM	UNSIGNED SHORT	<p>Die Anzahl der Prozesse oder Threads, die vom Dienstprogramm LOAD gestartet werden, um Datensätze beim Erstellen von Tabellenobjekten syntaktisch zu analysieren, umzusetzen und zu formatieren. Dieser Parameter soll die Nutzung der partitionsinternen Parallelität ermöglichen. Er ist besonders nützlich beim Laden von vorsortierten Daten, weil die Satzreihenfolge der Quelldaten beibehalten wird. Wenn der Wert dieses Parameters Null ist, verwendet das Dienstprogramm LOAD zur Laufzeit einen sinnvollen Standardwert.</p> <p><b>Anmerkung:</b> Wenn dieser Parameter bei Tabellen verwendet wird, die LOB- oder LONG VARCHAR-Felder enthalten, wird sein Wert Eins, unabhängig von der Anzahl der System-CPU's oder dem vom Benutzer angegebenen Wert.</p>

## Datenstruktur SQLULOAD-IN

Tabelle 6. Felder in der Struktur SQLULOAD-IN (Forts.)

Feldname	Datentyp	Beschreibung
DISK_PARALLELISM	UNSIGNED SHORT	Die Anzahl der Prozesse oder Threads, die vom Dienstprogramm LOAD gestartet werden, um Daten in die Tabellenbereichsbehälter zu schreiben. Wenn kein Wert angegeben wird, wählt das Dienstprogramm einen sinnvollen Standardwert auf Basis der Anzahl der Tabellenbereichsbehälter und der Merkmale der Tabelle aus.
NON_RECOVERABLE	UNSIGNED SHORT	<p>Auf <code>SQLU_NON_RECOVERABLE_LOAD</code> setzen, um festzulegen, daß eine Ladetransaktion als nicht wiederherstellbar gekennzeichnet werden soll und daß es nicht möglich sein wird, sie durch eine nachfolgende aktualisierende Wiederherstellung wiederherzustellen. Das Dienstprogramm zur aktualisierenden Wiederherstellung überspringt die Transaktion und kennzeichnet die Tabelle, in die die Daten geladen wurden, als „ungültig“. Vom Dienstprogramm werden auch alle nachfolgenden Transaktionen für diese Tabelle ignoriert. Nach Beendigung der aktualisierenden Wiederherstellung kann eine solche Tabelle nur gelöscht werden.</p> <p>Mit dieser Option werden Tabellenbereiche nach der Ladeoperation nicht in den Status „Sicherheit anstehend“ gesetzt, und während der Ladeoperation muß keine Kopie der geladenen Daten erstellt werden.</p> <p>Auf <code>SQLU_RECOVERABLE_LOAD</code> setzen, wenn die Ladetransaktion als wiederherstellbar gekennzeichnet werden soll.</p>
INDEXING_MODE	UNSIGNED SHORT	Gibt an, ob das Dienstprogramm LOAD Indizes erneut erstellen oder sie inkrementell erweitern soll. Die Werte sind unten ersichtlich.

Gültige Werte für *STATSOPT* (definiert in `sqlutil`) sind:

**SQLU\_STATS\_NONE**

**SQL\_STATS\_EXTTABLE\_ONLY**

**SQL\_STATS\_EXTTABLE\_INDEX**

**SQL\_STATS\_INDEX**

**SQL\_STATS\_TABLE**

**SQL\_STATS\_EXTINDEX\_ONLY**

**SQL\_STATS\_EXTINDEX\_TABLE**

**SQL\_STATS\_ALL**

**SQL\_STATS\_BOTH**

Gültige Werte für *INDEXING\_MODE* (definiert in `sqlutil`) sind:

**SQLU\_INX\_AUTOSELECT**

**SQLU\_INX\_REBUILD**

**SQLU\_INX\_INCREMENTAL**

**SQLU\_INX\_DEFERRED**

Eine Erläuterung dieser Indexierungsmodi können Sie unter „LOAD - Befehl“ auf Seite 101 nachlesen.

# Datenstruktur SQLULOAD-IN

## Sprachsyntax

### C-Struktur

```
/* File: sqlutil.h */
/* Structure: SQLULOAD-IN */
/* ... */
SQL_STRUCTURE sqluload_in
{
    sqluint32      sizeofStruct;
    sqluint32      savecnt;
    sqluint32      restartcnt;
    sqluint32      rowcnt;
    sqluint32      warningcnt;
    sqluint32      data_buffer_size;
    sqluint32      sort_buffer_size; /* No longer used. */
    unsigned short hold_quiesce;
    char           restartphase;
    char           statsopt;
    unsigned short cpu_parallelism;
    unsigned short disk_parallelism;
    unsigned short non_recoverable;
    unsigned short indexing_mode;
};
/* ... */
```

### COBOL-Struktur

```
* File: sqlutil.cbl
01 SQLULOAD-IN.
   05 SQL-SIZE-OF-STRUCT      PIC 9(9) COMP-5 VALUE 40.
   05 SQL-SAVECNT            PIC 9(9) COMP-5.
   05 SQL-RESTARTCOUNT     PIC 9(9) COMP-5.
   05 SQL-ROWCNT            PIC 9(9) COMP-5.
   05 SQL-WARNINGCNT        PIC 9(9) COMP-5.
   05 SQL-DATA-BUFFER-SIZE  PIC 9(9) COMP-5.
   05 SQL-SORT-BUFFER-SIZE  PIC 9(9) COMP-5. * No longer used.
   05 SQL-HOLD-QUIESCE      PIC 9(4) COMP-5.
   05 SQL-RESTARTPHASE     PIC X.
   05 SQL-STATSOPT         PIC X.
   05 SQL-CPU-PARALLELISM  PIC 9(4) COMP-5.
   05 SQL-DISK-PARALLELISM PIC 9(4) COMP-5.
   05 SQL-NON-RECOVERABLE  PIC 9(4) COMP-5.
   05 SQL-INDEXING-MODE    PIC 9(4) COMP-5.
*
```



Datenstruktur SQLLLOAD-OUT

Diese Struktur wird verwendet, um Informationen nach einem Aufruf an die LOAD-API zu übergeben (siehe „LOAD - API“ auf Seite 118).

Tabelle 7. Felder in der Struktur SQLLLOAD-OUT

Feldname	Datentyp	Beschreibung
SIZEOFSTRUCT	sqluint32	Größe der Struktur in Byte.
ROWSREAD	sqluint32	Anzahl der während der Ladeoperation gelesenen Datensätze
ROWSSKIPPED	sqluint32	Anzahl der Datensätze, die vor Beginn der Ladeoperation übersprungen wurden.
ROWSLOADED	sqluint32	Anzahl der Zeilen, die in die Zieltabelle geladen wurden.
ROWSREJECTED	sqluint32	Anzahl der Datensätze, die nicht geladen werden konnten.
ROWSDELETED	sqluint32	Anzahl der gelöschten doppelten Datensätze
ROWSCOMMITTED	sqluint32	Die Gesamtzahl der verarbeiteten Datensätze: die Anzahl der erfolgreich in die Datenbank geladenen und dort festgeschriebenen Datensätze und zusätzlich die Anzahl der übersprungenen und zurückgewiesenen Datensätze.

Sprachsyntax

C-Struktur

```

/* File: sqlutil.h */
/* Structure: SQLLLOAD-OUT */
/* ... */
SQL_STRUCTURE sqlload_out
{
    sqluint32    sizeofStruct;
    sqluint32    rowsRead;
    sqluint32    rowsSkipped;
    sqluint32    rowsLoaded;
    sqluint32    rowsRejected;
    sqluint32    rowsDeleted;
    sqluint32    rowsCommitted;
};
/* ... */

```

## Datenstruktur SQLLOAD-OUT

### COBOL-Struktur

```
* File: sqlutil.cbl
01 SQLLOAD-OUT.
   05 SQL-SIZE-OF-STRUCT      PIC 9(9) COMP-5 VALUE 28.
   05 SQL-ROWS-READ          PIC 9(9) COMP-5.
   05 SQL-ROWS-SKIPPED       PIC 9(9) COMP-5.
   05 SQL-ROWS-LOADED        PIC 9(9) COMP-5.
   05 SQL-ROWS-REJECTED      PIC 9(9) COMP-5.
   05 SQL-ROWS-DELETED       PIC 9(9) COMP-5.
   05 SQL-ROWS-COMMITTED     PIC 9(9) COMP-5.
*
```

**db2LoadQuery - API Load Query**

Überprüft den Status einer Ladeoperation während der Verarbeitung.

**Berechtigung**

Keine

**Erforderliche Verbindung**

Datenbank

**API-Include-Datei**

*db2ApiDf.h*

**C-API-Syntax**

```

/* File: db2ApiDf.h */
/* API: Load Query */
/* ... */
SQL_API_RC SQL_API_FN
db2LoadQuery (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca *pSqlca);

typedef struct
{
    db2UInt32 iStringType;
    char * piString;
    db2UInt32 iShowLoadMessages;
    db2LoadQueryOutputStruct * poOutputStruct;
    char * piLocalMessageFile;
} db2LoadQueryStruct;

typedef struct
{
    db2UInt32 oRowsRead;
    db2UInt32 oRowsSkipped;
    db2UInt32 oRowsCommitted;
    db2UInt32 oRowsLoaded;
    db2UInt32 oRowsRejected;
    db2UInt32 oRowsDeleted;
    db2UInt32 oCurrentIndex;
    db2UInt32 oNumTotalIndexes;
    db2UInt32 oCurrentMPPNode;
    db2UInt32 oLoadRestarted;
    db2UInt32 oWhichPhase;
    db2UInt32 oWarningCount;
} db2LoadQueryOutputStruct;
/* ... */

```

## db2LoadQuery - API Load Query

### Generische API-Syntax

```
/* File: db2ApiDf.h */
/* API: Load Query */
/* ... */
SQL_API_RC SQL_API_FN
db2gLoadQuery (
    db2UInt32 versionNumber,
    void * pParmStruct,
    struct sqlca *pSqlca);

typedef struct
{
    db2UInt32 iStringType;
    db2UInt32 iStringLen;
    char * piString;
    db2UInt32 iShowLoadMessages;
    db2LoadQueryOutputStruct * poOutputStruct;
    db2UInt32 iLocalMessageFileLen;
    char * piLocalMessageFile
} db2gLoadQueryStruct;

typedef struct
{
    db2UInt32 oRowsRead;
    db2UInt32 oRowsSkipped;
    db2UInt32 oRowsCommitted;
    db2UInt32 oRowsLoaded;
    db2UInt32 oRowsRejected;
    db2UInt32 oRowsDeleted;
    db2UInt32 oCurrentIndex;
    db2UInt32 oNumTotalIndexes;
    db2UInt32 oCurrentMPPNode;
    db2UInt32 oLoadRestarted;
    db2UInt32 oWhichPhase;
    db2UInt32 oWarningCount;
} db2LoadQueryOutputStruct;
/* ... */
```

### API-Parameter

#### versionNumber

Eingabe. Gibt den Versions- und Release-Stand der Struktur an, die als zweiter Parameter, *pParmStruct*, übergeben wird.

#### pParmStruct

Eingabe. Ein Zeiger auf die Struktur *db2LoadQueryStruct*.

#### pSqlca

Ausgabe. Ein Zeiger auf die Struktur *sqlca*. Weitere Informationen zu dieser Struktur finden Sie im Handbuch *Administrative API Reference* und im Handbuch *SQL Reference*.

### **iStringType**

Eingabe. Gibt einen Typ für *piString* an. Gültige Werte (definiert in *db2ApiDf.h*) sind:

#### **DB2LOADQUERY\_TABLENAME**

Steht für die Angabe eines Tabellennamens, der von der API **db2LoadQuery** verwendet wird.

### **iStringLen**

Eingabe. Gibt die Länge von *piString* in Byte an.

### **piString**

Eingabe. Gibt abhängig vom Wert von *iStringType* einen Pfadnamen für temporäre Dateien oder einen Tabellennamen an.

### **iShowLoadMessages**

Eingabe. Gibt die Stufe der Nachrichten an, die vom Dienstprogramm LOAD zurückgegeben werden. Gültige Werte (definiert in *db2ApiDf.h*) sind:

#### **DB2LOADQUERY\_SHOW\_ALL\_MSGS**

Alle LOAD-Nachrichten zurückgeben.

#### **DB2LOADQUERY\_SHOW\_NO\_MSGS**

Keine LOAD-Nachrichten zurückgeben.

#### **DB2LOADQUERY\_SHOW\_NEW\_MSGS**

Seit dem letzten Aufruf dieser API generierte Nachrichten zurückgeben.

### **poOutputStruct**

Ausgabe. Ein Zeiger auf die Struktur *db2LoadQueryOutputStruct*, die einen LOAD-Ergebnistext enthält. Auf NULL setzen, wenn keine Zusammenfassung erforderlich ist.

### **iLocalMessageFileLen**

Eingabe. Gibt die Länge von *piLocalMessageFile* in Byte an.

### **piLocalMessageFile**

Eingabe. Gibt den Namen der lokalen Datei an, die für Ausgabenachrichten verwendet werden soll.

### **oRowsRead**

Ausgabe. Anzahl der Datensätze, die bislang vom Dienstprogramm LOAD gelesen wurden.

### **oRowsSkipped**

Ausgabe. Anzahl der Datensätze, die vor Beginn der Ladeoperation übersprungen wurden.

## db2LoadQuery - API Load Query

### **oRowsCommitted**

Ausgabe. Anzahl der Zeilen, die bislang in der Zieltabelle festgeschrieben wurden.

### **oRowsLoaded**

Ausgabe. Anzahl der Zeilen, die bislang in die Zieltabelle geladen wurden.

### **oRowsRejected**

Ausgabe. Anzahl der Zeilen, die bislang von der Zieltabelle zurückgewiesen wurden.

### **oRowsDeleted**

Ausgabe. Anzahl der Zeilen, die bislang aus der Zieltabelle gelöscht wurden (in der DELETE-Phase).

### **oCurrentIndex**

Ausgabe. Index, der gerade erstellt wird (während der BUILD-Phase).

### **oCurrentMPPNode**

Ausgabe. Gibt an, welcher Knoten abgefragt wird (nur für MPP-Modus).

### **oLoadRestarted**

Ausgabe. Eine Markierung, deren Wert TRUE ist, wenn die abgefragte Ladeoperation eine LOAD RESTART-Operation ist.

### **oWhichPhase**

Ausgabe. Gibt die aktuelle Phase der abgefragten Ladeoperation an. Gültige Werte (definiert in db2ApiDf.h) sind:

**DB2LOADQUERY\_LOAD\_PHASE**

LOAD-Phase

**DB2LOADQUERY\_BUILD\_PHASE**

BUILD-Phase

**DB2LOADQUERY\_DELETE\_PHASE**

DELETE-Phase.

### **oNumTotalIndexes**

Ausgabe. Gesamtzahl der zu erstellenden Indizes (während der BUILD-Phase).

### **oWarningCount**

Ausgabe. Gesamtzahl der bislang zurückgegebenen Warnungen.

### REXX-API-Syntax

Diese API kann aus REXX über die Schnittstelle SQLDB2 aufgerufen werden. Weitere Informationen finden Sie im Handbuch *Application Development Guide*. Eine Beschreibung der Syntax finden Sie in „LOAD QUERY - Befehl“ auf Seite 116.

### Beispielprogramme

C                    \sqllib\samples\c\loadqry.sqc

COBOL                \sqllib\samples\cobol\loadqry.sqb

### Anmerkungen zur Verwendung

Diese API liest den Status der Ladeoperation für die Tabelle, die mit *piString* angegeben ist, und schreibt den Status in die Datei, die mit *pLocalMsgFileName* angegeben ist.

## Dateitypwerte (LOAD)

---

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD)

Wert	Beschreibung
<b>Alle Dateiformate</b>	
anyorder	Dieser Wert wird zusammen mit dem Parameter <i>cpu_parallelism</i> verwendet. Er gibt an, daß die Beibehaltung der Quellendatenreihenfolge nicht erforderlich ist, was zu erheblichen Leistungssteigerungen auf SMP-Systemen führt. Wenn der Wert von <i>cpu_parallelism</i> 1 ist, wird diese Option ignoriert. Diese Option wird nicht unterstützt, wenn <i>SAVECOUNT</i> > 0, da die Wiederherstellung nach einem Systemabsturz nach einem Konsistenzzustand erfordert, daß die Daten geordnet geladen werden.
fastparse	<p>Für vom Benutzer angegebene Spaltenwerte erfolgt eine geringere Syntaxprüfung, und die Leistung wird verbessert. Tabellen, die mit dieser Option geladen werden, besitzen einen richtigen Aufbau, und das Dienstprogramm führt eine hinreichende Datenprüfung durch, um eine Segmentierungsverletzung oder einen Trap-Fehler zu verhindern. Daten im richtigen Format werden auch richtig geladen.</p> <p>Beispiel: Wenn der Wert 123qwr4 als Feldeintrag für eine INTEGER-Spalte in einer ASC-Datei vorkommt, markiert das Dienstprogramm normalerweise einen Syntaxfehler, da dieser Wert keine gültige Zahl darstellt. Bei Angabe von fastparse wird kein Syntaxfehler erkannt, und in das INTEGER-Feld wird eine Zufallszahl geladen. Dieser Wert darf daher nur für fehlerfreie Daten verwendet werden. Die Leistungsverbesserung bei Verwendung dieser Option für ASCII-Daten kann erheblich sein. fastparse führt jedoch zu keiner wesentlichen Leistungssteigerung bei PC/IXF-Daten, da IXF ein Binärformat ist und sich fastparse auf die syntaktische Analyse und Umwandlung von ASCII in interne Formate auswirkt.</p>



Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
generatedignore	Dieser Wert informiert das Dienstprogramm LOAD darüber, daß für alle generierten Spalten in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Bei Spalten, die Nullwerte enthalten dürfen, bewirkt dies, daß Nullwerte in die Spalte geladen werden. Bei Spalten, die keine Nullwerte enthalten dürfen, führt dieser Wert dazu, daß der Standardwert für den Datentyp der generierten Spalte geladen wird. Am Ende der Ladeoperation kann die Anweisung SET INTEGRITY aufgerufen werden, um das Ersetzen geladener Werte durch Werte zu erzwingen, die gemäß der Definition der generierten Spalte berechnet werden. Dieser Wert kann nicht zusammen mit dem Wert generatedmissing oder dem Wert generatedoverride verwendet werden.
generatedmissing	Wird dieser Wert angegeben, geht das Dienstprogramm davon aus, daß die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die generierte Spalte enthält, und lädt daher Nullwerte in die Spalte. Am Ende der Ladeoperation können die Nullwerte mit der Anweisung SET INTEGRITY durch Werte ersetzt werden, die gemäß der Definition der generierten Spalte berechnet werden. Dieser Wert kann nicht zusammen mit dem Wert generatedignore oder dem Wert generatedoverride verwendet werden.
generatedoverride	Dieser Wert weist das Dienstprogramm LOAD an (entgegen der normalen Regeln für diese Spaltentypen), explizite Datenwerte, die nicht Null sind, für alle generierten Spalten in der Tabelle zu akzeptieren. Dies ist bei der Migration von Daten aus einem anderen Datenbanksystem oder beim Laden einer Tabelle aus Daten nützlich, die mit der Option DROPPED TABLE RECOVERY des Befehls ROLLFORWARD DATABASE wiederhergestellt wurden. Bei Verwendung dieses Werts werden alle Zeilen ohne Daten oder mit Nulldaten für eine generierte Spalte, die keine Nullwerte enthalten darf, zurückgewiesen (SQL3116W). <b>Anmerkung:</b> Wird diese Option verwendet, versucht das Dienstprogramm LOAD nicht, generierte Spaltenwerte auszuwerten.  Dieser Wert kann nicht zusammen mit dem Wert generatedmissing oder dem Wert generatedignore verwendet werden.

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
identityignore	Dieser Wert informiert das Dienstprogramm LOAD darüber, daß für die Identitätsspalte in der Datendatei Daten vorhanden sind, aber ignoriert werden sollen. Dies bewirkt, daß alle Identitätswerte durch das Dienstprogramm generiert werden. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dies bedeutet, daß bei als GENERATED ALWAYS definierten Spalten keine Zeilen zurückgewiesen werden. Dieser Wert kann nicht zusammen mit dem Wert identitymissing oder dem Wert identityoverride verwendet werden.
identitymissing	Wird dieser Wert angegeben, geht das Dienstprogramm davon aus, daß die Eingabedatendatei keine Daten (nicht einmal Nullwerte) für die Identitätsspalte enthält, und generiert daher für jede Zeile einen Wert. Auf diese Weise wird sowohl bei als GENERATED ALWAYS definierten Identitätsspalten als auch bei als GENERATED BY DEFAULT definierten Identitätsspalten verfahren. Dieser Wert kann nicht zusammen mit dem Wert identityignore oder dem Wert identityoverride verwendet werden.
identityoverride	Dieser Wert sollte nur dann verwendet werden, wenn die zu ladende Tabelle eine als GENERATED ALWAYS definierte Identitätsspalte enthält. Er weist das Dienstprogramm an, entgegen der normalen Regeln für diese Typen von Identitätsspalten für eine solche Spalte explizite Daten zu akzeptieren, die keine Nulldaten sind. Dies ist bei der Migration von Daten aus einem anderen Datenbanksystem, bei der die Tabelle als GENERATED ALWAYS definiert sein muß, oder beim Laden einer Tabelle aus Daten nützlich, die mit der Option DROPPED TABLE RECOVERY des Befehls ROLLFORWARD DATABASE wiederhergestellt wurden. Bei Verwendung dieses Werts werden alle Zeilen ohne Daten oder mit Nulldaten für die Identitätsspalte zurückgewiesen (SQL3116W). Dieser Wert kann nicht zusammen mit dem Wert identitymissing oder dem Wert identityignore verwendet werden. <b>Anmerkung:</b> Wird diese Option verwendet, versucht das Dienstprogramm LOAD nicht, die Eindeutigkeit der Werte in der Identitätsspalte der Tabelle zu verwalten oder zu prüfen.

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
indexfreespace=x	<p>x ist eine ganze Zahl zwischen 0 und 99. Der Wert wird als Prozentsatz jeder Indexseite gewertet, der beim Laden des Index als freier Speicherbereich freigelassen werden soll. Der erste Eintrag auf einer Seite wird ohne Einschränkung hinzugefügt. Weitere Einträge werden hinzugefügt, wenn die prozentuale Schwelle für freien Speicherbereich eingehalten werden kann. Der Standardwert ist der zum Zeitpunkt der Indexerstellung (CREATE INDEX) verwendete Wert.</p> <p>Dieser Wert hat Vorrang vor dem in der Anweisung CREATE INDEX angegebenen Wert für PCTFREE und wirkt sich nur auf Indexseiten aus.</p>
lobsinfile	<p><i>lob-pfad</i> gibt den Pfad zu den Dateien an, die LOB-Werte enthalten. Die ASC-, DEL- oder IXF-Eingabedateien für LOAD enthalten die Namen der Dateien, die LOB-Daten in den LOB-Spalten aufweisen.</p>
noheader	<p>Überspringt den Prüfcode für den Kopfbereich. (Dieser Wert kann nur beim Laden in Tabellen angewendet werden, die sich in einer Knotengruppe mit einem Knoten befinden.)</p> <p>Das Dienstprogramm AutoLoader (siehe „Kapitel 4. AutoLoader“ auf Seite 181) schreibt Kopfdaten in jede Datei, die Daten zu einer Tabelle in einer Knotengruppe mit mehreren Knoten liefert. Der Kennsatz enthält die Knotennummer, den Partitionierungsschlüssel und die Partitionierungsschlüsselspezifikation. Diese Informationen werden vom Dienstprogramm LOAD benötigt, um zu prüfen, ob die Daten am richtigen Knoten geladen werden. Beim Laden von Dateien in eine Tabelle, die in einer Knotengruppe mit einem Knoten vorhanden ist, sind keine Kennsätze vorhanden, und diese Option bewirkt, daß das Dienstprogramm LOAD den Kennsatzprüfungscode überspringt.</p>
norowwarnings	<p>Unterdrückt alle Warnungen zu zurückgewiesenen Zeilen.</p>

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
pagefreespace= <i>x</i>	<p><i>x</i> ist eine ganze Zahl zwischen 0 und 100 (einschließlich). Der Wert wird als Prozentsatz jeder Datenseite gewertet, der als freier Speicherbereich freigelassen werden soll.</p> <p>Wenn der angegebene Wert aufgrund der Mindestzeilengröße ungültig ist (z. B. bei einer Zeile, die mindestens 3 000 Byte lang ist und bei einem <i>x</i>-Wert von 50), wird die Zeile auf eine neue Seite gestellt. Wenn der Wert 100 angegeben wird, wird jede Zeile auf eine neue Seite gestellt.</p> <p><b>Anmerkung:</b> Der Wert PCTFREE legt die Menge des freien Speicherbereichs pro Seite fest. Wenn kein pagefreespace-Wert für die Ladeoperation oder kein PCTFREE-Wert für die Tabelle festgelegt wurde, füllt das Dienstprogramm so viel Speicherbereich wie möglich auf jeder Seite. Der mit pagefreespace festgelegte Wert überschreibt den für die Tabelle angegebenen PCTFREE-Wert.</p>
totalfreespace= <i>x</i>	<p><i>x</i> ist eine ganze Zahl zwischen 0 und 100 (einschließlich). Der Wert wird als Prozentsatz der Gesamtseiten in der Tabelle gewertet, der als freier Speicherbereich am Ende der Tabelle angefügt werden soll. Wenn Sie beispielsweise für <i>x</i> 20 angeben und die Tabelle 100 Datenseiten umfaßt, werden 20 zusätzliche leere Seiten angefügt. Die Gesamtzahl der Datenseiten für die Tabelle ist somit 120.</p>
usedefaults	<p>Wenn eine Quellenspalte für eine Zieltabellenspalte angegeben wurde, jedoch für mindestens ein Zeilenexemplar keine Daten enthält, werden Standardwerte geladen. Beispiele für fehlende Daten sind:</p> <ul style="list-style-type: none"> <li>• Für DEL-Dateien: ",," wird für die Spalte angegeben.</li> <li>• Für DEL/ASC/WSF-Dateien: Eine Zeile, die nicht genug Spalten enthält oder für die ursprüngliche Spezifikation nicht lang genug ist.</li> </ul> <p>Ohne diese Option geschieht folgendes, wenn eine Quellenspalte keine Daten für ein Zeilenexemplar enthält:</p> <ul style="list-style-type: none"> <li>• Wenn die Spalte Nullwerte enthalten darf, wird NULL geladen.</li> <li>• Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile vom Dienstprogramm zurückgewiesen.</li> </ul>
<b>ASCII-Dateiformate (ASC/DEL)</b>	

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
codepage=x	<p>x ist eine ASCII-Zeichenfolge. Der Wert wird als Codepage der Daten im Eingabedatensatz gewertet. Wandelt Zeichen- daten (und in Zeichenform angegebene numerische Daten) aus dieser Codepage während der Ladeoperation in die Codepage der Datenbank um.</p> <p>Dabei gelten die folgenden Regeln:</p> <ul style="list-style-type: none"> <li>• Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.</li> <li>• Bei DEL-Daten, die in einer EBCDIC-Codepage angegeben werden, dürfen die Begrenzer nicht mit den DBCS-Startzeichen und -Endezeichen identisch sein.</li> <li>• nullindchar muß Symbole angeben, die im Standard-ASCII-Satz zwischen den Codepunkten x20 und x7F (jeweils einschließlich) enthalten sind. Das gilt für ASCII-Symbole und -Codepunkte. Bei EBCDIC-Daten können die entsprechenden Symbole verwendet werden, auch wenn die Codepunkte anders sind.</li> </ul>
dateformat="x"	<p>x ist das Format des Datums in der Quelldatei.<sup>a</sup> Gültige Datumselemente sind:</p> <p>YYYY - Jahr (vier Ziffern von 0000 bis 9999)  M - Monat (eine oder zwei Ziffern von 1 bis 12)  MM - Monat (zwei Ziffern von 1 bis 12;  dieses Element und das Element M  schließen sich gegenseitig aus)  D - Tag (eine oder zwei Ziffern von 1 bis 31)  DD - Tag (zwei Ziffern von 1 bis 31; dieses Element  und das Element D schließen sich gegen-  seitig aus)</p> <p>DDD - Tag des Jahres (drei Ziffern von 001  bis 366; dieses Element und andere  Elemente für Tag oder Monat schließen  sich gegenseitig aus)</p> <p>Für jedes Element, das nicht angegeben wird, wird der  Standardwert 1 zugeordnet. Einige Beispiele für Datums-  formate:</p> <p>"D-M-YYYY"  "MM.DD.YYYY"  "YYYYDDD"</p>

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
dumpfile = x	<p>x ist der vollständig qualifizierte (bezüglich des Server-Knotens) Name einer Ausnahmedatei, in die zurückgewiesene Zeilen geschrieben werden. Pro Satz werden maximal 32 KB Daten geschrieben. Es folgt ein Beispiel, das zeigt, wie eine Speicherauszugsdatei angegeben wird:</p> <pre>db2 load from data of del   modified by dumpfile = /u/user/dateiname   insert into tabellenname</pre> <p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"><li>1. In einer Umgebung mit partitionierten Datenbanken muß der Pfad lokal zum ladenden Knoten sein, damit gleichzeitig aktive Ladeoperationen nicht versuchen, in die gleiche Datei zu schreiben.</li><li>2. Der Inhalt der Datei wird in einem asynchronen, gepufferten Modus auf Platte geschrieben. Bei einer fehlgeschlagenen oder unterbrochenen Ladeoperation ist die auf Platte festgeschriebene Anzahl von Datensätzen nicht zweifelsfrei bekannt, und nach einem LOAD REST-ART kann die Konsistenz nicht garantiert werden. Die Datei kann nur bei einer Ladeoperation, die nach dem Start in einem Durchlauf ausgeführt wird, als vollständig angesehen werden.</li><li>3. Dieser Wert unterstützt keine Dateinamen mit mehreren Dateierweiterungen. Beispiel: <pre>dumpfile = /home/svtdbm6/DUMP.FILE</pre> wird vom Dienstprogramm LOAD akzeptiert, <pre>dumpfile = /home/svtdbm6/DUMP.LOAD.FILE</pre> hingegen nicht.</li></ol>
implieddecimal	<p>Die Position eines implizierten Dezimalzeichens wird durch die Spaltendefinition bestimmt. Sie wird nicht mehr am Ende des Werts angenommen. Zum Beispiel wird der Wert 12345 in eine Spalte, für die DECIMAL(8,2) definiert ist, als 123.45 und <i>nicht</i> als 12345.00 geladen.</p>

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
timeformat="x"	<p>x ist das Zeitformat in der Quelldatei.<sup>a</sup> Gültige Zeitelemente sind:</p> <p>H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System)</p> <p>HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus)</p> <p>M - Minute (eine oder zwei Ziffern von 0 bis 59)</p> <p>MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Element M schließen sich gegenseitig aus)</p> <p>S - Sekunde (eine oder zwei Ziffern von 0 bis 59)</p> <p>SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus)</p> <p>SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus)</p> <p>TT - Meridiananzeiger (AM oder PM)</p> <p>Für jedes Element, das nicht angegeben wird, wird der Standardwert 0 zugeordnet. Einige Beispiele für Zeitformate:</p> <p>"HH:MM:SS"</p> <p>"HH.MM TT"</p> <p>"SSSSS"</p>

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
timestampformat="x"	x ist das Format der Zeitmarke in der Quelldatei. <sup>a</sup> Gültige Zeitmarkenelemente sind: YYYY - Jahr (vier Ziffern von 0000 bis 9999) M - Monat (eine oder zwei Ziffern von 1 bis 12) MM - Monat (zwei Ziffern von 1 bis 12; dieses Element und das Monatelement M schließen sich gegenseitig aus) D - Tag (eine oder zwei Ziffern von 1 bis 31) DD - Tag (zwei Ziffern von 1 bis 31; dieses Element und das Element D schließen sich gegenseitig aus) DDD - Tag des Jahres (drei Ziffern von 001 bis 366; dieses Element und andere Elemente für Tag oder Monat schließen sich gegenseitig aus) H - Stunde (eine oder zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System) HH - Stunde (zwei Ziffern von 0 bis 12 bei einem 12-Stunden-System und 0 bis 24 bei einem 24-Stunden-System; dieses Element und das Element H schließen sich gegenseitig aus) M - Minute (eine oder zwei Ziffern von 0 bis 59) MM - Minute (zwei Ziffern von 0 bis 59; dieses Element und das Minutenelement M schließen sich gegenseitig aus) S - Sekunde (eine oder zwei Ziffern von 0 bis 59) SS - Sekunde (zwei Ziffern von 0 bis 59; dieses Element und das Element S schließen sich gegenseitig aus) SSSSS - Sekunde des Tags ab 0.00 Uhr gerechnet (5 Ziffern von 00000 - 86399; dieses Element und alle anderen Zeitelemente schließen sich gegenseitig aus) UUUUUU - Mikrosekunde (6 Ziffern von 000000 bis 999999) TT - Meridiananzeiger (AM oder PM)



Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
Fortsetzung	<p>Für nicht angegebene Elemente YYYY, M, MM, D, DD oder DDD wird der Standardwert 1 zugeordnet. Für alle anderen nicht angegebenen Elemente wird der Standardwert 0 zugeordnet. Beispiel für ein Zeitmarkenformat:</p> <pre>"YYYY/MM/DD HH:MM:SS.UUUUUU"</pre> <p>Das folgende Beispiel veranschaulicht, wie Daten in eine Tabelle namens SCHEDULE importiert werden, die benutzerdefinierte Datums- und Zeitformate enthalten:</p> <pre>db2 import from delfile2 of del modified by timestampformat="yyyy.mm.dd hh:mm tt" insert into schedule</pre>
noeofchar	Das wahlfreie Dateiendezeichen x'1A' wird nicht als Ende der Datei erkannt. Die Verarbeitung wird wie bei einem normalen Zeichen fortgesetzt.
<b>ASC (ASCII-Format mit universellen Zeilenbegrenzern)</b>	
binarynumerics	<p>Numerische Daten (nicht jedoch DECIMAL-Daten) müssen im Binärformat und nicht in der Zeichendarstellung vorliegen. Dadurch werden aufwendige Umsetzungen vermieden.</p> <p>Diese Option wird nur mit positionsgebundenem ASC unterstützt, bei dem Sätze mit fester Länge verwendet werden, die mit der Option reclen festgelegt wird. Die Option noeofchar wird vorausgesetzt.</p> <p>Es gelten folgende Regeln:</p> <ul style="list-style-type: none"> <li>• Es wird keine Umsetzung zwischen Datentypen durchgeführt, mit Ausnahme von BIGINT, INTEGER und SMALLINT.</li> <li>• Die Datenlänge muß mit der jeweiligen Zielspaltendefinition übereinstimmen.</li> <li>• FLOAT-Werte müssen im IEEE-Gleitkommaformat vorliegen.</li> <li>• Binärdaten in der Ladequellendatei werden als Big-Endian gewertet, unabhängig von der Plattform, auf der die Ladeoperation ausgeführt wird.</li> </ul> <p><b>Anmerkung:</b> NULL-Werte dürfen in den Daten der Spalten, für die dieser Wert gilt, nicht vorkommen. Leerzeichen (normalerweise als NULL gewertet) werden als Binärwert gewertet, wenn dieser Wert verwendet wird.</p>

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
nochecklengths	<p>Wenn nochecklengths angegeben wird, wird versucht, jede Zeile zu laden, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltabellenspalte überschreitet. Solche Zeilen können erfolgreich geladen werden, wenn die Codepage-Umsetzung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, daß die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.</p>
nullindchar=x	<p>x ist ein einzelnes Zeichen. Ändert das Zeichen, das einen NULL-Wert angibt, in x. Der Standardwert für x ist <math>\backslash</math>.<sup>b</sup></p> <p>Bei diesem Wert wird bei EBCDIC-Datendateien zwischen Groß-/Kleinschreibung unterschieden, sofern es sich beim Zeichen nicht um ein lateinisches Zeichen handelt. Wenn beispielsweise für das NULL-Anzeigerzeichen der Buchstabe N angegeben wird, wird auch n als NULL-Anzeiger erkannt.</p>
packeddecimal	<p>Lädt gepackte Dezimaldaten direkt, da der Wert binarynumerics den Feldtyp DECIMAL nicht beinhaltet.</p> <p>Diese Option wird nur mit positionsgebundenem ASC unterstützt, bei dem Sätze mit fester Länge verwendet werden, die mit der Option reclen festgelegt wird. Die Option noeofchar wird vorausgesetzt.</p> <p>Unterstützte Werte für das Vorzeichenhalbbyte sind:</p> <ul style="list-style-type: none"><li>+ = 0xC 0xA 0xE 0xF</li><li>- = 0xD 0xB</li></ul> <p><b>Anmerkung:</b> NULL-Werte dürfen in den Daten der Spalten, für die dieser Wert gilt, nicht vorkommen. Leerzeichen (normalerweise als NULL gewertet) werden als Binärwert gewertet, wenn dieser Wert verwendet wird.</p> <p>Unabhängig von der Server-Plattform wird die Byteanordnung der Binärdaten in der Ladequellendatei als Big-Endian gewertet. Wenn dieser Wert unter OS/2 oder einem Windows-Betriebssystem verwendet wird, darf die Byteanordnung nicht zurückgenommen werden.</p>

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
reclen=x	x ist eine ganze Zahl mit dem Maximalwert 32 767. x Zeichen werden für jede Zeile gelesen, und es wird kein Zeilenvorschubzeichen verwendet, um das Ende der Zeile anzugeben.
striptblanks	<p>Schneidet abschließende Leerzeichen ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden Leerstellen beibehalten.</p> <p>Diese Option kann nicht zusammen mit striptnulls angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option t, die nur noch aus Gründen der Kompatibilität zu früheren Versionen unterstützt wird.</p>
striptnulls	<p>Schneidet abschließende NULL-Zeichen (0x00-Zeichen) ab, wenn Daten in ein Feld mit variabler Länge geladen werden. Wenn diese Option nicht angegeben wird, werden NULL-Zeichen beibehalten.</p> <p>Diese Option kann nicht zusammen mit striptblanks angegeben werden. Diese Optionen schließen sich gegenseitig aus.</p> <p><b>Anmerkung:</b> Diese Option ersetzt die veraltete Option padwithzero, die nur noch aus Gründen der Kompatibilität zu früheren Versionen unterstützt wird.</p>
zoneddecimal	<p>Lädt gezonte Dezimaldaten, da der Wert BINARYNUMERICS die Feldart DECIMAL nicht enthält. Diese Option wird nur mit positionsgebundenem ASC unterstützt, bei dem Sätze mit fester Länge verwendet werden, die mit der Option RECLen festgelegt wird. Die Option NOEOFCHAR wird vorausgesetzt.</p> <p>Die folgenden Halbbytevorzeichenwerte sind möglich:</p> <ul style="list-style-type: none"> <li>+ = 0xC 0xA 0xE 0xF</li> <li>- = 0xD 0xB</li> </ul> <p>Für Ziffern werden Werte von 0x0 bis 0x9 unterstützt.</p> <p>Für Zonen werden die Werte 0x3 und 0xF unterstützt.</p>
<b>DEL-Dateiformat (ASCII-Format ohne universelle Zeilenbegrenzer)</b>	

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
chardelx	<p><i>x</i> ist ein einzelner Zeichenfolgebegrenzer. Der Standardwert ist ein doppeltes Anführungszeichen ("). Das angegebene Zeichen wird anstelle von doppelten Anführungszeichen verwendet, um eine Zeichenfolge einzuschließen.<sup>bc</sup></p> <p>Auch das einfache Anführungszeichen (') kann wie folgt als Zeichenfolgebegrenzer festgelegt werden:</p> <pre>modified by chardel''</pre>
coldelx	<p><i>x</i> ist ein einzelner Zeichenspaltenbegrenzer. Der Standardwert ist ein Komma (.). Das angegebene Zeichen wird anstelle eines Kommas verwendet, um das Ende einer Spalte anzugeben.<sup>bc</sup></p>
datesiso	Datumsformat. Bewirkt, daß alle Datumswerte im ISO-Format geladen werden.
decplusblank	Zeichen für Pluszeichen. Bewirkt, daß positiven Dezimalwerten statt eines Pluszeichens (+) ein Leerzeichen vorangestellt wird. Die Standardaktion besteht darin, Dezimalwerten ein Pluszeichen voranzustellen.
decptx	<i>x</i> ist ein einzelnes Ersatzzeichen für den Punkt als Dezimalzeichen. Der Standardwert ist ein Punkt (.). Das angegebene Zeichen wird anstelle eines Punktes als Dezimalzeichen verwendet. <sup>bc</sup>
delprioritychar	<p>Die aktuelle Standardpriorität für Begrenzer ist: Satzbegrenzer, Zeichenbegrenzer, Spaltenbegrenzer. Durch diesen Wert werden vorhandene Anwendungen geschützt, die auf die früher geltende Priorität angewiesen sind, indem die Begrenzerpriorität wie folgt geändert wird: Zeichenbegrenzer, Satzbegrenzer, Spaltenbegrenzer. Syntax:</p> <pre>db2 load ... modified by delprioritychar ...</pre> <p>Beispiel anhand der folgenden DEL-Datendatei:</p> <pre>"Smith, Joshua",4000,34.98&lt;zeilenbegrenzer&gt; "Vincent,&lt;zeilenbegrenzer&gt;, is a manager", ... ... 4005,44.37&lt;zeilenbegrenzer&gt;</pre> <p>Wenn der Wert <code>delprioritychar</code> angegeben wird, gibt es in dieser Datendatei nur zwei Zeilen. Der zweite <code>&lt;zeilenbegrenzer&gt;</code> wird als Teil der ersten Datenspalte der zweiten Zeile gewertet, während der erste und dritte <code>&lt;zeilenbegrenzer&gt;</code> als eigentliche Satzbegrenzer gewertet werden. Wenn dieser Wert <i>nicht</i> angegeben wird, gibt es in dieser Datendatei drei Zeilen, jeweils getrennt durch einen <code>&lt;zeilenbegrenzer&gt;</code>.</p>

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
dlldelx	<p>x ist ein DATALINK-Begrenzer (Einzelzeichen). Der Standardwert ist ein Semikolon (;). Das angegebene Zeichen wird anstelle eines Semikolons als Trennzeichen zwischen Feldern für einen DATALINK-Wert verwendet. Es ist erforderlich, weil ein DATALINK-Wert mehrere Teilwerte enthalten kann. <sup>bcd</sup></p> <p><b>Anmerkung:</b> x darf nicht mit dem Zeichen übereinstimmen, das als Zeilen-, Spalten- oder Zeichenfolgebegrenzer angegeben wurde.</p>
keepblanks	<p>Erhält die führenden und abschließenden Leerzeichen in allen Feldern des Typs CHAR, VARCHAR, LONG VARCHAR oder CLOB. Ohne diese Option werden alle führenden und abschließenden Leerzeichen, die nicht innerhalb von Zeichenbegrenzern stehen, entfernt, und in die Tabelle wird für jedes leere Feld ein Nullwert eingefügt.</p> <p>Das folgende Beispiel veranschaulicht, wie Daten in eine Tabelle namens TABLE1 geladen werden, wobei alle führenden und abschließenden Leerzeichen in der Daten-datei erhalten bleiben:</p> <pre>db2 load from delfile3 of del modified by keepblanks insert into table1</pre>
nodoubledel	<p>Unterdrückt die Erkennung von doppelten Zeichenbegrenzern. Weitere Informationen finden Sie unter „Einschränkungen für Begrenzer“ auf Seite 20.</p>
<b>IXF-Dateiformat</b>	
forcein	<p>Weist das Dienstprogramm an, Daten trotz Codepage-Abweichungen zu akzeptieren und die Umsetzung zwischen Codepages zu unterdrücken.</p> <p>Zielfelder mit fester Länge werden überprüft, um sicherzustellen, daß sie groß genug für die Daten sind. Wenn der Wert nochecklengths angegeben ist, wird keine Überprüfung durchgeführt und nach Möglichkeit jede Zeile geladen.</p>

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
nochecklengths	Wenn nochecklengths angegeben wird, wird versucht, jede Zeile zu laden, auch wenn die Quelldaten eine Spaltendefinition enthalten, die die Größe der Zieltablenspalte überschreitet. Solche Zeilen können erfolgreich geladen werden, wenn die Codepage-Umsetzung eine Verkleinerung der Quelldaten bewirkt. Beispielsweise könnten 4-Byte-EUC-Daten in der Quelle auf 2-Byte-DBCS-Daten im Ziel verkleinert werden und den halben Speicherplatz belegen. Diese Option ist besonders nützlich, wenn bekannt ist, daß die Quelldaten in jedem Fall Platz finden, unabhängig von abweichenden Spaltendefinitionen.

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
	<p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"> <li>Das Dienstprogramm LOAD setzt keine Warnung ab, wenn versucht wird, nicht unterstützte Dateitypen mit der Option MODIFIED BY zu verwenden. Wenn dies versucht wird, schlägt die Ladeoperation fehl, und es wird ein Fehlercode zurückgegeben.</li> <li><sup>a</sup> Die Zeichenfolge für das Datumsformat muß in doppelte Anführungszeichen gesetzt werden. Feldtrennzeichen dürfen die folgenden Zeichen nicht enthalten: a-z, A-Z und 0-9. Das Feldtrennzeichen darf nicht mit dem Zeichenbegrenzer oder dem Feldbegrenzer im DEL-Dateiformat identisch sein. Die Verwendung eines Feldtrennzeichens ist wahlfrei, sofern die Anfangs- und Endposition eines Elements eindeutig ist. Dies ist beispielsweise dann nicht der Fall, wenn (abhängig vom Wert) Elemente wie D, H, M oder S verwendet werden, da die Einträge in ihrer Länge variabel sind.</li> </ol> <p>Bei Zeitmarkenformaten muß darauf geachtet werden, eine Mehrdeutigkeit zwischen den Deskriptoren für Monat und Minute zu vermeiden, da beide den Buchstaben M verwenden. Ein Monatsfeld muß neben anderen Datumsfeldern stehen. Ein Minutenfeld muß neben anderen Zeitfeldern stehen. Die folgenden Zeitmarkenformate sind beispielsweise mehrdeutig:</p> <p>"M" (Hierdurch könnte sowohl eine Monats- als auch eine Minutenangabe dargestellt sein.)  "M:M" (In diesem Fall ist nicht eindeutig, welches Element für den Monat und welches für die Minutenangabe steht.)  "M:YYYY:M" (Hier werden beide Elemente M als Monat interpretiert.)  "S:M:YYYY" (In diesem Fall steht das Element M sowohl neben einem Zeit- als auch neben einem Datumswert.)</p> <p>Im Fall einer Mehrdeutigkeit meldet das Dienstprogramm eine Fehlermeldung, und die Operation schlägt fehl.</p> <p>Die folgenden Zeitmarkenformate sind beispielsweise eindeutig:</p> <p>"M:YYYY" (Monat)  "S:M" (Minute)  "M:YYYY:S:M" (Monat...Minute)  "M:H:YYYY:M:D" (Minute...Monat)</p> <p><b>Anmerkung:</b> Einigen Zeichen (z. B. doppelten Anführungszeichen und umgekehrten Schrägstrichen) muß ein Escape-Zeichen vorangestellt sein (beispielsweise \).</p>

## Dateitypwerte (LOAD)

Tabelle 8. Gültige Dateitypwerte (LOAD) (Forts.)

Wert	Beschreibung
	<p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"><li><sup>b</sup> Das Zeichen muß in der Codepage der Quelldaten angegeben werden. Der Zeichencodepunkt (statt des Zeichensymbols) kann mit der Syntax xJJ bzw. 0xJJ angegeben werden, wobei JJ die hexadezimale Darstellung des Codepunkts ist. Beispiel: Das Zeichen # kann wie folgt als Spaltenbegrenzungszeichen angegeben werden: ... modified by coldel# ... ... modified by coldel0x23 ... ... modified by coldelX23 ...</li><li><sup>c</sup> Unter „Einschränkungen für Begrenzer“ auf Seite 20 werden Einschränkungen aufgelistet, die für zur Überschreibung von Begrenzern verwendbare Zeichen gelten.</li><li><sup>d</sup> Auch wenn das DATALINK-Begrenzungszeichen in der URL-Syntax ein gültiges Zeichen ist, verliert es im Rahmen der Ladeoperation seine besondere Bedeutung.</li></ol>



---

## Ausnahmetabelle

Die Ausnahmetabelle ist eine durch den Benutzer erstellte Tabelle, die die Definition der zu ladenden Tabelle nachbildet und einige zusätzliche Spalten beinhaltet. Angegeben wird sie mit der Klausel FOR EXCEPTION des Befehls LOAD. Eine Ausnahmetabelle darf keine Identitätsspalte oder generierte Spalten anderen Typs enthalten. Ist in der Primärtabelle eine Identitätsspalte vorhanden, sollte die entsprechende Spalte in der Ausnahmetabelle nur den Spaltentyp, die Spaltenlänge und die Attribute enthalten, die angeben, ob die Spalte Nullwerte enthalten kann oder nicht. Die Ausnahmetabelle wird verwendet, um Kopien der Zeilen zu speichern, die die Regeln eines eindeutigen Index verletzen. Vom Dienstprogramm werden keine Überprüfungen auf ungültige Integritätsbedingungen oder Fremdschlüssel vorgenommen. Hiervon ausgenommen sind Verletzungen der Eindeutigkeit. DATALINK-Ausnahmen werden ebenfalls in der Ausnahmetabelle erfaßt.

Ein eindeutiger Schlüssel ist ein Schlüssel, bei dem keine zwei Werte gleich sind. Der zur Umsetzung dieser Integritätsbedingung verwendete Mechanismus wird als eindeutiger Index bezeichnet. Ein Primärschlüssel ist ein Sonderfall eines eindeutigen Schlüssels. Eine Tabelle kann nicht mehr als einen Primärschlüssel aufweisen.

**Anmerkung:** Zeilen, die vor der Indexerstellung aufgrund ungültiger Daten zurückgewiesen wurden, werden nicht in die Ausnahmetabelle aufgenommen.

Zeilen werden an vorhandene Informationen in der Ausnahmetabelle angefügt. Dies können auch ungültige Zeilen aus früheren Ladeoperationen sein. Wenn die Ausnahmetabelle nur die ungültigen Zeilen der aktuellen Ladeoperation enthalten soll, müssen Sie die bereits vorhandenen Zeilen löschen, bevor Sie das Dienstprogramm aufrufen.

Die Ausnahmetabelle, die mit dem Dienstprogramm LOAD verwendet wird, ist identisch mit den Ausnahmetabellen, die von der Anweisung SET INTEGRITY verwendet werden.

Eine Ausnahmetabelle sollte verwendet werden, wenn Daten geladen werden, die einen eindeutigen Index, aber möglicherweise auch doppelte Datensätze aufweisen. Wenn keine Ausnahmetabelle angegeben wird und doppelte Datensätze auftreten, wird die Ladeoperation fortgesetzt, und es erscheint nur eine Warnung, die auf die gelöschten doppelten Datensätze hinweist. Die Datensätze selbst werden nicht protokolliert.

Nach Beendigung der Ladeoperation können Sie die Informationen in der Ausnahmetabelle verwenden, um fehlerhafte Daten zu korrigieren. Die korrigierten Daten können dann in die Tabelle eingefügt werden.

## LOAD-Ausnahmetabelle

Weitere Informationen zu Ausnahmetabellen finden Sie im Handbuch *SQL Reference*.

---

### Speicherauszugsdatei

Der Wert *dumpfile* teilt dem Dienstprogramm LOAD den Namen und die Speicherposition der Ausnahmedatei mit, in die die zurückgewiesenen Zeilen geschrieben werden. Bei Ausführung in einer Umgebung mit partitionierten Datenbanken erhält der Name eine Erweiterung, die die Partitionsnummer kennzeichnet, in der die Ausnahmen generiert wurden. Beispiel:

```
dumpfile = "/u/username/dumpit"
```

Auf Partition null wird damit eine Datei namens `/u/username/dumpit.000` generiert. Auf Partition fünf wird damit eine Datei namens `/u/username/dumpit.005` generiert usw.

Ist der Datensatz kürzer als 32 768 Byte, wird er vollständig in die Speicherauszugsdatei kopiert. Ist er länger, wird ein Datensatzfragment (das auch die Endbyte des Datensatzes enthält) in die Datei geschrieben.

Weitere Informationen zu LOAD-Dateitypwerten finden Sie unter „Dateitypwerte (LOAD)“ auf Seite 140.

---

### Temporäre Dateien (LOAD)

DB2 erstellt bei der LOAD-Verarbeitung temporäre Binärdateien. Diese Dateien werden zur Wiederherstellung einer Ladeoperation nach einem Systemabsturz, bei LOAD TERMINATE-Operationen, für Warnungen und Fehlernachrichten sowie die Laufzeitsteuerdaten verwendet. Die temporären Dateien werden entfernt, nachdem die Ladeoperation ohne Fehler beendet wurde.

Die temporären Dateien werden in einen Pfad geschrieben, der mit dem Parameter *temp-pfadname* des Befehls LOAD oder mit dem Parameter *pRemoteMsgFileName* der API **sqlload** angegeben werden kann. Der Standardpfad ist ein Unterverzeichnis des Datenbankverzeichnisses.

Die temporären Dateien befinden sich auf der Server-Maschine, und es wird darauf ausschließlich vom DB2-Exemplar zugegriffen. Daher muß jede Angabe eines Pfadnamens im Parameter *temp-pfadname* die Verzeichnisstruktur auf dem Server und nicht die auf dem Client berücksichtigen. Der Eigner des DB2-Exemplars muß darüber hinaus Schreib- und Lesezugriff für den Pfad haben.

**Anmerkung:** In einem MPP-System muß sich der Pfad für die temporären Dateien auf einem lokalen Datenträger und nicht auf einem NFS-Mount befinden. Befindet sich der Pfad auf einem NFS-Mount, kommt es bei der Ladeoperation zu signifikanten Leistungseinbußen.

**Achtung:** Die in diesen Pfad geschriebenen temporären Dateien dürfen unter keinen Umständen manipuliert werden. Andernfalls treten bei der Ladeoperation Störungen auf, und die Integrität der Datenbank wird gefährdet.

---

### Protokollsätze zum Dienstprogramm LOAD

Der Dienstprogrammanager erzeugt Protokollsätze für eine Reihe von DB2-Dienstprogrammen. Dazu zählt auch das Dienstprogramm LOAD. Die Protokollsätze kennzeichnen den Anfang oder das Ende eines bestimmten Vorgangs. Die folgenden Protokollsätze sind Ladeoperationen zugeordnet:

- **LOAD START.** Dieser Protokollsatz ist dem Anfang einer Ladeoperation zugeordnet.
- **TABLE LOAD DELETE START.** Dieser Protokollsatz ist dem Anfang der DELETE-Phase einer Ladeoperation zugeordnet. Die DELETE-Phase wird nur gestartet, wenn doppelte Primärschlüsselwerte vorliegen. Während der DELETE-Phase wird jede Löschoperation für einen Tabellensatz oder einen Indexschlüssel protokolliert.
- **LOAD DELETE START COMPENSATION.** Dieser Protokollsatz ist dem Ende der DELETE-Phase einer Ladeoperation zugeordnet.
- **LOAD PENDING LIST.** Dieser Protokollsatz wird geschrieben, wenn eine Ladetransaktion festgeschrieben wird. Die Zurückstellungsliste ist eine verbundene Liste von nicht wiederherstellbaren Operationen, die verzögert werden, bis die Transaktion festgeschrieben wird. Auf diese Transaktion folgt kein COMMIT-Protokollsatz.
- Während einer nicht wiederherstellbaren Ladeoperation wird immer dann ein Protokollsatz geschrieben, wenn eine Erweiterung des Tabellenbereichs zugeordnet oder gelöscht wird.

Eine Beschreibung der Struktur dieser Protokollsätze finden Sie im Handbuch *Administrative API Reference*.

### Zeichensatz und NLS

Die DB2 UDB-Dienstprogramme zum Versetzen von Daten bieten die folgende Unterstützung in der Landessprache (National Language Support - NLS):

- Die Dienstprogramme IMPORT und EXPORT stellen eine automatische Umsetzung aus einer Client-Codepage in die Server-Codepage bereit.
- Für die Dienstprogramme LOAD und AutoLoader können Daten aus einer beliebigen Codepage in die Server-Codepage umgesetzt werden, indem bei DEL- und ASC-Dateien der Wert codepage verwendet wird.
- Bei allen Dienstprogrammen werden IXF-Daten automatisch aus ihrer ursprünglichen Codepage (wie in der IXF-Datei gespeichert) in die Server-Codepage umgesetzt.

Manchmal kann es aufgrund ungleicher Codepages zu Situationen kommen, in denen die Zeichendaten möglicherweise verlängert oder gekürzt werden. Solche Situationen können bei EUC- und DBCS-Zeichensätzen für Japanisch oder traditionelles Chinesisch auftreten, die verschiedene Längencodierungen für dasselbe Zeichen verwenden können. Normalerweise wird die Länge der Eingabedaten vor dem Einlesen von Daten mit der Länge der Zielspalte verglichen. Wenn die Länge der Eingabedaten die Länge der Zielspalte überschreitet, werden Nullwerte in die Spalte eingefügt, wenn für die Spalte Nullwerte zulässig sind. Andernfalls wird die Anforderung zurückgewiesen. Wenn der Wert `nochecklengths` (siehe „Dateitypwerte (LOAD)“ auf Seite 140) angegeben ist, wird vorab kein Vergleich durchgeführt, und die Daten werden nach Möglichkeit geladen. Wenn die Daten nach der Umsetzung zu lang sind, wird die entsprechende Zeile zurückgewiesen. Ansonsten werden die Daten geladen.

---

### Beispiele für LOAD-Sitzungen

#### CLP-Beispiele

##### Beispiel 1

TABLE1 enthält 5 Spalten:

- COL1 VARCHAR 20 NOT NULL WITH DEFAULT
- COL2 SMALLINT
- COL3 CHAR 4
- COL4 CHAR 2 NOT NULL WITH DEFAULT
- COL5 CHAR 2 NOT NULL

ASCFILE1 enthält 6 Elemente:

- ELE1 Positionen 01 bis 20
- ELE2 Positionen 21 bis 22
- ELE5 Positionen 23 bis 23

- ELE3 Positionen 24 bis 27
- ELE4 Positionen 28 bis 31
- ELE6 Positionen 32 bis 32
- ELE6 Positionen 33 bis 40

Datensätze:

```
1...5...10...15...20...25...30...35...40
Testdaten 1      XXN 123abcdN
Testdaten 2 und 3  QQY   wxyzN
Testdaten 4,5 und 6 WWN6789   Y
```

Mit dem folgenden Befehl wird die Tabelle aus der Datei geladen:

```
db2 load from ascfile1 of asc modified by striptblanks reclen=40
method L (1 20, 21 22, 24 27, 28 31)
null indicators (0,0,23,32)
insert into table1 (col1, col5, col2, col3)
```

### Anmerkungen:

1. Die Angabe `striptblanks` im Parameter `MODIFIED BY` bewirkt, daß Leerzeichen in `VARCHAR`-Spalten (zum Beispiel `COL1`, die 11, 17 und 19 Byte lang ist, in den Zeilen 1, 2 bzw. 3) abgeschnitten werden.
2. Die Angabe `reclen=40` im Parameter `MODIFIED BY` legt fest, daß es am Ende der Eingabedatensätze kein Zeilenvorschubzeichen gibt und jeder Datensatz 40 Byte lang ist. Die letzten 8 Byte werden nicht zum Laden der Tabelle verwendet.
3. Da `COL4` nicht in der Eingabedatei bereitgestellt wird, wird sie in `TABLE1` mit dem Standardwert eingefügt (sie ist mit `NOT NULL WITH DEFAULT` definiert).
4. Die Positionen 23 und 32 werden verwendet, um anzugeben, ob in `COL2` und `COL3` von `TABLE1` für eine gegebene Zeile `NULL` geladen wird. Wenn die Nullanzeigerposition der Spalte für einen gegebenen Satz `Y` enthält, ist die Spalte `NULL`. Enthält sie `N`, werden die Datenwerte in den Datenpositionen der Spalte des Eingabedatensatzes (wie in `L(.....)` definiert) als Quelle der Spaltendaten für die Zeile verwendet. In diesem Beispiel ist keine Spalte in Zeile 1 `NULL`. `COL2` in Zeile 2 ist `NULL`, und `COL3` in Zeile 3 ist `NULL`.
5. In diesem Beispiel werden die `NULL`-Anzeiger für `COL1` und `COL5` als 0 (Null) angegeben, wodurch festgelegt wird, daß die Daten keine Nullwerte enthalten dürfen.
6. Der `NULL`-Anzeiger für eine gegebene Spalte kann sich an beliebiger Position im Eingabedatensatz befinden, diese muß jedoch angegeben werden, und der Wert `Y` oder `N` muß ebenfalls angegeben werden.

## Beispiele für LOAD-Sitzungen

### Beispiel 2 (Laden von großen Objekten aus Dateien)

TABLE1 enthält 3 Spalten:

- COL1 CHAR 4 NOT NULL WITH DEFAULT
- LOB1 LOB
- LOB2 LOB

ASCFILE1 enthält 3 Elemente:

- ELE1 Positionen 01 bis 04
- ELE2 Positionen 06 bis 13
- ELE3 Positionen 15 bis 22

Die folgenden Dateien befinden sich in /u/user1 oder /u/user1/bin:

- ASCFILE2 enthält LOB-Daten
- ASCFILE3 enthält LOB-Daten
- ASCFILE4 enthält LOB-Daten
- ASCFILE5 enthält LOB-Daten
- ASCFILE6 enthält LOB-Daten
- ASCFILE7 enthält LOB-Daten

Datensätze in ASCFILE1:

```
1...5....10...15...20...25...30.  
REC1 ASCFILE2 ASCFILE3  
REC2 ASCFILE4 ASCFILE5  
REC3 ASCFILE6 ASCFILE7
```

Mit dem folgenden Befehl wird die Tabelle aus der Datei geladen:

```
db2 load from ascfile1 of asc  
 lobs from /u/user1, /u/user1/bin  
  modified by lobsinfile reclen=22  
  method L (1 4, 6 13, 15 22)  
  insert into table1
```

### Anmerkungen:

1. Die Angabe `lobsinfile` im Parameter `MODIFIED BY` teilt dem Ladeprogramm mit, daß alle LOB-Daten aus Dateien geladen werden.
2. Die Angabe `reclen=22` im Parameter `MODIFIED BY` legt fest, daß es am Ende der Eingabedatensätze kein Zeilenvorschubzeichen gibt und jeder Datensatz 22 Byte lang ist.
3. LOB-Daten sind in 6 Dateien enthalten, ASCFILE2 bis ASCFILE7. Jede Datei enthält die Daten, die verwendet werden, um eine LOB-Spalte für eine bestimmte Zeile zu laden. Die Beziehung zwischen großen Objekten und anderen Daten wird in ASCFILE1 angegeben. Der erste Satz dieser Datei weist das Ladeprogramm an, REC1 in COL1 von Zeile 1 zu stellen.

Der Inhalt von ASCFILE2 wird verwendet, um LOB1 von Zeile 1 zu laden, und der Inhalt von ASCFILE3 wird verwendet, um LOB2 von Zeile 1 zu laden. ASCFILE4 und ASCFILE5 werden wiederum verwendet, um LOB1 und LOB2 von Zeile 2 zu laden, ASCFILE6 und ASCFILE7 werden verwendet, um die großen Objekte von Zeile 3 zu laden.

4. Der Parameter LOBS FROM enthält zwei Pfade, die nach den angegebenen LOB-Dateien durchsucht werden, wenn diese Dateien vom Ladeprogramm angefordert werden.
5. Zum Laden von großen Objekten aus ASCFILE1 (einer ASCII-Datei mit universellen Zeilenbegrenzern) ohne den Wert `lobsinfile` müssen die folgenden Regeln eingehalten werden:
  - Die Gesamtlänge eines jeden Datensatzes, einschließlich großer Objekte, darf 32 KB nicht überschreiten.
  - LOB-Felder in Eingabedatensätzen müssen eine feste Länge haben und LOB-Daten gegebenenfalls mit Leerzeichen aufgefüllt werden.
  - Der Wert `striptblanks` muß angegeben werden, damit die abschließenden Leerzeichen, die zum Auffüllen von großen Objekten verwendet werden, entfernt werden können, wenn die großen Objekte in die Datenbank eingefügt werden.

### Beispiel 3 (unter Verwendung von Speicherauszugsdateien)

Tabelle FRIENDS ist wie folgt definiert:

```
table friends "( c1 INT NOT NULL, c2 INT, c3 CHAR(8) )"
```

Wenn versucht wird, die folgenden Datensätze in diese Tabelle zu laden:

```
23, 24, bobby  
, 45, john  
4,, mary
```

wird die zweite Zeile zurückgewiesen, weil das erste INT NULL ist und die Spaltendefinition NOT NULL lautet. Spalten, die Anfangszeichen enthalten, die nicht mit dem DEL-Format konsistent sind, führen zu einem Fehler, und der Datensatz wird zurückgewiesen. Solche Datensätze können in eine Speicherauszugsdatei geschrieben werden (siehe Tabelle 8 auf Seite 140).

DEL-Daten, die in einer Spalte außerhalb von Zeichenbegrenzern vorkommen, werden ignoriert, führen jedoch zur Generierung einer Warnung. Beispiel:

```
22,34,"bob"  
24,55,"sam" sdf
```

Das Dienstprogramm lädt "sam" in die dritte Spalte der Tabelle, und die Zeichen "sdf" werden in einer Warnung markiert. Der Datensatz wird nicht zurückgewiesen. Anderes Beispiel:

```
22 3, 34,"bob"
```

## Beispiele für LOAD-Sitzungen

Das Dienstprogramm lädt 22,34, "bob" und generiert eine Warnung, daß einige Daten in Spalte eins nach 22 ignoriert wurden. Der Datensatz wird nicht zurückgewiesen.

### Beispiel 4 (Laden von DATALINK-Daten)

Mit dem folgenden Befehl wird die Tabelle MOVIE TABLE aus der Eingabedatei delfile1 geladen, die Daten im DEL-Format enthält:

```
db2 load from delfile1 of del
modified by dldel|
insert into movietable (actorname, description, url_making_of, url_movie)
datalink specification (dl_url_default_prefix "http://narang"),
(dl_url_replace_prefix "http://bomdel" dl_url_suffix ".mpeg")
for exception excptab
```

### Anmerkungen:

1. Die Tabelle enthält vier Spalten:

actorname	VARCHAR(n)
description	VARCHAR(m)
url_making_of	DATALINK (mit LINKTYPE URL)
url_movie	DATALINK (mit LINKTYPE URL)

- Die DATALINK-Daten in der Eingabedatei verwenden den vertikalen Balken (|) als Teilfeldbegrenzer.
- Wenn ein Spaltenwert für url\_making\_of keine Präfixzeichenfolge aufweist, wird "http://narang" verwendet.
- Jeder Spaltenwert ungleich NULL für url\_movie erhält "http://bomdel" als Präfix. Vorhandene Werte werden ersetzt.
- An jeden Spaltenwert ungleich NULL für url\_movie wird ".mpeg" an den Pfad angefügt. Beispiel: Wenn ein Spaltenwert für url\_movie "http://server1/x/y/z" ist, wird er als "http://bomdel/x/y/z.mpeg" gespeichert; ist der Wert "/x/y/z", wird er als "http://bomdel/x/y/z.mpeg" gespeichert.
- Wenn Verletzungen des eindeutigen Indexes oder DATALINK-Verletzungen auftreten, während die Tabelle geladen wird, werden die betroffenen Datensätze aus der Tabelle gelöscht und in die Ausnahmetabelle excptab gestellt.



### Beispiel 5 (Laden einer Tabelle mit einer Identitätsspalte)

TABLE1 enthält 4 Spalten:

- C1 VARCHAR(30)
- C2 INT GENERATED BY DEFAULT AS IDENTITY
- C3 DECIMAL(7,2)
- C4 CHAR(1)

TABLE2 entspricht TABLE1, wobei jedoch C2 eine als GENERATED ALWAYS definierte Identitätsspalte ist.

Die Datensätze in DATAFILE1 (DEL-Format) lauten:

```
"Liszt"
"Holst",,187.43, H
"Grieg",100, 66.34, G
"Satie",101, 818.23, I
```

Die Datensätze in DATAFILE2 (DEL-Format) lauten:

```
"Liszt", 74.49, A
"Holst", 0.01, H
"Grieg", 66.34, G
"Satie", 818.23, I
```

#### Anmerkungen:

1. Der folgende Befehl generiert Identitätswerte für die Zeilen 1 und 2, da in der Datei DATAFILE1 keine Identitätswerte für diese Zeilen zur Verfügung gestellt werden. Den Zeilen 3 und 4 wird jedoch der benutzerdefinierte Identitätswert 100 bzw. 101 zugeordnet.

```
db2 load from datafile1.del of del replace into table1
```

2. Um die Datei DATAFILE1 so in die Tabelle TABLE1 zu laden, daß für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 load from datafile1.del of del method P(1, 3, 4)
replace into table1 (c1, c3, c4)
db2load from datafile1.del of del modified by identityignore
replace into table1
```

3. Um die Datei DATAFILE2 so in die Tabelle TABLE1 zu laden, daß für alle Zeilen Identitätswerte generiert werden, setzen Sie einen der folgenden Befehle ab:

```
db2 load from datafile2.del of del replace into table1 (c1, c3, c4)
db2 load from datafile2.del of del modified by identitymissing
replace into table1
```

## Beispiele für LOAD-Sitzungen

- Um die Datei DATAFILE1 so in die Tabelle TABLE2 zu laden, daß den Zeilen 3 und 4 die Identitätswerte 100 bzw. 101 zugeordnet werden, setzen Sie den folgenden Befehl ab:

```
db2 load from datafile1.del of del modified by identityoverride  
replace into table2
```

In diesem Fall werden die Zeilen 1 und 2 zurückgewiesen, da das Dienstprogramm angewiesen wurde, die durch das System generierten Identitätswerte mit den benutzerdefinierten Werten zu überschreiben. Sind jedoch keine benutzerdefinierten Werte vorhanden, muß die Zeile zurückgewiesen werden, da Identitätsspalten implizit nicht NULL sind.

- Wird die Datei DATAFILE1 in die Tabelle TABLE2 geladen, ohne daß einer der identitätsbezogenen Datentypen verwendet wird, werden die Zeilen 1 und 2 geladen, die Zeilen 3 und 4 jedoch zurückgewiesen, da diese Zeilen eigene Werte, die nicht NULL sind, liefern und die Identitätsspalte als GENERATED ALWAYS definiert wurde.

## API-Beispiele

Das folgende Beispielprogramm zeigt, wie folgende Aufgaben ausgeführt werden:

- Exportieren von Informationen aus der Tabelle EMP\_RESUME der Beispieldatenbank SAMPLE in die Datei EXPTABLE.DEL
- Laden dieser Informationen aus der Textdatei mit begrenzter Satzlänge in eine neue Tabelle, LOADTABLE

Zusatzinformationen zur Beispieldatenbank SAMPLE finden Sie im Handbuch *Systemverwaltung*.

Die Quellendatei für dieses Beispielprogramm (tload.sqc) befindet sich beim Betriebssystem Windows bzw. OS/2 im Verzeichnis sqllib\samples\c. Bei Systemen auf UNIX-Basis befindet es sich im Verzeichnis sqllib/samples/c. Es enthält sowohl DB2-API-Aufrufe als auch Aufrufe mit eingebettetem SQL. Die Prozedurdatei bldapp, die sich im gleichen Verzeichnis befindet, enthält die Befehle zur Erstellung dieses und anderer Beispielprogramme. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, und Zusatzinformationen zu Kompilierungs- und Programmverbindungsoptionen finden Sie im Handbuch *Application Building Guide*. Gehen Sie wie folgt vor, um das Beispielprogramm tload aus der Quellendatei tload.sqc zu erstellen:

1. Kopieren Sie die Dateien tload.sqc, bldapp\*, utilemb.c und utilemb.h in ein Arbeitsverzeichnis.
2. Setzen Sie, falls der Datenbankmanager nicht aktiv ist, den Befehl db2start ab.
3. Geben Sie bldapp tload sample ein. Die folgenden Dateien werden generiert:

```
tload.bnd
tload.c
util.obj/util.o
tload.obj/tload.o
tload.exe/tload
```

Geben Sie tload ein, um das Beispielprogramm (ausführbare Datei) zu starten. Es kann unter Umständen nützlich sein, einige der generierten Dateien näher zu betrachten, etwa die Nachrichtendateien und die ASCII-Datendatei mit begrenzter Satzlänge.

```

/*****
**
** Source File Name = tload.sqc 1.4
**
** Licensed Materials - Property of IBM
**
** (C) COPYRIGHT International Business Machines Corp. 1995, 1997
** All Rights Reserved.
**
** US Government Users Restricted Rights - Use, duplication or
** disclosure restricted by GSA ADP Schedule Contract with IBM Corp.
**
**
** PURPOSE :
**   To show the use of the QUIESCE TABLESPACE and the LOAD APIs.
**   - EXPORT the EMP_RESUME table into a comma delimited file.
**   - create a temporary table ('loadtable').
**   - QUIESCE the TABLESPACES, preparing the temporary table to be
**     LOADable.
**   - LOAD the comma delimited file into a temporary table ('loadtable').
**
** STRUCTURES USED :
**   sqldcol
**   sqlchar
**   sqluexpt_out
**   sqlca
**
** APIs USED :
**   EXPORT                                sqluexpr
**   QUIESCE TABLESPACE FOR TABLES      sqluvqdp
**   LOAD                                  sqluload
**
** FUNCTIONS DECLARED :
**   'C' COMPILER LIBRARY :
**     stdio.h - printf
**     string.h - fgets, strncpy
**
** DBMS LIBRARY :
**   sqlenv.h - see "APIs USED" above
**
** OTHER :
**   external :
**     check_error :    Checks for SQLCODE error, and prints out any

```

## Beispiele für LOAD-Sitzungen

```
**          [in UTIL.C]          related information available.
**
**      EXTERNAL DEPENDANCIES :
**      - Ensure existence of database (SAMPLE) for precompile purposes.
**      - Precompile with the SQL precompiler (PREP in DB2)
**      - Bind to a database (BIND in DB2)
**      - Compile and link with the IBM Cset++ compiler (AIX and OS/2)
**          or the Microsoft Visual C++ compiler (Windows)
**          or the compiler supported on your platform.
**
**      *****/
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sqlenv.h>
#include <sqlutil.h>
#include <malloc.h>
#include "utilemb.h"

#ifdef DB2AIX
#define WORKDIR "/u/workdir"
#else
#define WORKDIR "."
#endif

EXEC SQL INCLUDE SQLCA;
int main (int argc, char *argv[]) {
    short int          callerAction = 0;
    struct sql_dcol    DataDescriptor;
    struct sqlchar     *ActionString;
    struct sqlchar     *FileTypeMod;
    struct sqluexpt_out outputInfo;
    char               datafile[] = "EXPTABLE.DEL";
    char               statement[] = "SELECT empno, photo_format, picture FROM emp_photo";
    char               impStatement[] = "INSERT INTO loadtable (num, format, photo)";
/*
    char               statement[] = "SELECT empno, photo_format FROM emp_photo";
    char               impStatement[] = "INSERT INTO loadtable (num, format)";
*/
    char               msgfile_x[] = "EXPMSG.TXT";
    char               FileType[] = SQL_DEL;

    char               table_name[18];

    /* Variables for the LOAD API */
    struct sqlu_media_list DataFileList;
    struct sqlu_media_list *pLobPathList;
    struct sqlu_load_in    InputInfo;
    struct sqlu_load_out   OutputInfo;
    struct sqlu_media_list *pWorkDirectoryList = NULL;
    struct sqlu_media_list *pCopyTargetList;
    char                   LocalMsgFileName[] = "LOADMSG";
    char                   RemoteMsgFileName[] = "RLOADMSG";
```

```

short          CallerAction;
sqlint32      *pNullIndicators;
void          *pReserved;

EXEC SQL BEGIN DECLARE SECTION;
    char userid[9];
    char passwd[19];
EXEC SQL END DECLARE SECTION;

printf ("This is sample program 'tload.sqc'\n");

/* Initialize structures. */
memset(&DataFileList, 0, sizeof(struct sqlu_media_list));
memset(&InputInfo, 0, sizeof(struct sqluload_in));
memset(&OutputInfo, 0, sizeof(struct sqluload_out));

/* need to preset the size of structure field and counts */
outputInfo.sizeOfStruct = SQLUEXPT_OUT_SIZE;

/*****\
* need to allocate the proper amount of space for the SQL statement *
\*****/
ActionString = (struct sqlchar *)malloc(strlen(statement)
    + sizeof (struct sqlchar));
ActionString->length = strlen(statement);
strncpy (ActionString->data, statement, strlen(statement));

FileTypeMod = (struct sqlchar *)malloc(strlen("lobsinfile")
    + sizeof (struct sqlchar));
FileTypeMod->length = strlen("lobsinfile");
strncpy (FileTypeMod->data, "lobsinfile", FileTypeMod->length);

/* DELimited format can not have specified names, therefore the
    column method is 'D'efault */
DataDescriptor.dcolmeth = SQL_METH_D;

if (argc == 1) {
    EXEC SQL CONNECT TO sample;
    EMB_SQL_CHECK("CONNECT TO SAMPLE");
}
else if (argc == 3) {
    strcpy (userid, argv[1]);
    strcpy (passwd, argv[2]);
    EXEC SQL CONNECT TO sample USER :userid USING :passwd;
    EMB_SQL_CHECK("CONNECT TO SAMPLE");
}
else {
    printf ("\nUSAGE: tload [userid passwd]\n\n");
    return 1;
} /* endif */

printf ("Exporting EMP_RESUME table into file '%s'\n", datafile);
/*****\
* EXPORT API called *

```

## Beispiele für LOAD-Sitzungen

```
\*****/  
sqluexpr (datafile, NULL, NULL, &DataDescriptor, ActionString,  
          FileType, FileTypeMod, msgfile_x, 0, &outputInfo, NULL, &sqlca);  
EMB_SQL_CHECK("exporting table");  
printf ("Rows exported %d\n", outputInfo.rowsExported);  
  
free (ActionString);  
  
/* need to allocate the proper amount of space for the SQL statement */  
ActionString = (struct sqlchar *)malloc(strlen(impStatement)  
    + sizeof (struct sqlchar));  
ActionString->length = strlen(impStatement);  
strncpy (ActionString->data, impStatement, strlen(impStatement));  
  
printf ("Creating a temporary table 'loadtable' to load into\n");  
/* create a temporary table to import into */  
EXEC SQL CREATE TABLE loadtable (num CHARACTER(6), format VARCHAR(10),  
    photo BLOB(100K));  
EMB_SQL_CHECK("CREATE TABLE");  
  
/* end the transaction so the program can quiesce the tablespace */  
EXEC SQL COMMIT;  
  
printf ("Quiescing tablespaces for table 'loadtable'\n");  
\*****\  
* QUIESCE TABLESPACE FOR TABLE *  
\*****\  
sqluvqdp ("loadtable", SQLU QUIESCEMODE_EXCLUSIVE, NULL, &sqlca);  
EMB_SQL_CHECK("QUIESCE TABLESPACES FOR TABLE");  
  
printf ("Loading the file '%s' into 'loadtable'\n", datafile);  
  
/* initializing the variables for the LOAD API */  
/* the DataFileList structure */  
DataFileList.media_type = SQLU_SERVER_LOCATION;  
DataFileList.sessions = 1;  
DataFileList.target.location = (sqlu_location_entry *) malloc  
    (sizeof(sqlu_location_entry) * DataFileList.sessions);  
strcpy (DataFileList.target.location->location_entry, datafile);  
  
pLobPathList = NULL;  
CallerAction = SQLU_INITIAL;  
  
/* the sqluload input structure */  
InputInfo.sizeOfStruct = SQLULOAD_IN_SIZE; /* this should never change */  
InputInfo.savecnt = 0; /* consistency points as */  
/* infrequently as possible */  
InputInfo.restartcnt = 0; /* start at row 1 */  
InputInfo.rowcnt = 0; /* load all rows */  
InputInfo.warningcnt = 0; /* don't stop for warnings */  
InputInfo.data_buffer_size = 0; /* default data buffer size */  
InputInfo.sort_buffer_size = 0; /* default warning buffer */  
/* size */
```

## Beispiele für LOAD-Sitzungen

```
InputInfo.hold_quiesce = 0;                /* don't hold the quiesce */
InputInfo.restartphase = ' ';             /* ignored anyway, but */
                                           /* must be ' ',L,B,D */
InputInfo.statsopt = SQLU_STATS_NONE;     /* don't bother collecting */
InputInfo.indexing_mode = SQLU_INX_AUTOSELECT; /* let load choose indexing */
                                           /* mode

/* the sqluload output structure */
OutputInfo.sizeOfStruct = SQLULOAD_OUT_SIZE;

/* the CopyTargetList structure */

    pCopyTargetList = NULL;

OutputInfo.sizeOfStruct = SQLULOAD_OUT_SIZE;
\*****\
* LOAD *
\*****/
sqluload (&DataFileList,
          pLobPathList,
          &DataDescriptor,
          ActionString,
          FileType,
          FileTypeMod,
          LocalMsgFileName,
          RemoteMsgFileName,
          CallerAction,
          &InputInfo,
          &OutputInfo,
          pWorkDirectoryList,
          pCopyTargetList,
          pNullIndicators,
          pReserved,
          &sqlca);
EMB_SQL_CHECK("LOADing table");

printf ("Rows loaded %d\nrows committed %d\n", OutputInfo.rowsLoaded,
        OutputInfo.rowsCommitted);

free (ActionString);

/* drop the table */
EXEC SQL DROP TABLE loadtable;

EXEC SQL CONNECT RESET;
EMB_SQL_CHECK("CONNECT RESET");
}
/* end of program : tload.sqc */
```

Die Quellendatei für das Beispielprogramm loadqry.sqc befindet sich im Verzeichnis \sqllib\samples\c. Dieses Beispielprogramm zeigt die Verwendung der API zum Abfragen des aktuellen Status einer Ladeoperation für eine Datenbank, mit der das Programm verbunden ist. Es enthält sowohl DB2-API-Aufrufe als auch Aufrufe mit eingebettetem SQL. Die Prozedurdatei bldvaemb-

## Beispiele für LOAD-Sitzungen

.cmd, die sich im gleichen Verzeichnis befindet, enthält die Befehle zur Erstellung dieses und anderer Beispielprogramme. Basisinformationen zum Erstellen von Anwendungen, die DB2-Verwaltungs-APIs enthalten, und Zusatzinformationen zu Kompilierungs- und Programmverbindungsoptionen finden Sie im Handbuch *Application Building Guide*. Gehen Sie wie folgt vor, um das Beispielprogramm loadqry aus der Quelldatei loadqry.sqc unter OS/2 zu erstellen:

1. Kopieren Sie die Dateien loadqry.sqc, bldvaemb.cmd, util.c und util.h in ein Arbeitsverzeichnis.
2. Setzen Sie, falls der Datenbankmanager nicht aktiv ist, den Befehl db2start ab.
3. Geben Sie bldvaemb loadqry sample ein. Die folgenden Dateien werden generiert:

```
loadqry.bnd  
loadqry.c  
util.obj  
loadqry.obj  
loadqry.exe
```

Geben Sie loadqry ein, um das Beispielprogramm (ausführbare Datei) zu starten. Es kann hilfreich sein, die Nachrichtendatei näher zu betrachten. Diese Datei enthält nur Informationen, wenn das Programm ausgeführt wird, während eine Ladeoperation stattfindet.



---

### Statusangaben für anstehende Aktionen nach einer Ladeoperation

Da keine reguläre Protokollierung erfolgt, verwendet das Dienstprogramm LOAD die Statusangaben *anstehend*, um die Konsistenz der Datenbank zu gewährleisten. Diese Statusangaben können mit dem Befehl LIST TABLES-PACES überprüft werden (siehe Handbuch *Command Reference*).

Die LOAD und BUILD-Phase des LOAD-Prozesses setzen zugehörige Tabellenbereiche in den Status „Laden anstehend“. Führen Sie eine der folgenden Aktionen aus, um den Status „Laden anstehend“ aufzuheben, wenn die Ladeoperation fehlgeschlagen ist oder unterbrochen wurde:

- Starten Sie die Ladeoperation erneut. Ermitteln Sie zunächst die Ursache des Fehlers. Verfügte das Dienstprogramm LOAD beispielsweise nicht über genügend Plattenspeicherplatz, fügen Sie dem Tabellenbereich Behälter hinzu, bevor Sie versuchen, eine LOAD RESTART-Operation auszuführen.
- Beenden Sie die Ladeoperation.
- Rufen Sie eine LOAD REPLACE-Operation für die Tabelle auf, bei der eine Ladeoperation fehlgeschlagen ist.
- Stellen Sie Tabellenbereiche für die Ladetabelle wieder her. Verwenden Sie dazu den Befehl RESTORE DATABASE mit der letzten Sicherung eines Tabellenbereichs oder einer Datenbank, und führen Sie danach weitere Wiederherstellungsfunktionen aus.
- Löschen Sie die Tabellenbereiche für die Ladetabelle, und erstellen Sie sie danach erneut.

Die DELETE-Phase setzt zugehörige Tabellenbereiche in den Status „Löschen anstehend“. Dies kann eintreten, wenn die Ladeoperation während der DELETE-Phase (beim Entfernen ungültiger eindeutiger Schlüssel) fehlschlägt oder unterbrochen wird. Da die Löschvorgänge protokolliert werden, können Fehler auftreten, wenn nicht mehr genügend Speicherplatz für das Protokoll vorhanden ist. Der Status „Löschen anstehend“ kann mit denselben Aktionen aufgehoben werden, mit denen der Status „Laden anstehend“ entfernt wird.

Tabellenbereiche werden in den Status „Sicherung anstehend“ gesetzt, wenn der LOAD-Prozess beendet ist, und folgende Bedingungen erfüllt sind:

- Datenbankkonfigurationsparameter *logretain* ist auf *Recovery* gesetzt, oder *userexit* ist aktiviert, und
- LOAD-Option COPY YES ist nicht angegeben, und
- LOAD-Option NONRECOVERABLE ist nicht angegeben.

## Statusangaben für anstehende Aktionen nach einer Ladeoperation

Der vierte mögliche Status, der dem LOAD-Prozeß zugeordnet ist Status „Überprüfung anstehend“), bezieht sich auf referentielle Integritätsbedingungen und Prüfungen auf Integritätsbedingungen, auf DATALINKS-Integritätsbedingungen, auf AST-Integritätsbedingungen oder auf Integritätsbedingungen für generierte Spalten. Ist beispielsweise eine vorhandene Tabelle eine übergeordnete Tabelle mit einem Primärschlüssel, auf den ein Fremdschlüssel in einer abhängigen Tabelle verweist, wird die abhängige *Tabelle* (nicht der Tabellenbereich) durch das Ersetzen der Daten in der übergeordneten Tabelle in den Status „Überprüfung anstehend“ gesetzt. Setzen Sie zur Auswertung einer Tabelle auf referentielle Integrität und Prüfung auf Integritätsbedingungen nach Beendigung des LOAD-Prozesses die Anweisung SET INTEGRITY ab, wenn die Tabelle im Status „Überprüfung anstehend“ verblieben ist. Weitere Informationen zum Status „Überprüfung anstehend“ finden Sie unter „Überprüfung auf ungültige Integritätsbedingungen“ auf Seite 96.

## Optimieren der Leistung von LOAD

Die Leistung des Dienstprogramms LOAD hängt von der Art und dem Umfang der Daten, der Anzahl der Indizes sowie den angegebenen LOAD-Optionen ab.

Eindeutige Indizes verringern die Leistung von LOAD, wenn doppelte Indizes vorkommen. In den meisten Fällen ist es effizienter, Indizes während der Ladeoperation zu erstellen, als nach Beendigung der Ladeoperation für jeden Index die Anweisung CREATE INDEX aufzurufen (siehe Abb. 5).

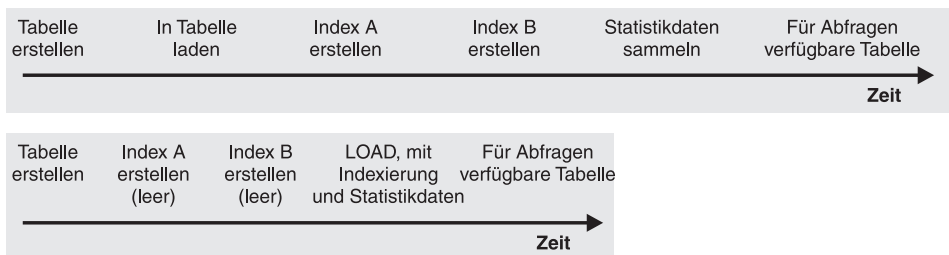


Abbildung 5. Erhöhen der Leistung von LOAD durch gleichzeitige Indexierung und Sammlung von Statistikdaten. Tabellen werden normalerweise in drei Schritten erstellt: Laden von Daten, Erstellen von Indizes und Sammeln von Statistikdaten. Das führt zu mehrfacher Daten-E/A während der Ladeoperation, während der Indexerstellung (es sind mehrere Indizes pro Tabelle möglich) und während der Sammlung von Statistikdaten (die zu E/A bei den Tabellendaten und allen Indizes führt). Eine viel schnellere Methode ist, alle diese Aufgaben vom Dienstprogramm LOAD in einem Datendurchlauf ausführen zu lassen.

Bei der Optimierung der Indexierung wird die Speichermenge, die zum Sortieren der Indexschlüssel während einer Ladeoperation verwendet wird, mit dem Datenbankkonfigurationsparameter *sortheap* gesteuert. Beispiel: Damit vom Dienstprogramm LOAD pro Index 4000 Seiten Hauptspeicher zum Sortieren der Schlüssel verwendet werden, muß der Datenbankkonfigurations-

parameter *sorthheap* auf 4000 Seiten gesetzt werden. Dann müssen alle Anwendungen von der Datenbank getrennt werden, und anschließend muß der Befehl LOAD aufgerufen werden. Wenn ein Index so umfangreich ist, daß er nicht im Hauptspeicher sortiert werden kann, tritt ein Sortierüberlauf ein. Dies bedeutet, daß die Daten auf mehrere „Sortierläufe“ aufgeteilt und in einem temporären Tabellenbereich gespeichert werden, der später zusammengeführt wird. Besteht keine Möglichkeit, einen Sortierüberlauf durch ein Heraufsetzen des Parameters *sorthheap* zu verhindern, muß der Pufferpool für temporäre Tabellenbereiche unbedingt groß genug sein, um den Umfang der durch den Überlauf verursachten Platten-E/A zu minimieren. Es empfiehlt sich des weiteren, temporäre Tabellenbereiche mit mehreren Containern zu deklarieren, die sich jeweils auf unterschiedlichen Platteneinheiten befinden, damit während der Zusammenführung der Sortierläufe ein E/A-Parallelismus erzielt wird.

Die Leistung von LOAD kann durch Installieren von Hochleistungssortierbibliotheken anderer Hersteller verbessert werden, um Indizes während der Ladeoperation zu erstellen. Ein Beispiel für Sortierprogramme anderer Hersteller ist das Produkt SyncSort. Verwenden Sie die Umgebungsvariable **DB2SORT** (Registrierungswert), um die Speicherposition der Sortierbibliothek anzugeben, die zur Laufzeit geladen werden soll. Weitere Informationen zu Umgebungsvariablen finden Sie im Handbuch *Systemverwaltung*.

Bei Verwendung der Anweisung SET INTEGRITY verlängert sich die Zeitspanne, die insgesamt erforderlich ist, um eine Tabelle zu laden und wieder verwendbar zu machen. Wenn alle Ladeoperationen im Modus INSERT ausgeführt werden, wird die Tabelle mit der Anweisung SET INTEGRITY inkrementell auf ungültige Integritätsbedingungen überprüft, indem nur der angefügte Teil der Tabelle überprüft wird. Wenn die Tabelle nicht inkrementell auf ungültige Integritätsbedingungen überprüft werden kann, wird die vollständige Tabelle überprüft, und es kann einige Zeit dauern, bis die Tabelle wieder verwendet werden kann.

Das Dienstprogramm LOAD arbeitet in den Modi INSERT und REPLACE annähernd gleich gut.

Das Dienstprogramm liefert die bestmögliche Leistung, indem optimale Werte für DISK\_PARALLELISM, CPU\_PARALLELISM und DATA\_BUFFER ermittelt werden, wenn diese Parameter nicht vom Benutzer angegeben wurden. Die Optimierung erfolgt auf der Grundlage der Größe und des freien Speicherbereichs, der im Zwischenspeicher der Dienstprogramme verfügbar ist. Ziehen Sie in Betracht, das Dienstprogramm LOAD die Werte für diese Parameter wählen zu lassen und anschließend zu versuchen, die Parameterwerte für Ihre speziellen Anforderungen zu optimieren.

## Optimieren der Leistung von LOAD

Die folgenden Informationen beziehen sich auf die Auswirkungen der verschiedenen, beim Dienstprogramm LOAD verfügbaren Optionen auf die Leistung:

### **ANYORDER**

Geben Sie diesen Dateitypwert an, um die Beibehaltung der Reihenfolge der geladenen Daten auszusetzen und die Leistung zu verbessern. Wenn die zu ladenden Daten bereits vorsortiert sind, beschädigt die Angabe von anyorder möglicherweise die Vorsortierung, und die Vorteile der Vorsortierung für nachfolgende Abfragen gehen verloren.

### **BINARY NUMERICS und PACKED DECIMAL**

Verwenden Sie diese Dateitypwerte, um die Leistung beim Laden von positionsgebundenen numerischen ASC-Daten in Sätze mit fester Länge zu verbessern.

### **COPY YES oder NO**

Verwenden Sie diesen Parameter, um anzugeben, ob während einer Ladeoperation eine Kopie der Eingabedaten erstellt werden soll. COPY YES verringert die Leistung von LOAD, weil alle Ladedaten während der Ladeoperation kopiert werden (Vorwärtswiederherstellung muß aktiviert sein). Der Zuwachs an E/A-Vorgängen kann eine längere Ladezeit auf einem E/A-abhängigen System bewirken. Das Angeben mehrerer Einheiten oder Verzeichnisse (auf verschiedenen Platten) kann die durch diese Operation verursachte Leistungseinbuße ausgleichen. COPY NO kann zu einer geringeren Gesamtleistung führen, da die Tabelle bei aktivierter Vorwärtswiederherstellung in den Status „Sicherung anstehend“ gesetzt wird und die Datenbank bzw. ausgewählte Tabellenbereiche gesichert werden müssen, bevor auf die Tabelle zugegriffen werden kann.

### **CPU\_PARALLELISM**

Mit diesem Parameter können Sie die partitionsinterne Parallelität ausschöpfen (wenn diese von Ihrer Maschine unterstützt wird) und die Leistung von LOAD signifikant verbessern. Mit diesem Parameter wird die Anzahl von Prozessen bzw. Threads angegeben, die vom Dienstprogramm LOAD zum syntaktischen Analysieren, Umsetzen und Formatieren von Datensätzen verwendet werden. Der maximal zulässige Wert ist 30. Wenn für den angegebenen Wert zu wenig Speicher verfügbar ist, wird der Wert vom Dienstprogramm angepaßt. Wenn dieser Parameter nicht angegeben wird, wählt das Dienstprogramm LOAD einen Standardwert auf Grundlage der Anzahl CPUs auf dem System aus.

Die Reihenfolge der Datensätze in den Quelldaten (siehe Abb. 6 auf Seite 177) bleibt unabhängig vom Wert dieses Parameters erhalten.

Wenn Tabellen Daten des Typs LOB oder LONG VARCHAR enthalten, wird CPU\_PARALLELISM auf den Wert 1 gesetzt. In diesem Fall wird Parallelität nicht unterstützt.

Die Verwendung dieses Parameters ist zwar nicht auf SMP-Hardware begrenzt, aber bei Verwendung in Nicht-SMP-Umgebungen wird möglicherweise keine erkennbare Leistungsverbesserung erzielt.



Abbildung 6. Satzreihenfolge der Quelldaten wird beibehalten, wenn partitionsinterne Parallelität bei einer Ladeoperation ausgeschöpft wird

### DATA BUFFER

Mit dem Parameter DATA BUFFER wird die Gesamtmenge von Speicher angegeben, die dem Dienstprogramm LOAD als Puffer zugeordnet wird. Es wird empfohlen, diesen Puffer auf ein Vielfaches von *extentsz* einzustellen. Ein Speicherbereich ist die Einheit für das Versetzen von Daten innerhalb von DB2. Die Größe des Speicherbereichs kann eine oder mehr 4 KB-Seiten betragen. Der Parameter DATA BUFFER ist beim Arbeiten mit großen Objekten (LOBs) nützlich; er setzt die E/A-Wartezeit herab. Der Datenpuffer wird aus dem Zwischenspeicher der Dienstprogramme zugeordnet. Je nach der Menge des verfügbaren Speichers auf dem System sollten Sie das Zuordnen von mehr Speicher zur Verwendung durch die DB2-Dienstprogramme in Erwägung ziehen. Der Datenbankkonfigurationsparameter *util\_heap\_sz* kann entsprechend geändert werden. Weitere Informationen zum Befehl UPDATE DATABASE CONFIGURATION finden Sie im Handbuch *Command Reference*. Der Standardwert für den Konfigurationsparameter *util\_heap\_sz* ist 5 000 4 KB-Seiten. Da LOAD nur eines der zahlreichen Dienstprogramme ist, die Speicher aus dem Zwischenspeicher für Dienstprogramme verwenden, wird empfohlen, nicht mehr als fünfzig Prozent der Seiten, die durch diesen Parameter definiert sind, für das Dienstprogramm LOAD zur Verfügung zu stellen. Darüber hinaus muß der Zwischenspeicher für Dienstprogramme groß genug definiert werden. Weitere Informationen zu *util\_heap\_sz* finden Sie im Handbuch *Systemverwaltung*.

### DISK\_PARALLELISM

Mit dem Parameter DISK\_PARALLELISM wird die Anzahl von Prozessen bzw. Threads angegeben, die vom Dienstprogramm LOAD zum Schreiben von Datensätzen auf die Platte verwendet werden. Mit diesem Parameter können Sie verfügbare Behälter beim Laden von Daten ausschöpfen und die Leistung von LOAD signifikant verbessern. Der maximal zulässige Wert ist der größere Wert von 50 oder

## Optimieren der Leistung von LOAD

des Vierfachen von CPU\_PARALLELISM. DISK\_PARALLELISM entspricht standardmäßig der Summe der Tabellenbereichsbehälter in allen Tabellenbereichen, die Objekte für die zu ladende Tabelle enthalten, außer wenn dieser Wert den maximal zulässigen Standardwert übersteigt.

### **FASTPARSE**

Mit dem Dateitypwert fastparse können Sie die Datenprüfung verringern, die für vom Benutzer bereitgestellte Spaltenwerte ausgeführt wird, und die Leistung verbessern. Diese Option sollte verwendet werden, wenn an der Gültigkeit der zu ladenden Daten kein Zweifel besteht. Sie kann zu einer um 10 oder 20 Prozent besseren Leistung beitragen.

### **NONRECOVERABLE**

Verwenden Sie diesen Parameter, wenn LOAD-Transaktionen für eine Tabelle nicht wiederherstellbar sein müssen. Die Leistung von LOAD wird verbessert, weil neben dem Versetzen der Daten in die Tabelle keine zusätzlichen Vorgänge erforderlich sind und die Ladeoperation durchgeführt wird, ohne daß die Tabellenbereiche im Status „Sicherung anstehend“ verbleiben.

**Anmerkung:** Wenn diese LOAD-Transaktionen bei einer nachfolgenden Wiederherstellung und aktualisierenden Wiederherstellung vorgefunden werden, wird die Tabelle nicht aktualisiert und als „ungültig“ gekennzeichnet. Weitere Funktionen für diese Tabelle werden ignoriert. Nach der Beendigung der aktualisierenden Wiederherstellung kann die Tabelle nur gelöscht werden.

### **NOROWWARNINGS**

Verwenden Sie den Dateitypwert norowarnings, um die Aufzeichnung von Warnungen zu zurückgewiesenen Zeilen zu unterdrücken und die Leistung zu verbessern, wenn Sie eine große Zahl von Warnungen erwarten.

### **SAVECOUNT**

Verwenden Sie diesen Parameter, um ein Intervall für die Festlegung von Konsistenzzuständen während einer Ladeoperation einzustellen. Die Synchronisation der Vorgänge zum Herstellen eines Konsistenzzustands dauert einige Zeit. Wenn dies zu häufig geschieht, sinkt die Leistung von LOAD merklich. Wenn eine sehr große Zahl von Zeilen geladen werden muß, sollte ein großer Wert für SAVECOUNT angegeben werden (z. B. bei einer Ladeoperation mit 100 Millionen Sätzen ein Wert von 10 Millionen).

Eine LOAD RESTART-Operation wird automatisch vom letzten Konsistenzzustand ausgehend fortgesetzt.

### STATISTICS YES

Verwenden Sie diesen Parameter, um Datenverteilungs- und Indexstatistikdaten effizienter zu sammeln als durch den Aufruf des Dienstprogramms RUNSTATS nach Beendigung der Ladeoperation. Die Leistung der Ladeoperation selbst hingegen wird verringert (vor allem, wenn DETAILED INDEXES ALL angegeben ist).

Die Leistungsstärke der Anwendungen ist am höchsten, wenn sie die bestmöglichen Verteilungs- und Indexstatistikdaten verwenden. Nach der Aktualisierung der Statistikdaten können die Anwendungen die neuen Zugriffspfade zu den Tabellendaten auf der Grundlage der letzten Statistikdaten verwenden. Neue Zugriffspfade zu einer Tabelle können erstellt werden, indem Sie die Anwendungspakete mit dem DB2-Befehl BIND erneut binden (siehe Handbuch *Command Reference*).

Beim Laden von Daten in große Tabellen empfiehlt es sich, einen größeren Wert für den Datenbankkonfigurationsparameter *stat\_heap\_sz* anzugeben. Weitere Informationen zum Befehl UPDATE DATABASE CONFIGURATION finden Sie im Handbuch *Command Reference*. Weitere Informationen zu *stat\_heap\_sz* finden Sie im Handbuch *Systemverwaltung*.

### WARNINGCOUNT

Verwenden Sie diesen Parameter, um die Anzahl von Warnungen anzugeben, die vom Dienstprogramm zurückgegeben werden kann, bevor eine Ladeoperation beendet wird. Wenn Sie mit nur wenigen oder keinen Warnungen rechnen, stellen Sie den Parameter WARNINGCOUNT auf annähernd die erwartete Anzahl ein. Stellen Sie ihn auf 20 ein, wenn Sie keine Warnungen erwarten. Die Ladeoperation wird nach dem Erreichen des Werts für WARNINGCOUNT gestoppt. Damit haben Sie die Möglichkeit, die Daten zu korrigieren (bzw. zu löschen und die zu ladende Tabelle erneut zu erstellen), bevor Sie versuchen, die Ladeoperation zu beenden. Die Definition einer WARNINGCOUNT-Schwelle hat zwar keine direkte Auswirkung auf die Leistung der Ladeoperation, kann jedoch verhindern, daß Sie erst nach Beendigung der gesamten Ladeoperation feststellen, daß ein Fehler vorliegt.

### Rahmenbedingungen und Einschränkungen

Für das Dienstprogramm LOAD gelten die folgenden Rahmenbedingungen:

- Dieses Dienstprogramm bietet keine Unterstützung für Kurznamen.
- Das Laden von Daten in typisierte Tabellen oder in Tabellen mit Spalten strukturierten Typs wird nicht unterstützt.
- Das Laden von Daten in deklarierte temporäre Tabellen wird nicht unterstützt.
- Versuche, Tabellen in einem Tabellenbereich im Status „Laden anstehend“ zu erstellen oder zu löschen, schlagen fehl.
- Sie können keine Daten in eine Datenbank laden, auf die über DB2 Connect oder einen Server einer älteren Version als DB2 Version 2 zugegriffen wird. Optionen, die nur mit diesem Release von DB2 verfügbar sind, können nicht mit einem Server aus einem früheren Release verwendet werden.
- Wenn während einer LOAD REPLACE-Operation ein Fehler auftritt, gehen die Originaldaten in der Tabelle verloren. Behalten Sie eine Kopie der Eingabedaten, damit die Ladeoperation erneut gestartet werden kann.
- Für neu geladene Zeilen werden keine Auslöser aktiviert. Geschäftsregeln, die Auslösern zugeordnet sind, werden vom Dienstprogramm LOAD nicht umgesetzt.

---

### Fehlerbehebung

Bei DB2-Operationen wie dem Exportieren, Importieren, Laden, Binden oder Wiederherstellen von Daten können Sie angeben, daß Nachrichtendateien erstellt werden, die Fehlnachrichten, Warnungen und informative Nachrichten zu diesen Operationen enthalten. Geben Sie den Namen dieser Dateien mit dem Parameter MESSAGES an.

Diese Nachrichtendateien sind gewöhnliche ASCII-Dateien. Sie können sie mit dem bei Ihrem Betriebssystem üblichen Verfahren drucken oder mit einem beliebigen ASCII-Editor anzeigen.

#### **Anmerkungen:**

1. Sie können den Inhalt einer Nachrichtendatei erst anzeigen, wenn die Operation beendet ist.
2. Jede Nachricht in einer Nachrichtendatei beginnt in einer neuen Zeile und enthält Informationen, die von der DB2-Funktion zum Abrufen von Nachrichten bereitgestellt werden.



---

## Kapitel 4. AutoLoader

In diesem Kapitel wird das DB2 UDB-Dienstprogramm AutoLoader beschrieben, das in einer partitionierten Datenbankumgebung dazu verwendet werden kann, Daten gleichzeitig in alle oder bestimmte Partitionen zu laden.

Die folgenden Themen werden behandelt:

- „AutoLoader - Übersicht“
- „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von AutoLoader“ auf Seite 182
- „Verwenden von AutoLoader“ auf Seite 183
- „Laden in mehrere Datenbankpartitionen“ auf Seite 184
- „AutoLoader-Optionen“ auf Seite 185
- „AutoLoader-Beispielsitzung“ auf Seite 194
- „Migration und Kompatibilität zur früheren Version“ auf Seite 196
- „Hinweise und Tips für das Dienstprogramm AutoLoader“ auf Seite 197
- „Rahmenbedingungen und Einschränkungen“ auf Seite 199
- „Fehlerbehebung für AutoLoader“ auf Seite 199.

---

### AutoLoader - Übersicht

Mit dem Dienstprogramm AutoLoader können folgende Funktionen ausgeführt werden:

- Übertragen von Daten von einem System (wie MVS) auf ein anderes System (wie UNIX)
- Paralleles Partitionieren dieser Daten
- Gleichzeitiges Laden der Daten in die entsprechenden Datenbankpartitionen

AutoLoader kann in einem von vier Betriebsmodi ausgeführt werden:

- **SPLIT\_AND\_LOAD**. Die Daten werden partitioniert (vielleicht in Parallelverarbeitung) und gleichzeitig in die entsprechenden Datenbankpartitionen geladen.
- **SPLIT\_ONLY**. Die Daten werden partitioniert (vielleicht in Parallelverarbeitung), und die Ausgabe wird in Dateien an einer angegebenen Speicherposition oder im aktuellen AutoLoader-Arbeitsverzeichnis geschrieben.
- **LOAD\_ONLY**. Die Daten müssen bereits partitioniert sein. Der Teilungsprozeß wird übersprungen, und die Daten werden gleichzeitig in die entsprechenden Datenbankpartitionen geladen.

## AutoLoader - Übersicht

- ANALYZE. Es wird eine optimale Partitionierungszuordnung mit einer gleichmäßigen Datenverteilung über alle Datenbankpartitionen generiert.

In einer partitionierten Datenbank werden große Datenmengen partitionsübergreifend angeordnet. Mit Partitionierungsschlüsseln können Sie die Datenbankpartition bestimmen, in der die einzelnen Datenteile gespeichert sind. Die Daten müssen *geteilt* werden, bevor sie in die richtige Datenbankpartition geladen werden können. Mit dem Dienstprogramm AutoLoader können beide Operationen ausgeführt werden (siehe Abb. 7).

Das Dienstprogramm AutoLoader verwendet den Hash-Algorithmus, um die Daten zu partitionieren und auf so viele Ausgabe-Sockets zu verteilen, wie Datenbankpartitionen in der Knotengruppe vorhanden sind, in der die Tabelle definiert wurde. Anschließend lädt es diese Ausgabe-Sockets gleichzeitig in die Datenbankpartitionen der Knotengruppe. Ein wesentliches Merkmal des Dienstprogramms AutoLoader ist, daß es direkte TCP/IP-Kommunikation mit Sockets für die gesamte Datenübertragung verwendet, die für den Teilungs- und Ladeprozeß erforderlich ist. Es verwendet außerdem mehrere Datenbankpartitionen für die Teilungsphase, wodurch die Leistung erheblich verbessert wird.

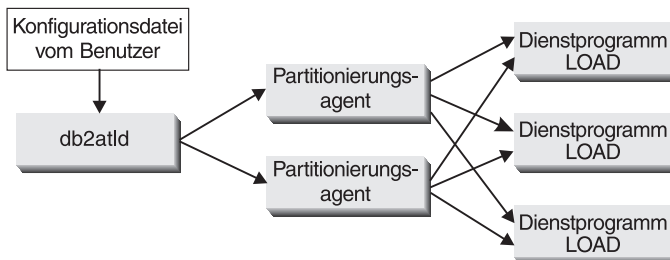


Abbildung 7. AutoLoader - Übersicht. In diesem Beispiel werden Quelldaten von AutoLoader gelesen, und jeweils die Hälfte der Daten wird an die beiden Partitionierungsagenten gesendet, die die Daten partitionieren und an eine von drei Datenbankpartitionen senden. Die Daten werden mit dem Dienstprogramm LOAD auf jeder Partition geladen.

---

## Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von AutoLoader

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie Operationen von Dienstprogrammen und zur Datenbankmanagerpflege auf höherer Ebene zu gruppieren. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf solche Objekte zugreifen, für die sie die entsprechende Berechtigung besitzen, d. h., für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

## Erforderliche Berechtigungen zur Verwendung von AutoLoader

Zur Verwendung des Dienstprogramms AutoLoader ist die gleiche Berechtigung erforderlich wie zur Verwendung des Dienstprogramms LOAD (siehe „Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von LOAD“ auf Seite 89). Zudem muß der Exemplareigner Schreibzugriff auf die LOAD-Speicherauszugsdatei und die temporäre Datei für das Dienstprogramm LOAD haben.

---

### Verwenden von AutoLoader

#### Vor der Verwendung von AutoLoader

Bevor Sie das Dienstprogramm AutoLoader aufrufen, müssen Sie folgende Schritte ausführen:

1. Erstellen Sie ein temporäres Arbeitsverzeichnis. Alle teilnehmenden Datenbankpartitionen müssen auf dieses Verzeichnis zugreifen können. Von diesem Verzeichnis aus wird das Dienstprogramm AutoLoader aufgerufen.
2. Ändern Sie die AutoLoader-Konfigurationsdatei (siehe „AutoLoader-Beispielsitzung“ auf Seite 194), und kopieren Sie sie ins Arbeitsverzeichnis.
3. Stellen Sie sicher, daß der Konfigurationsparameter *svcname* des Datenbankmanagers und die Profilregistrierdatenbankvariable **DB2COMM** richtig definiert sind. Dies ist wichtig, da das Dienstprogramm AutoLoader ferne Datenbankverbindungen von der Arbeitspartition (von der das Dienstprogramm aufgerufen wird) zu den Datenbankpartitionen (in denen die Tabelle definiert ist) herstellt.

#### Aufrufen von AutoLoader

Das Dienstprogramm AutoLoader wird mit dem Befehl **db2atld** aufgerufen.

```
db2atld [-config konfigurationsdatei] [-restart] [-terminate]
```

Dabei gilt folgendes:

```
"-config konfigurationsdatei" gibt eine AutoLoader-Konfigurationsdatei an (standardmäßig "autoloader.cfg");  
"-restart" fordert einen Neustart einer unterbrochenen AutoLoader-Operation an (die Konfigurationsdatei muß für den Neustart nicht geändert werden);  
"-terminate" fordert die Beendigung einer unterbrochenen AutoLoader-Operation an.
```

Eine Beispielkonfigurationsdatei, *autoloader.cfg*, befindet sich im Verzeichnis *sqlib/samples/autoloader*. Es wird empfohlen, die Beispielkonfigurationsdatei zu kopieren, umzubenennen und anzupassen, um mit dem Dienstprogramm die gewünschten Operationen auszuführen.

## Laden in mehrere Datenbankpartitionen

---

### Laden in mehrere Datenbankpartitionen

Wenn Sie Daten in eine Tabelle in einer Knotengruppe mit mehreren Datenbankpartitionen laden, setzt das Dienstprogramm LOAD voraus, daß die zu ladenden Dateien geteilt sind und die richtigen Kopfdaten enthalten. Das Dienstprogramm LOAD prüft die Kopfdaten, die von der AutoLoader-Teilungsoperation in jede Datendatei geschrieben werden, um sicherzustellen, daß die Daten an die richtige Speicherposition geschrieben werden.

Wenn Sie Daten in eine Tabelle in einer Knotengruppe mit nur einer einzigen Datenbankpartition laden, brauchen die Dateien nicht geteilt zu sein, auch wenn die Tabelle mit einem Partitionierungsschlüssel definiert ist. In diesem Fall müßten Sie den Wert `noheader` für die Ladeoperation angeben.

Das Dienstprogramm LOAD prüft, ob die Partitionierungszuordnung, die bei der AutoLoader-Teilungsoperation verwendet wurde, dieselbe ist wie die, die beim Laden der Tabelle angegeben wird. Ist dies nicht der Fall, wird ein Fehler zurückgegeben. Es prüft außerdem, ob die Dateipartition in die richtige Datenbankpartition geladen wird und ob die Datentypen der Spalten des Partitionierungsschlüssels, der bei der Teilung angegeben wurde, mit der aktuellen Definition im Katalog übereinstimmt. Die Daten der Knotengruppe, in die die Tabelle geladen wird, dürfen zwischen dem Zeitpunkt, zu dem die Datendatei partitioniert wird, und dem Zeitpunkt, zu dem die Teile in die entsprechende Datenbanktabelle geladen werden, nicht umverteilt werden. Wenn eine Umverteilung durchgeführt wurde, kann das Dienstprogramm die partitionierten Daten nicht laden.

Das Dienstprogramm LOAD unterstützt die folgenden unstrukturierten Dateiformate:

- ASC (ASCII-Format mit universellen Zeilenbegrenzern)
- DEL (ASCII-Format ohne universelle Zeilenbegrenzer)
- PC/IXF

AutoLoader kann nur zum Partitionieren von ASC- und DEL-Dateien verwendet werden. PC/IXF-Dateien können nicht geteilt werden, sie können jedoch in eine Knotengruppe mit einer einzelnen Datenbankpartition geladen werden. Dazu muß bei der Ladeoperation der Wert `noheader` angegeben werden. Wenn Sie eine IXF-Datei in eine Mehrpartitionstabelle laden wollen, müssen Sie sie zunächst in eine Einzelpartitionstabelle laden und anschließend die Daten durch Verwendung einer Anweisung `SELECT` aus der Einzelpartitionstabelle in die Mehrpartitionstabelle einfügen.

Die Klausel `LOAD ROWCOUNT` wird in einer AutoLoader-Operation nicht unterstützt. Dieser Parameter ist nur in einer Umgebung ohne partitionierte

Datenbanken gültig. Die Klausel LOAD SAVECOUNT wird nicht unterstützt, wenn in einer AutoLoader-Operation mehrere Verteilerprozesse verwendet werden.

Wenn eine Spalte, die Teil des Partitionierungsschlüssels ist, ungültig ist oder zurückgewiesen wurde, werden keine zu dieser Zeile gehörigen Daten geladen. Die Zeile wird nicht in die *Speicherauszugsdatei* geschrieben, auch wenn eine solche angegeben wurde. Statt dessen wird eine Nachricht, die angibt, daß der Datensatz zurückgewiesen wurde, in die Protokolldatei des Verteilerprozesses geschrieben. Sie sollten die Protokolldatei des Verteilerprozesses nach Beendigung des AutoLoader-Prozesses überprüfen.

---

### AutoLoader-Optionen

In einer AutoLoader-Konfigurationsdatei kann eine Vielzahl von Optionen angegeben werden.

#### **RELEASE-Stand**

Der Release-Stand dieser Konfigurationsdatei. Diese Zeile der Konfigurationsdatei bitte nicht löschen oder ändern.

#### **Befehl LOAD**

Die wichtigste Komponente der Konfigurationsdatei ist der Befehl LOAD. AutoLoader erfordert den Befehl LOAD zur Steuerung der Datenverarbeitung, auch wenn der ausgewählte Betriebsmodus keine Ladeoperation vorsieht. AutoLoader extrahiert beispielsweise nützliche Informationen aus dem Befehl LOAD, auch wenn nur eine Operation im Modus SPLIT\_ONLY ausgeführt wird. Die Parameter des Befehls LOAD geben den Zieltabellennamen an und spezifizieren, woher die Daten stammen, um welchen Typ von Daten es sich handelt (z. B. ASCII-Format ohne universelle Zeilenbegrenzer) und wie die Daten geladen werden sollen.

Geben Sie unbedingt einen vollständigen Befehl LOAD mit Schemenname, Dateiname, Dateityp und Tabellename an. Das Dienstprogramm AutoLoader erfordert außerdem, daß der Befehl LOAD dem Format der Datei „db2 -f“ entspricht. Davon ausgenommen ist das zusätzlich vorangestellte Schlüsselwort „db2“. Weitere Informationen zu dieser und zu anderen Optionen des Befehlszeilenprozessors (CLP) finden Sie im Handbuch *Command Reference*. Es ist nicht erforderlich, die Escape-Shell-Sonderzeichen zu verwenden. Wenn das letzte Zeichen in einer Zeile ein umgekehrter Schrägstrich (\) ist, gilt die nächste Zeile als Fortsetzung der aktuellen Zeile. In diesem Fall werden der umgekehrte Schrägstrich und das Zeilenendezeichen ignoriert.

Das Dienstprogramm AutoLoader gibt einen Fehler zurück, wenn im Befehl LOAD das Schlüsselwort CLIENT angegeben ist.

## AutoLoader-Optionen

Weitere Informationen zu allen Parametern des Befehls LOAD finden Sie in „LOAD - Befehl“ auf Seite 101.

### Parameter DATABASE

Dieser Parameter wird verwendet, um die Datenbank anzugeben, in die die Daten geladen werden sollen. Wenn kein Name angegeben wird, wird SAMPLE als Standardwert verwendet.

### Parameter HOSTNAME

Dieser Parameter gibt den Namen des fernen Rechners an, auf dem sich die Datendatei befindet. Dieser Rechner kann ein MVS-Host oder eine andere Workstation sein. Wenn dieser Parameter nicht und der Parameter FILE\_TRANSFER\_CMD angegeben ist, wird der Host-Name nohost im Argument <host-name> an den Parameter FILE\_TRANSFER\_CMD übergeben. Diesem Parameter ist kein Standardwert zugeordnet.

### Parameter FILE\_TRANSFER\_CMD

Von der Vorgängerversion von AutoLoader wurde die Funktion der Host-Dateiübertragung unterstützt, d. h., das Dienstprogramm AutoLoader konnte zur Übertragung von Datendateien von einem fernen Host konfiguriert werden. Diese Option wurde durch die Option FILE\_TRANSFER\_CMD ersetzt. Dieser Parameter gibt den vollständig qualifizierten Namen einer ausführbaren Datei, Stapeldatei oder Prozedurdatei an, die zur Übertragung von Daten von einem fernen Host verwendet wird. AutoLoader muß auf den Pfad zugreifen können. Der gesamte Pfad, einschließlich des Programmdateinamens, darf nicht länger als 254 Zeichen sein.

Vor dem Aufruf der angegebenen Datei richtet AutoLoader in Erwartung der vom Host gesendeten Daten benannte Pipes ein. Die Anzahl der benannten Pipes, die erstellt werden, entspricht der Anzahl der in der Klausel FROM des Befehls LOAD aufgeführten Dateien oder Einheiten. Diese Informationen aus dem Befehl LOAD werden auch verwendet, um die Parameter anzugeben, die der ausführbaren Datei, Stapeldatei oder Prozedurdatei übergeben werden.

AutoLoader erstellt anhand dieser Informationen den folgenden Befehl:

```
<BEFEHL> <protokollpfad> <host-name> <pipe-basisdateiname>  
<anzdatenträger> <quellendatenträgerliste>
```

Dabei gilt folgendes:

- <BEFEHL> ist der vollständig qualifizierte Pfad zu einer ausführbaren Datei, Stapel- oder Prozedurdatei, die verwendet wird, um Daten vom Host zu versetzen.

Die übrigen Angaben sind Parameter, die von dem Befehl verwendet werden können:

- <protokollpfad> ist der Protokollpfad von AutoLoader. Mit dem <BEFEHL> können Diagnosedaten oder temporäre Daten in diesen Pfad geschrieben werden.
- <host-name> ist der mit dem Parameter HOSTNAME angegebene Host-Name.
- <pipe-basisdateiname> ist der Basisdateiname für benannte Pipes, die von AutoLoader erstellt werden. Das Dienstprogramm AutoLoader generiert den Basisdateinamen und stellt sicher, daß er auf dem System eindeutig ist. An den Basisdateinamen werden vom Dienstprogramm Zeichen angefügt, um die erforderlichen benannten Pipes zu erstellen.
- <anzdatenträger> ist die Anzahl der Dateien oder Einheiten, die Daten bereitstellen (in der Klausel FROM des Befehls LOAD aufgeführt).
- <quellendatenträgerliste> enthält die Namen aller Dateien oder Einheiten, die Daten bereitstellen (in der Klausel FROM des Befehls LOAD aufgeführt). Die Namen sind durch doppelte Anführungszeichen begrenzt, um Probleme zu vermeiden, die durch Sonderzeichen in Namen verursacht werden können.

Eine AIX-Beispieldatei namens atldftp.drvc befindet sich im Verzeichnis sqllib/samples/autoloader. Das Beispiel zeigt, wie Daten mit FTP von einem fernen Host versetzt werden können.

### Parameter SPLIT\_FILE\_LOCATION

Dieser Parameter wird auf zwei Arten verwendet:

- Zur Angabe des Pfades zur Speicherposition der Teildateien, wenn das Dienstprogramm im Modus LOAD\_ONLY arbeitet.
- Zur Angabe des Pfades zu der Speicherposition, in die die partitionierten Dateien geschrieben werden sollen (wenn das Dienstprogramm im Modus SPLIT\_ONLY arbeitet).

Wenn für diesen Parameter kein Wert angegeben wird und das Dienstprogramm im Modus SPLIT\_ONLY arbeitet, werden die Teildateien in das aktuelle Arbeitsverzeichnis geschrieben. Arbeitet das Dienstprogramm im Modus LOAD\_ONLY, werden die Teildateien im aktuellen Arbeitsverzeichnis gesucht.

Wenn der Parameter SPLIT\_FILE\_LOCATION auf ein NFS-Verzeichnis verweist, das nach einer SPLIT\_ONLY-Operation partitionsübergreifend angehängt wird, ist eine LOAD\_ONLY-Operation in der Lage, ohne ein Eingreifen des Benutzers auf die Dateien mit den geteilten Daten zuzugreifen. Verweist der Parameter SPLIT\_FILE\_LOCATION jedoch nicht auf ein NFS-Verzeichnis nach einer SPLIT\_ONLY-

## AutoLoader-Optionen

Operation, müssen die Dateien mit den geteilten Daten durch Benutzer manuell aus jeder Partition in das Verzeichnis versetzt werden, in dem die LOAD\_ONLY-Operation ausgeführt wird.

### Parameter OUTPUT\_NODES

Dieser Parameter gibt die Datenbankpartitionen an, für die die Ladeoperation ausgeführt werden soll. Die angegebenen Partitionsnummern müssen eine Untermenge der Datenbankpartitionen sein, in denen die Tabelle definiert ist. Standardmäßig werden in alle Datenbankpartitionen, in denen die Tabelle definiert ist, Daten geladen.

### Parameter SPLIT\_NODES

Mit diesem Parameter werden die Datenbankpartitionen angegeben, die am Teilungsprozeß teilnehmen. Diese Datenbankpartitionen können mit den Datenbankpartitionen, die geladen werden, identisch sein oder sich von ihnen unterscheiden. Wenn für diesen Parameter kein Wert angegeben ist, wird von AutoLoader ermittelt, wie viele Partitionen für die Teilungsoperation erforderlich sind und welche Partitionen verwendet werden, um optimale Leistungswerte zu erzielen. Die folgenden Regeln werden verwendet, um zu bestimmen, wie viele Partitionen für die Teilungsoperation erforderlich sind:

- Wenn der Wert ANYORDER im Befehl LOAD nicht angegeben ist, kann nur ein einziger Verteilerprozeß in der AutoLoader-Sitzung verwendet werden, und es gilt folgendes:
  - Wenn mit dem Parameter OUTPUT\_NODES nur eine einzige Partition angegeben wird oder die Arbeitspartition von AutoLoader nicht Teil des mit dem Parameter OUTPUT\_NODES angegebenen Werts ist, wird die Arbeitspartition von AutoLoader als Teilungspartition verwendet.
  - Andernfalls wird die erste außer der AutoLoader-Arbeitspartition im Parameter OUTPUT\_NODES enthaltene Partition als Teilungspartition verwendet.
- Wenn der Wert anyorder im Befehl LOAD angegeben ist, gilt folgendes:
  1. Die Anzahl der Teilungspartitionen wird wie folgt ermittelt:  
$$(\text{Anzahl der Partitionen in OUTPUT\_NODES})/4 + 1$$
  2. Diese Anzahl von Partitionen wird aus den Partitionen gewählt, die mit dem Parameter OUTPUT\_NODES angegeben wurden. Davon ausgenommen ist die AutoLoader-Arbeitspartition.

### Parameter RUN\_STAT\_NODE

In Verbindung mit der Angabe STATISTICS YES im Befehl LOAD können Sie die Datenbankpartition angeben, auf der Statistikdaten gesammelt werden sollen. Wenn kein Wert oder der Wert -1 angegeben wird, wird standardmäßig die erste Datenbankpartition in der Ausgabepartitionsliste verwendet.



**Parameter MODE**

Dieser Parameter bestimmt den Modus, in dem das Dienstprogramm AutoLoader ausgeführt werden soll. Gültige Werte sind: SPLIT\_AND\_LOAD (Standardwert), SPLIT\_ONLY, LOAD\_ONLY oder ANALYZE.

**SPLIT\_AND\_LOAD**

In diesem Modus werden die Daten partitioniert und anschließend in die richtigen Datenbankpartitionen geladen. Daten werden durch direkte TCP/IP-Kommunikation mittels Sockets übertragen. Es sind mehrere Eingabedateien zulässig.

**SPLIT\_ONLY**

In diesem Modus werden die Daten nur geteilt. Eine Gruppe von Dateien mit den geteilten Daten wird für die angegebenen Datenbankpartitionen generiert. Sie müssen über ausreichend Speicher für jede der Dateien mit den geteilten Daten verfügen. Die Teilungsfunktion schreibt die Dateien in die Speicherposition, die mit dem Parameter *SPLIT\_FILE\_LOCATION* angegeben wurde, oder in das aktuelle Arbeitsverzeichnis. In die Verzeichnisposition muß geschrieben werden können. Die Daten werden in separate Dateien partitioniert, die nach der Konvention *dateiname.xxx* benannt werden, wobei *xxx* die Nummer der Partition ist, zu der die Teildatei gehört. Wenn es im Befehl LOAD mehrere Eingabedatendateien gibt, werden sie alle geteilt. Für jede Datenbankpartition wird jedoch nur eine einzige Teildatei generiert. Der Name der Teildatei entspricht dem Namen der ersten Eingabedatendatei.

**LOAD\_ONLY**

In diesem Modus werden zuvor geteilte Daten geladen. Die Daten sind in separaten Dateien enthalten, die nach der Konvention *dateiname.xxx* oder *dateiname.00xxx* benannt werden, wobei *xxx* die Nummer der Partition ist, zu der die Teildatei gehört. AutoLoader erwartet diese Dateien an der durch den Parameter *SPLIT\_FILE\_LOCATION* definierten Speicherposition oder im aktuellen Arbeitsverzeichnis. Für die Verzeichnisposition muß Lesezugriff bestehen. Die Teildateien werden gleichzeitig in die entsprechenden Partitionen geladen. Wenn beim Befehl LOAD mehrere Eingabedateien angegeben werden (wie *eingabedatei1*, *eingabedatei2* usw.), lädt AutoLoader *eingabedatei1.xxx*, sofern diese Datei vorhanden ist. Ansonsten wird *eingabedatei1.00xxx* geladen, sofern diese Datei vorhanden ist. Wenn keine dieser Dateien vorhanden ist, gibt AutoLoader einen Fehler zurück. Wenn beide Dateien vorhanden sind, lädt das Dienstprogramm AutoLoader *eingabedatei1.xxx*. Nach dem Laden der ersten *eingabeda-*

## AutoLoader-Optionen

teil, deren Dateityp xxx oder 00xxx ist, wird eingabedatei2 überprüft, und dieser Prozeß wird wiederholt, bis alle Eingabedateien geladen sind.

### ANALYZE

In diesem Modus wird eine angepaßte optimale Partitionierungszuordnung für eine Knotengruppe generiert. Es empfiehlt sich, eine Datendatei mit einer großen Anzahl von Datensätzen als Eingabedatei anzugeben (mehrere Eingabedateien sind zulässig). In diesem Fall erzeugt die Zuordnung eine gleichmäßigere Verteilung der Daten auf die einzelnen Datenbankpartitionen in der Knotengruppe. Die Ausgabe wird in die mit dem Parameter *MAP\_FILE\_OUTPUT* angegebene Datei geschrieben. Der Befehl *REDISTRIBUTE NODEGROUP* (siehe *Command Reference*) muß aufgerufen werden, bevor die neue Partitionierungszuordnung wirksam wird. Bei nachfolgenden Aufrufen von AutoLoader im Modus *SPLIT\_AND\_LOAD* wird automatisch die neue Partitionierungszuordnung verwendet. Der Parameter *MAP\_FILE\_INPUT* kann verwendet werden, wenn die Daten nach der neuen Partitionierungszuordnung partitioniert werden, ohne die Standardpartitionierung der Knotengruppe zu ändern.

### Parameter LOGFILE

Dieser Parameter wird verwendet, um den Basisdateinamen der temporären und permanenten Dateien anzugeben, die vom Dienstprogramm AutoLoader verwendet werden:

```
<protokolldatei>.split.cfg ...  
    Konfigurationsdatei für alle Verteilerprozesse.  
<protokolldatei>.split.<dreistellige-knotennummer>.log ...  
    Protokolldatei für jeden Verteilerprozeß.  
<protokolldatei>.pmap.<pid> ...  
    Interne temporäre Datei, wobei <pid> die Prozeß-ID  
    dieses AutoLoader-Jobs ist.  
<protokolldatei>.load.<dreistellige-knotennummer> ...  
    Nachrichtendatei für jeden Ladeprozeß, wenn keine  
    Nachrichtendatei im Befehl LOAD angegeben ist.
```

Sie können einen Pfad für den Parameter LOGFILE angeben, müssen dann jedoch sein Vorhandensein und die Zugriffsmöglichkeit prüfen. Der Standardwert ist `./autoloader.log`.

**Anmerkung:** Bei mehreren gleichzeitigen AutoLoader-Sitzungen müssen Sie sicherstellen, daß der angegebene Basisdateiname oder der Pfadname eindeutig ist.

**Parameter AUTHENTICATION und PASSWORD**

Diese Parameter sind notwendig, wenn für den fernen Aufruf des Verteilungsprogramms oder für Client/Server-Datenbankverbindungen ein Kennwort beim Laden erforderlich ist. Der Standardwert für AUTHENTICATION ist NO (keine Kennwortüberprüfung), und der für den Parameter PASSWORD angegebene Wert wird ignoriert.

Das Konzept einer lokalen Datenbankverbindung wurde für MPP-Umgebungen erweitert und beinhaltet nun auch Verbindungen von einem beliebigen Knoten eines gegebenen MPP-Exemplars. Das bedeutet, daß kein Kennwort erforderlich ist, wenn versucht wird, eine Verbindung von einem der in der Datei `db2nodes.cfg` definierten Knoten herzustellen, auch wenn für das Exemplar `AUTHENTICATION=server` angegeben ist. AutoLoader verwendet dieses neue Verbindungsverhalten, wenn die Markierung AUTHENTICATION in der AutoLoader-Konfigurationsdatei nicht oder mit NO definiert und kein Wert für den Parameter PASSWORD angegeben ist. Ein Kennwort für AutoLoader ist nur erforderlich, wenn ein Kennwort für die ferne Ausführung von Programmen auf Ihrem System benötigt wird. Beispiel: Ein Kennwort ist erforderlich, wenn die Datei `.rhosts` auf einem UNIX-System nicht richtig konfiguriert wurde, um die Ausführung von **rsh** zuzulassen.

Wenn ein Kennwort erforderlich ist, kann alternativ die DB2-Registrierungsvariable `DB2ATLD_PWFILE` festgelegt werden, die den vollständig qualifizierten Pfad zu einer vom Benutzer erstellten Kennwortdatei definiert. Das Dienstprogramm AutoLoader muß sowohl auf die Kennwortdatei als auch auf den vollständig qualifizierten Pfad zugreifen können. Wenn diese Variable definiert ist, wird das erste Wort in der Datei, auf die ihr Wert verweist, als Kennwort verwendet.

**Parameter MAX\_NUM\_SPLITTERS**

Dieser Parameter gibt die maximale Anzahl von Verteilerprozessen an, die in einem AutoLoader-Job verwendet werden können. Der Standardwert ist 25.

## AutoLoader-Optionen

### Parameter FORCE

Dieser Parameter erzwingt die Fortsetzung eines AutoLoader-Jobs auch dann, wenn das Dienstprogramm (beim Starten) feststellt, daß einige Zielpartitionen oder Tabellenbereiche offline sind. Wenn der Wert NO lautet und einige Partitionen nicht verfügbar sind, werden keine Daten verarbeitet. Wenn der Wert YES lautet, werden die verfügbaren Datenbankpartitionen geladen und alle anderen ignoriert. Der Standardwert für diesen Parameter ist NO.

### Parameter STATUS\_INTERVAL

Dieser Parameter gibt an, wie viele Megabyte (MB) Daten geladen werden sollen, bevor eine Statusnachricht generiert wird. Zulässige Werte sind ganze Zahlen im Bereich von 1 bis 4000. Der Standardwert ist 100.

### Parameter PORTS

Dieser Parameter gibt den Bereich von TCP-Anschlüssen an, die zur Erstellung von Sockets für die interne AutoLoader-Kommunikation verwendet werden. Der Standardbereich ist 6000 bis 6063. Wenn die DB2-Registrierungsvariable **DB2ATLD\_PORTS** beim Aufruf von AutoLoader definiert ist, überschreibt ihr Wert den für diesen Parameter angegebenen Wert.

### Parameter CHECK\_LEVEL

Dieser Parameter gibt an, ob eine Überprüfung auf unvollständige Datensätze bei der Ein- oder Ausgabe vorgenommen werden soll. Gültige Werte sind CHECK und NOCHECK. Der Standardwert ist NOCHECK.

### **Parameter MAP\_FILE\_INPUT**

Dieser Parameter gibt den Namen der Eingabedatei an, die auf eine Datei verweist, die die angepaßte Partitionierungszuordnung enthält. Wenn es sich bei der Partitionierungszuordnung um eine angepaßte Zuordnung, d. h. keine Standardzuordnung, handelt, muß dieser Parameter angegeben werden. Sie erhalten eine angepaßte Zuordnung, indem Sie AutoLoader im Modus ANALYZE aufrufen, um eine optimale Zuordnung zu generieren. Diese Zuordnung muß in jede Datenbankpartition versetzt werden, bevor die eigentliche Ladeoperation ausgeführt werden kann.

### **Parameter MAP\_FILE\_OUTPUT**

Dieser Parameter gibt einen Namen für die Partitionierungszuordnung an, wenn AutoLoader im Modus ANALYZE aufgerufen wird. Eine optimale Partitionierungszuordnung verteilt Daten gleichmäßig auf alle Datenbankpartitionen. Wenn für diesen Parameter kein Wert angegeben und das Dienstprogramm im Modus ANALYZE ausgeführt wird, wird ein Fehler zurückgegeben.

### **Parameter TRACE**

Dieser Parameter gibt die Anzahl der Datensätze an, deren Ablauf verfolgt werden soll, wenn Sie einen Speicherauszug des gesamten Datenumsetzungsprozesses und die Ausgabe der Hash-Werte prüfen müssen. Der Standardwert ist Null (keine Ablaufverfolgung).

### **Parameter NEWLINE**

Dieser Parameter gibt das Zeichen an, das verwendet wird, um die Datensätze in der Datendatei zu begrenzen. Dieser Parameter ist nur sinnvoll, wenn die Eingabedatendatei eine ASC-Datei mit fester Länge ist, in der jeder Datensatz durch ein Zeilenvorschubzeichen begrenzt wird, und wenn der Wert `recLen` im Befehl `LOAD` angegeben ist. Wenn der Wert `YES` angegeben ist, überprüft AutoLoader immer, ob der Datensatz durch ein Zeilenvorschubzeichen beendet ist. Es wird auch überprüft, ob die Satzlänge mit der durch den Wert `recLen` angegebenen Satzlänge übereinstimmt. Der Standardwert ist `N0`.

## AutoLoader-Beispielsitzung

---

### AutoLoader-Beispielsitzung

Das folgende Beispiel zeigt eine AutoLoader-Konfigurationsdatei (unter AIX):

```
#####
# release level
#####
RELEASE=V7.00

#####
# CLP load command
#####
db2 load from /home/user/atld_work/test.dat of del replace into user.test

#####
# database name
#####
database=wddb

#####
# split partition list
#####
SPLIT_NODES=(0,2)

#####
# running mode
#####
mode=split_and_load

#####
# log file token
#####
logfile=mylog

#####
# frequency of progressive information
#
# print out progressive info every 10
# mega-bytes of data
#####
STATUS_INTERVAL=10
```

Der folgende unter Verwendung dieser Konfigurationsdatei abgesetzte Befehl beinhaltet den Pfad und das temporäre Arbeitsverzeichnis, auf den bzw. das von jeder der teilnehmenden Datenbankpartitionen zugegriffen werden kann. Auch der Name der Konfigurationsdatei, `sample.atld.cfg`, wird angegeben:

```
/home/user/atld_work/ $ db2atld -config sample.atld.cfg
```

Der Aufruf dieses Befehls führt zu folgender Ausgabe:

```

/home/user/atld_work/ $ db2atld -config sample.atld.cfg
Utility program: "db2atld". Version: "07010".
Start reading autoloader configuration file: sample.atld.cfg.
Finish reading autoloader configuration file: sample.atld.cfg.
Start initializing autoloader process.
Finish initializing autoloader process.
The AutoLoader is now issuing all LOAD requests.
The LOAD operation has begun on partition "2".
The LOAD operation has begun on partition "0".
The LOAD operation has begun on partition "1".
The AutoLoader is now issuing all split requests.
Start db2split on node "0" in background.
Start db2split on node "2" in background.
The AutoLoader is waiting for all splitters to complete.
The utility has read "10" megabytes from the source data.
The utility has read "20" megabytes from the source data.
The utility has read "30" megabytes from the source data.
The utility has read "40" megabytes from the source data.
The utility has read "50" megabytes from the source data.
The utility has read "60" megabytes from the source data.
The utility has completed reading "62" megabytes from the user data.
The remote execution of the splitter utility on partition "0"
finished with remote execution code "0".
The AutoLoader is waiting for all LOAD operations to complete.
The remote execution of the splitter utility on partition "2"
finished with remote execution code "0".

```

Operation	Node	SQL Code	Result
LOAD	002	+00000000	Success.
LOAD	000	+00000000	Success.
LOAD	001	+00000000	Success.
SPLIT	000	+00000000	Success.
SPLIT	002	+00000000	Success.
PRE-SPLIT	000	+00000000	Success.
RESULTS:	3 of 3 LOADs completed successfully.		

Summary of Splitters:

```

Rows Read      500003
Rows Rejected  125214
Rows Partitioned 374789

```

Summary of LOADs:

```

Rows Read      374789
Rows Skipped   0

```

## AutoLoader-Beispielsitzung

```
Rows Loaded      374789
Rows Rejected    0
Rows Deleted     0
Rows Committed   374789
```

Der Hauptteil der durch AutoLoader generierten Nachrichten bezieht sich auf die Initialisierung der teilnehmenden Datenbankpartitionen. Der Fortschritt sowohl des Teilungs- als auch des Ladeprozesses wird in einer separaten Textdatei aufgezeichnet. Auch die Beendigung der AutoLoader-Prozesse wird aufgezeichnet.

Es wird auch eine Übersichtstabelle mit den ausgeführten Operationen, verwendeten Partitionen, zurückgegebenen SQL-Codes und erzielten Ergebnissen generiert. Wenn ein anderer SQL-Code als Null zurückgegeben wird, können die aufgezeichneten Warnungen oder Fehler in der Nachrichtendatei überprüft werden.

Eine Satzübersicht für den AutoLoader-Job vervollständigt die Ausgabe.

---

### Migration und Kompatibilität zur früheren Version

Beim Dienstprogramm AutoLoader müssen bestimmte Punkte hinsichtlich der Migration und Kompatibilität zur früheren Version berücksichtigt werden:

- Eine frühere Version des Dienstprogramms AutoLoader wurde mit dem Befehl **db2autold** aufgerufen. Die aktuelle Version wird mit dem Befehl **db2atld** aufgerufen.
- **db2atld** verwendet Sockets als interne Kommunikationskanäle (im Gegensatz zu benannten Pipes) und wählt eine TCP-Anschlußnummer aus dem Standardbereich von 6063 abwärts bis zu 6000. Wenn dieser Bereich bei Ihrem System jedoch für andere Anwendungen benötigt wird, haben Sie bei der Migration zwei Möglichkeiten:
  - Mit dem Parameter **PORTS** in der AutoLoader-Konfigurationsdatei kann ein anderer als der Standardanschlußbereich angegeben werden.
  - Die DB2-Registrierungsvariable **DB2ATLD\_PORTS** kann definiert werden, indem der Bereich wie folgt angegeben wird:

```
<niedrige-ananschlußnummer>:<hohe-ananschlußnummer>
```

Zur Ermittlung des TCP-Anschlußbereichs werden folgende Quellen in der genannten Reihenfolge herangezogen: DB2-Registrierungsvariable **DB2ATLD\_PORTS**, AutoLoader-Konfigurationsparameter **PORTS** und Standardanschlußbereich.



- Wenn für Datenbankverbindungen zwischen Client und Server ein Kennwort benötigt wird, haben Sie bei der Migration zwei Möglichkeiten:
  - Die AutoLoader-Konfigurationsparameter `AUTHENTICATION` und `PASSWORD` können verwendet werden. Wenn `AUTHENTICATION` auf `YES` gesetzt und `PASSWORD` definiert ist, wird das Kennwort für die Authentifizierung verwendet. Wenn `AUTHENTICATION` auf `YES` gesetzt und `PASSWORD` nicht definiert ist, werden Sie zur Eingabe eines Kennworts aufgefordert.
  - Der DB2-Registrierungswert `DB2ATLD_PWFILE` kann festgelegt werden, um eine Datei anzugeben, in der das Kennwort gespeichert ist. Wenn dieser Wert festgelegt wurde, wird der Inhalt der Datei ausgewertet, und die erste durch Leerzeichen begrenzte Zeichenfolge wird als Kennwort verwendet. Da der Registrierungswert als letzter ausgewertet wird, überschreibt er andere Kennwortwerte, sofern er definiert ist.

---

### Hinweise und Tips für das Dienstprogramm AutoLoader

Bevor Sie das Dienstprogramm AutoLoader verwenden, sollten Sie folgende Punkte beachten:

- Machen Sie sich mit AutoLoader-Operationen vertraut, indem Sie das Dienstprogramm mit kleinen Datenmengen verwenden.
- Wenn die Eingabedaten bereits sortiert sind oder in einer bestimmten Reihenfolge vorliegen und Sie diese Reihenfolge während des Ladevorgangs beibehalten wollen, darf nur eine einzige Datenbankpartition zum Teilen verwendet werden. Durch paralleles Teilen kann nicht sichergestellt werden, daß die Daten in derselben Reihenfolge geladen werden, in der sie empfangen wurden. Das Dienstprogramm AutoLoader wählt standardmäßig einen einzigen Verteilerprozeß aus, wenn der Wert `anyorder` im Befehl `LOAD` nicht angegeben wird.
- Wenn große Objekte (LOBs) aus separaten Dateien geladen werden (wenn Sie also den Wert `lobsinfile` über das Dienstprogramm `LOAD` verwenden), müssen alle Datenbankpartitionen, auf denen Ladevorgänge stattfinden, Lesezugriff auf alle Verzeichnisse mit LOB-Dateien haben. Der `LOAD`-Parameter `lob-pfad` muß beim Arbeiten mit LOBs vollständig qualifiziert sein.
- Alle temporären AutoLoader-Dateien befinden sich in dem Verzeichnis, das mit dem AutoLoader-Konfigurationsparameter `LOGFILE` angegeben wird. Auf dieses Verzeichnis muß im Netzwerk zugegriffen werden können. Zum Teilen ist Lese- und Schreibzugriff für alle Partitionen erforderlich. Durch Angabe von anderen Verzeichnissen für temporäre Dateien können Sie mehrere AutoLoader-Jobs gleichzeitig ausführen, um Daten in separate Tabellen in unterschiedlichen Tabellenbereichen zu laden.

## Hinweise und Tips für das Dienstprogramm AutoLoader

- Die maximale Anzahl von aktiven Datenbankverbindungen in einem AutoLoader-Job entspricht der Anzahl von Ladepartitionen, die im AutoLoader-Konfigurationsparameter `OUTPUT_NODES` definiert ist. Stellen Sie sicher, daß der Konfigurationsparameter *maxappls* hoch genug eingestellt ist.
- Sie können die Fortsetzung eines AutoLoader-Jobs erzwingen, selbst wenn AutoLoader (beim Starten) feststellt, daß einige Ladepartitionen oder zugehörige Tabellenbereiche offline sind, indem Sie `FORCE=YES` in der AutoLoader-Konfigurationsdatei angeben.
- Verwenden Sie den AutoLoader-Konfigurationsparameter `STATUS_INTERVAL`, um den Fortschritt eines AutoLoader-Jobs zu überwachen. AutoLoader gibt in den angegebenen Intervallen Nachrichten zurück, aus denen ersichtlich ist, wie viele Megabyte Daten verarbeitet wurden.
- Sie können mit besserer Leistung rechnen, wenn sich die Teilungspartitionen (durch den Parameter `SPLIT_NODES` definiert) von den Ladepartitionen (durch den Parameter `OUTPUT_NODES` definiert) unterscheiden, da es weniger Konkurrenzsituationen für CPU-Zyklen gibt. Das Dienstprogramm AutoLoader selbst sollte in einer Datenbankpartition aufgerufen werden, die nicht an der Teilungs- oder Ladeoperation teilnimmt. Bei einem SMP-System können Sie die Leistung verbessern, indem Sie sicherstellen, daß mindestens eine Verteilungs-Task für jede CPU verfügbar ist.
- AutoLoader ignoriert den Parameter `MESSAGES` des Befehls `LOAD` und leitet alle Nachrichten des Befehls `LOAD` in die Datei `load_log.XXX`; diese Datei enthält Nachrichten des Ladeprozesses auf der Datenbankpartition `XXX`. AutoLoader erstellt auch eine Datei namens `split_log.XXX`; diese Datei enthält Nachrichten des Teilungsprozesses auf Datenbankpartition `XXX`. Das Dienstprogramm erstellt außerdem eine Datei namens `autoload.log`, die Nachrichten der AutoLoader-Hauptprozedur enthält. Überprüfen Sie diese Datei, um sicherzustellen, daß alle Pipes und temporären Verzeichnisse richtig definiert wurden.
- AutoLoader wählt nur eine einzige Ausgabedatenbankpartition, auf der Statistikdaten gesammelt werden. Der AutoLoader Konfigurationsparameter `RUN_STAT_NODE` kann verwendet werden, um diese Partition anzugeben.
- AutoLoader kann mehrfach aufgerufen werden, um Daten gleichzeitig in separate Tabellen zu laden. Stellen Sie folgendes sicher:
  - Die Tabellen befinden sich in separaten Tabellenbereichen.
  - Alle AutoLoader-Operationen werden aus separaten Verzeichnissen aufgerufen.
  - Der Datendateiname, der zur Erstellungen temporärer Dateien verwendet wird, ist für jede AutoLoader-Operation eindeutig.

### Rahmenbedingungen und Einschränkungen

Für das Dienstprogramm AutoLoader gelten die folgenden Rahmenbedingungen:

- Die Eingabedateien für die Ladeoperation dürfen nicht auf einer Bandeinheit gespeichert sein.
- AutoLoader bietet keine Unterstützung für die Option CLIENT des Befehls LOAD.
- AutoLoader bietet keine Unterstützung für die Option ROWCOUNT des Befehls LOAD.
- Wenn Sie mehrere Datenbankpartitionen zur Partitionierung verwenden und anschließend die Daten laden, wird SAVECOUNT größer als Null im Befehl LOAD nicht unterstützt.
- Die Verwendung von AutoLoader zum Laden von Daten in Tabellen mit Spalten strukturierten Typs wird nicht unterstützt.

Informationen über Rahmenbedingungen für Ladevorgänge, die auch für AutoLoader gelten, finden Sie unter „Rahmenbedingungen und Einschränkungen“ auf Seite 180.

---

### Fehlerbehebung für AutoLoader

Wenn Sie den Eindruck haben, daß das Dienstprogramm AutoLoader blockiert, haben Sie folgende Möglichkeiten:

- Verwenden Sie den Parameter STATUS\_INTERVAL in der AutoLoader-Konfigurationsdatei, um den Fortschritt eines AutoLoader-Jobs zu überwachen.
- Überprüfen Sie die Dateien <protokolldatei>.split.<dreistellige-knotennummer>.log, um den Status der Verteilerprozesse in jeder Teildatenbankpartition abzurufen. Wenn alles funktioniert und der Parameter TRACE in der AutoLoader-Konfigurationsdatei definiert ist, sollten sich in diesen Protokolldateien Trace-Nachrichten für eine bestimmte Anzahl von Datensätzen befinden.
- Überprüfen Sie die LOAD-Nachrichtendatei oder die Dateien <protokolldatei>.load.<dreistellige-knotennummer> auf Ladefehlernachrichten.
- Wenn Fehlernachrichten vorhanden sind, die vermuten lassen, daß bei einem der AutoLoader-Prozesse Fehler aufgetreten sind, sollten Sie den aktuellen AutoLoader-Job unterbrechen.

## Fehlerbehebung für AutoLoader

Wenn beim Dienstprogramm AutoLoader weiterhin Fehler auftreten, haben Sie folgende Möglichkeiten:

1. Setzen Sie den Parameter MODE in der AutoLoader-Konfigurationsdatei auf SPLIT\_ONLY, und rufen Sie das Dienstprogramm erneut auf.
2. Überprüfen Sie die Dateien mit den geteilten Daten auf Unregelmäßigkeiten. Wenn die Teildateien in Ordnung zu sein scheinen, sollten Sie versuchen, eine dieser Dateien manuell in die richtige Datenbankpartition zu laden.
3. Wenn die Daten fehlerfrei geladen werden, liegen möglicherweise weitere AutoLoader-Probleme oder Probleme des Datenbanksystems vor. Wenden Sie sich an den IBM Kundendienst.

Bei AIX Version 4.2 kommt es unter Umständen zu einer Blockierung von **db2atld**, bis die folgenden Bibliotheken (mit der angegebenen oder einer höheren Stufe) vorhanden sind:

bos.rte.libc	4.2.1.13 (PTF U458582)
bos.rte.libpthreads	4.2.1.5 (PTF U458538)

Folgende Angaben beziehen sich auf ein Fehlerszenario mit dem Dienstprogramm AutoLoader bei Verwendung von IBM DB2 Universal Database Enterprise - Extended Edition für Windows NT.

Wenn **db2atld** auf einem Rechner mit mehreren Netzwerkkarten ausgeführt wird, müssen Sie sicherstellen, daß der Rechner korrekt konfiguriert ist. Geben Sie dazu den Befehl **hostname** auf dem Rechner ein, auf dem AutoLoader ausgeführt wird, und setzen Sie danach auf dem gleichen Rechner den Befehl **ping** für diesen Host-Namen ab. Die zurückgegebene IP-Adresse muß mit der IP-Adresse übereinstimmen, die beim Absetzen des Befehls **ping** für diesen Host-Namen von einem anderen Rechner in der DB2 MPP-Knotenliste aus zurückgegeben wird. Wenn der Rechner nicht richtig konfiguriert ist, gibt das Dienstprogramm den Fehler SQL6555N zurück, und Sie sehen die Fehlermeldung `errno = 10061 (Verbindung verweigert)` in den Dateien `db2diag.log` auf einigen der Ladeknoten, die durch den Parameter OUTPUT\_NODES in der AutoLoader-Konfigurationsdatei definiert sind.

Auf Windows NT-Rechnern wird die IP-Adresse, die für einen lokalen Host-Namen zurückgegeben wird, nicht vom Domain Name System oder aus der Datei `hosts` abgerufen, sondern aus Informationen, die lokal mit Hilfe des Symbols **Netzwerk** in der Systemsteuerung konfiguriert wurden. Ein Fehler von Windows NT Version 4.0 bewirkt, daß von der IP-Adressenfolge, die von einem Rechner mit mehreren Netzwerkkarten zurückgegeben wird, die Bindungsreihenfolge ignoriert wird, die mit Hilfe des Symbols **Netzwerk** in der Systemsteuerung konfiguriert wurde. Informationen zur Behebung dieses Problems finden Sie bei Microsoft Support im Online-Artikel Q171320.

---

## Kapitel 5. Versetzen von DB2 Data Links Manager-Daten

In diesem Kapitel wird beschrieben, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um DB2 Data Links Manager-Daten zu versetzen.

Informationen zu den Dateiformaten, die Sie mit diesen Dienstprogrammen verwenden können, finden Sie in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

Ausführliche Informationen über den DB2 Data Links Manager, finden Sie im Handbuch *DB2 Data Links Manager Einstieg*.

Die folgenden Themen werden behandelt:

- „Versetzen von DB2 Data Links Manager-Daten mit EXPORT“
- „Versetzen von DB2 Data Links Manager-Daten mit IMPORT“ auf Seite 206
- „Versetzen von DB2 Data Links Manager-Daten mit LOAD“ auf Seite 207.

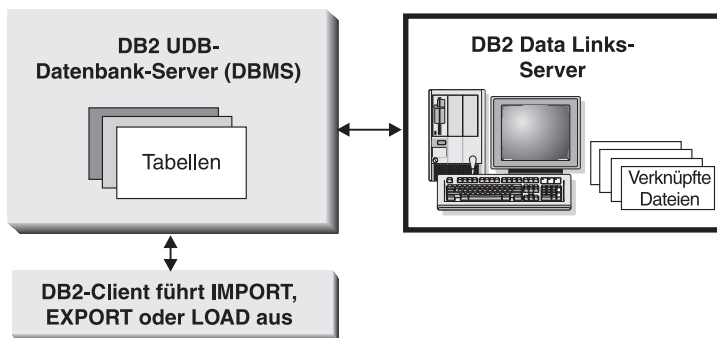


Abbildung 8. Versetzen von DB2 Data Links Manager-Daten. Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, müssen die Dienstprogramme EXPORT, IMPORT und LOAD sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen.

---

### Versetzen von DB2 Data Links Manager-Daten mit EXPORT

Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, muß das Dienstprogramm EXPORT sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen (siehe Abb. 8). Dazu erstellt das Dienstprogramm EXPORT eine Steuerdatei für jeden Data Links-Server. Der Name der erstellten Steuerdatei entspricht dem Namen des

## Versetzen von DB2 Data Links Manager-Daten mit EXPORT

Data Links-Servers. Die Steuerdateien werden in einem neu erstellten Verzeichnis namens `d1fm\JJJMMTT\SSMMSS` erstellt. Dabei steht *JJJMMTT* für *JahrMonatTag* und *SSMMSS* für *StundeMinuteSekunde*. Dieses Verzeichnis wird unter dem gleichen Verzeichnis erstellt, in dem auch die EXPORT-Datendatei erstellt wird. Eine Steuerdatei listet die Namen der entsprechenden DB2 Data Links Manager-Dateien auf, auf die die zu exportierenden DATALINK-Spaltenzeilen verweisen.

Beim verteilten Dateisystem unter AIX erstellt das Dienstprogramm EXPORT eine Steuerdatei für alle Data Links-Server innerhalb einer Zelle. Für jede Zelle gibt es eine Steuerdatei. Der Name der Steuerdatei entspricht dem Namen der Zelle. Die Steuerdatei listet die URL-Adressen aller DB2 Data Links Manager-Dateien innerhalb einer Zelle auf, auf die in den DATALINK-Spalten der exportierten Zeilen verwiesen wird.

Beim Betriebssystem Windows NT erstellt das Dienstprogramm EXPORT nur eine einzige Steuerdatei für alle Data Links-Server. Der Name dieser Steuerdatei lautet `ctrlfile.lst`. Sie wird in einem neu erstellten Verzeichnis namens `d1fm\JJJMMTT\SSMMSS` erstellt. Dieses Verzeichnis wird unter dem gleichen Verzeichnis erstellt, in dem auch die EXPORT-Datendatei erstellt wird. Die Steuerdatei listet die URL-Adressen aller DB2 Data Links Manager-Dateien auf, auf die in den DATALINK-Spalten der exportierten Zeilen verwiesen wird.

DATALINK-Werte mit dem Merkmal NO LINK CONTROL werden nicht in die Steuerdatei aufgenommen.

Die Steuerdateien müssen auf die jeweiligen Data Links-Server übertragen werden. Bei einem verteilten Dateisystem muß die Steuerdatei für jede Zelle auf einen der Data Links-Server innerhalb der Zelle übertragen werden. Beim Betriebssystem Windows NT muß die Steuerdatei auf alle verwendeten Data Links-Server übertragen werden. Das Dienstprogramm **dlfm\_export** muß unter Angabe des Steuerdateinamens auf jedem Data Links-Server ausgeführt werden. Dieses Dienstprogramm erstellt ein Archiv der Dateien, die in der Steuerdatei für den betreffenden Data Links-Server aufgelistet sind. Bei einem verteilten Dateisystem muß das Dienstprogramm **dlfm\_export** auf dem Data Links-Server ausgeführt werden, auf den die Steuerdatei übertragen wurde. Dieses Dienstprogramm erstellt ein Archiv der Dateien, die in der Steuerdatei für alle Data Links-Server innerhalb der Zelle aufgelistet sind. Legen Sie die Registrierdatenbankvariable **DLFM\_FS\_ENVIRONMENT** entsprechend fest, bevor Sie das Dienstprogramm **dlfm\_export** ausführen.

## Versetzen von DB2 Data Links Manager-Daten mit EXPORT

Führen Sie die folgende Schritte aus, um sicherzustellen, daß eine konsistente Kopie der Tabelle und der entsprechenden Dateien, auf die in den DATA-LINK-Spalten verwiesen wird, kopiert wird:

1. Setzen Sie den folgenden Befehl ab, um sicherzustellen, daß während der Ausführung der Exportoperation keine Aktualisierungstransaktionen ablaufen:

```
db2 quiesce tablespaces for table tabellenname share
```

2. Rufen Sie das Dienstprogramm EXPORT auf.
3. Führen Sie das Dienstprogramm **dlfm\_export** mit Root-Berechtigung auf jedem Data Links-Server aus. Damit können Dateien archiviert werden, auf die der Data Links-Dateimanageradministrator keinen Zugriff hat. Bei einem verteilten Dateisystem erhält das Dienstprogramm **dlfm\_export** die Root-Berechtigungen für das DCE-Netzwerk, bevor die in der Steuerdatei aufgelisteten Dateien archiviert werden. Die Dienstprogramm erfaßt nicht die ACL-Informationen der archivierten Dateien. Geben Sie als Eingabe für **dlfm\_export** den Namen der Steuerdatei an, die durch das Dienstprogramm EXPORT generiert wurde.
4. Setzen Sie den folgenden Befehl ab, um die Tabelle für Aktualisierungen verfügbar zu machen:

```
db2 quiesce tablespaces for table tabellenname reset
```

Das Dienstprogramm EXPORT wird als SQL-Anwendung ausgeführt. Die Zeilen und Spalten, die die Bedingungen der Anweisung SELECT erfüllen, werden aus der Datenbank extrahiert. Bei DATALINK-Spalten darf in der Anweisung SELECT keine Skalarfunktion angegeben werden.

Vom Dienstprogramm EXPORT werden die folgenden Dateien generiert:

- Die Exportdatendatei. Ein DATALINK-Spaltenwert in dieser Datei hat das gleiche Format wie das von den Dienstprogrammen LOAD und IMPORT verwendete. Wenn der DATALINK-Spaltenwert NULL ist, wird er auf die gleiche Art behandelt wie für andere NULL-Spalten.
- Steuerdateien für jeden Data Links-Server. Die Steuerdatei listet den vollständigen Pfad und die Namen aller Dateien auf, die von diesem Data Links-Server exportiert werden sollen. Bei einem verteilten Dateisystem gibt es für jede Zelle eine Steuerdatei. Beim Betriebssystem Windows NT gibt es nur eine einzige Steuerdatei für alle Data Links-Server, auf die von DATALINK-Spaltenwerten verwiesen wird.

Exportieren Sie die Dateien wie folgt mit dem Dienstprogramm **dlfm\_export** von einem oder mehreren Data Links-Servern:

```
dlfm_export steuerdateiname archivdateiname
```

## Versetzen von DB2 Data Links Manager-Daten mit EXPORT

Dabei ist *steuerdateiname* der Name der durch Ausführen des Dienstprogramms EXPORT auf dem DB2-Client generierten Datei, und *archivdateiname* ist der Name der Archivdatei, die generiert wird. Der Standardwert für *archivdateiname* ist `export.tar` im aktuellen Arbeitsverzeichnis.

Mit dem bereitgestellten Dienstprogramm **dlfm\_import** können Dateien aus dem von **dlfm\_export** generierten Archiv abgerufen und wiederhergestellt werden. Dieses Dienstprogramm muß verwendet werden, und zwar unabhängig davon, ob die Archivdateien auf dem gleichen oder auf einem anderen Data Links-Server wiederhergestellt werden.

Mit dem Dienstprogramm **dlfm\_import** können Sie wie folgt Dateien aus dem Archiv abrufen:

```
dlfm_import archivdateiname [LISTFILES]
```

Dabei ist *archivdateiname* der Name der Archivdatei, die zum Wiederherstellen der Dateien verwendet wird. Der Standardwert für *archivdateiname* ist `export.tar`. LISTFILES ist ein wahlfreies Schlüsselwort. Seine Angabe bewirkt, daß das Dienstprogramm eine Liste der im Archiv enthaltenen Dateien zurückgibt. Führen Sie das Dienstprogramm **dlfm\_import** mit Root-Berechtigung auf jedem Data Links-Server aus, weil Sie die Archivdateien möglicherweise auf einem anderen Data Links-Server wiederherstellen möchten, der über eine abweichende Verzeichnisstruktur und andere Benutzer-IDs verfügt als der Data Links-Server, auf dem das Dienstprogramm **dlfm\_export** ausgeführt wurde. Führen Sie bei einem verteilten Dateisystem das Dienstprogramm **dlfm\_import** auf einem der Data Links-Server innerhalb der Zielzelle aus. Dieses Dienstprogramm erhält die Root-Berechtigungen für das DCE-Netzwerk, bevor die Dateien aus dem Archiv extrahiert werden. Legen Sie die Registrierdatenbankvariable **DLFM\_FS\_ENVIRONMENT** entsprechend fest, bevor Sie das Dienstprogramm **dlfm\_import** ausführen.

**Anmerkung:** Wird das Dienstprogramm **dlfm\_import** nicht auf demselben Data Links-Server ausgeführt, auf dem das Dienstprogramm **dlfm\_export** ausgeführt wurde, werden die Dateien in den richtigen Pfaden wiederhergestellt. Der Eigner der Dateien ist Root, falls einige der Benutzer-IDs auf der Maschine, auf der die Dateien importiert werden, nicht vorhanden sind. Stellen Sie vor dem Einfügen dieser Dateien in eine Datenbank sicher, daß alle Dateien die richtigen Berechtigungen aufweisen und die richtigen Benutzer-IDs ihre Eigner sind. Erstellen Sie bei einem verteilten Dateisystem die erforderlichen Dateigruppen und definieren Sie die erforderlichen ACLs, bevor Sie das Dienstprogramm **dlfm\_import** ausführen. Dieses Dienstprogramm erstellt das Verzeichnis, das zum Extrahieren der Dateien benötigt wird, sofern es nicht vorhanden ist.



## Versetzen von DB2 Data Links Manager-Daten mit EXPORT

Die folgende Tabelle zeigt das Exportieren von DB2-Daten und der Dateien, auf die von der Datenbank namens SystemA in der Datenbank SystemB verwiesen wird. Sie gilt für Situationen ohne verteiltes Dateisystem. SystemA verwendet die Data Links-Server DLFM1 und DLFM2. SystemB verwendet die Data Links-Server DLFMX und DLFMY. Die Dateien auf DLFM1 werden nach DLFMX exportiert, und die Dateien auf DLFM2 werden nach DLFMY exportiert.

Datenbank SystemA mit Data Links-Servern DLFM1 und DLFM2			Schritt
DB2-Daten in Datei	Datei1 mit Dateinamen für DLFM1	Datei2 mit Dateinamen für DLFM2	1) Führen Sie den Befehl <b>dlfm_export</b> (mit Root-Berechtigung) auf beiden Data Links-Servern aus. Dadurch wird auf beiden Data Links-Servern ein Archiv erstellt.
Datenbank SystemB mit Data Links-Servern DLFMX und DLFMY			
	Bei DLFMX Wiederherstellen vom Archiv	Bei DLFMY Wiederherstellen vom Archiv	2) Führen Sie den Befehl <b>dlfm_import</b> (mit Root-Berechtigung) auf beiden Data Links-Servern aus.
			3) Führen Sie den Befehl <b>IMPORT</b> auf SystemB mit dem Parameter <b>DL_URL_REPLACE_PREFIX</b> aus, um den entsprechenden DataLinks-Server für jede exportierte Datei anzugeben.
Bei der Ausführung des Befehls <b>IMPORT</b> auf SystemB werden die Daten von SystemA und alle Dateien, auf die von DATALINK-Spalten verwiesen wird, importiert.			

### Versetzen von DB2 Data Links Manager-Daten mit IMPORT

Da sich die Tabellendaten in der Datenbank und die Dateien, auf die DATALINK-Spalten verweisen, auf Data Links-Servern befinden, muß das Dienstprogramm IMPORT sowohl die Datenbankdaten als auch die Datendateien auf den entsprechenden Data Links-Servern versetzen (siehe Abb. 8 auf Seite 201).

Vor der Ausführung des Dienstprogramms IMPORT für die Zieldatenbank sollten Sie die folgenden Schritte ausführen:

1. Kopieren Sie die Dateien, auf die verwiesen wird, auf die entsprechenden Data Links-Server. Mit dem Dienstprogramm **dlfm\_import** können Dateien aus einem Archiv extrahiert werden, das vom Dienstprogramm **dlfm\_export** erstellt wurde.
2. Definieren Sie auf den Data Links-Dateimanagern oder den Data Links-Servern den bzw. die Präfixnamen. (Sie müssen eventuell auch andere Verwaltungsaufgaben ausführen, wie das Registrieren der Datenbank.)
3. Aktualisieren Sie die Informationen zu den Data Links-Servern in den URL-Adressen (der DATALINK-Spalten) aus den exportierten Daten für die SQL-Tabelle (falls erforderlich). (Wenn die Data Links-Server der ursprünglichen Konfiguration mit den Servern an der Zielposition übereinstimmen, müssen die Namen der Data Links-Server nicht aktualisiert werden.) Aktualisieren Sie bei einem verteilten Dateisystem die Informationen über die Zellennamen in den URL-Adressen (der DATALINK-Spalten) aus den exportierten Daten für die SQL-Tabelle (falls erforderlich).
4. Definieren Sie die Data Links-Server in der Zielkonfiguration in der DB2 Data Links Manager-Konfigurationsdatei. Definieren Sie bei einem verteilten Dateisystem die Zellen in der Zielkonfiguration in der DB2 Data Links Manager-Konfigurationsdatei. Registrieren Sie die DB2 Data Links Manager innerhalb der Zielzelle mit dem Befehl **dlfm server\_conf** und dem Befehl **dlfm client\_conf**.

Wenn das Dienstprogramm IMPORT für die Zieldatenbank ausgeführt wird, werden Dateien, auf die in DATALINK-Spalten verwiesen wird, mit den entsprechenden Data Links-Servern verknüpft.

---

### Versetzen von DB2 Data Links Manager-Daten mit LOAD

Wenn Sie Daten in eine Tabelle mit einer DATALINK-Spalte laden, die mit FILE LINK CONTROL definiert ist, müssen Sie die folgenden Schritte ausführen, bevor Sie das Dienstprogramm LOAD aufrufen. (Wenn alle DATALINK-Spalten mit NO LINK CONTROL definiert sind, sind diese Schritte nicht erforderlich.)

1. Stellen Sie sicher, daß DB2 Data Links Manager auf den Data Links-Servern, auf die die DATALINK-Spaltenwerte verweisen, installiert ist. Stellen Sie bei einem verteilten Dateisystem sicher, daß die DB2 Data Links Manager innerhalb der Zielzelle registriert sind.
2. Stellen Sie sicher, daß die Datenbank im DB2 Data Links Manager eingetragen ist.
3. Kopieren Sie alle Dateien, die als DATALINK-Werte eingefügt werden, auf die entsprechenden Data Links-Server.
4. Definieren Sie für die DB2 Data Links Manager auf den Data Links-Servern den bzw. die Präfixnamen.
5. Registrieren Sie die Data Links-Server, auf die in den (zu ladenden) DATALINK-Daten verwiesen wird, in der DB2 Data Links Manager-Konfigurationsdatei. Registrieren Sie bei einem verteilten Dateisystem die Zellen in der Zielkonfiguration, auf die in den (zu ladenden) DATALINK-Daten verwiesen wird, in der DB2 Data Links Manager-Konfigurationsdatei.

Es kann vorkommen, daß die Verbindung zwischen DB2 und dem Data Links-Server während der Ausführung des Dienstprogramms LOAD fehlschlägt, wodurch die Ladeoperation fehlschlägt. Gehen Sie in diesem Fall wie folgt vor:

1. Starten Sie den Data Links-Server und den DB2 Data Links Manager.
2. Setzen Sie den Befehl LOAD RESTART ab (siehe „LOAD - Befehl“ auf Seite 101).

Während der Ladeoperation fehlgeschlagene Verbindungen werden als eine Art von Datenintegritätsverletzung eingestuft und wie Verletzungen eindeutiger Indizes gehandhabt. Es wurde daher eine besondere Ausnahmebedingung für das Laden von Tabellen definiert, die mindestens eine DATALINK-Spalte enthalten. Zusätzliche Informationen finden Sie in der Beschreibung von Ausnahmen im Handbuch *SQL Reference*.

## Versetzen von DB2 Data Links Manager-Daten mit LOAD

Die folgenden Funktionen des Dienstprogramms LOAD werden beim Laden von Tabellen mit DATALINK-Spalten nicht unterstützt:

- CPU\_PARALLELISM (der Wert wird auf 1 gesetzt)
- LOAD REPLACE
- LOAD TERMINATE
- LOAD COPY

---

## Kapitel 6. Versetzen von Daten zwischen Systemen

In diesem Kapitel wird beschrieben, wie die DB2-Dienstprogramme EXPORT, IMPORT und LOAD verwendet werden, um Daten zwischen verschiedenen Plattformen sowie in und aus DRDA-Host-Datenbanken zu übertragen. Auch DB2 DataPropagator, eine weitere Methode zum Versetzen von Daten innerhalb eines Unternehmens, wird beschrieben. Dieses Kapitel enthält außerdem eine Einführung in die Data Warehouse-Zentrale, mit deren Hilfe Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden können.

Die folgenden Themen werden behandelt:

- „Versetzen von Daten zwischen Plattformen“
- „Versetzen von Daten mit dem Tool db2move“ auf Seite 212
- „Versetzen von Daten mit DB2 Connect“ auf Seite 217
- „Versetzen von Daten zwischen typisierten Tabellen“ auf Seite 219
- „Verwenden der Replikation zum Versetzen von Daten“ auf Seite 225
- „Versetzen von Daten mit der Data Warehouse-Zentrale“ auf Seite 228.

---

### Versetzen von Daten zwischen Plattformen

Kompatibilität ist beim Exportieren, Importieren oder Laden von Daten zwischen verschiedenen Plattformen wichtig. In den folgenden Abschnitten werden Überlegungen zu den Dateiformaten PC/IXF, DEL (ASCII-Format ohne universelle Zeilenbegrenzer) und WSF beim Versetzen von Daten zwischen verschiedenen Betriebssystemen beschrieben. Weitere Informationen zu den Dateiformaten, die Sie für die DB2-Dienstprogramme zum Versetzen von Daten verwenden können, finden Sie in „Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD“ auf Seite 239.

#### PC/IXF-Dateiformat

PC/IXF ist das bevorzugte Dateiformat zum Versetzen von Daten zwischen Plattformen. PC/IXF-Dateien ermöglichen es dem Dienstprogramm LOAD oder IMPORT, numerische Daten, die normalerweise von der Maschine abhängig sind, maschinenunabhängig zu verarbeiten. Numerische Daten werden beispielsweise von der Intel-Architektur anders als von anderen Hardwarearchitekturen gespeichert und verwaltet.

Um die Kompatibilität der PC/IXF-Dateien für alle Produkte der DB2-Familie herzustellen, erstellt das Dienstprogramm EXPORT Dateien mit numerischen Daten im Intel-Format, während das Dienstprogramm IMPORT Daten dieses Formats erwartet.

## Versetzen von Daten zwischen Plattformen

Abhängig von der Hardwareplattform setzen DB2-Produkte numerische Werte zwischen Intel- und Nicht-Intel-Formaten (mittels Bytefolgeumkehrung) während der Export- und Importoperationen um.

Auf UNIX basierende Implementierungen von DB2 erstellen während des Exports keine mehrteiligen PC/IXF-Dateien. Der Import einer mehrteiligen PC/IXF-Datei ist dagegen zulässig, sofern die betreffende Datei von DB2 erstellt wurde. Wenn eine solche Datei importiert wird, müssen sich alle zugehörigen Teildateien im selben Verzeichnis befinden. Andernfalls meldet das Dienstprogramm einen Fehler.

Einteilige PC/IXF-Dateien, die auf UNIX basierenden Implementierungen des DB2-Dienstprogramms EXPORT erstellt wurden, können mit DB2 für OS/2 oder DB2 für Windows NT importiert werden.

### ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

DEL-Dateien weisen betriebssystemspezifische Unterschiede auf, je nachdem, unter welchem Betriebssystem sie erstellt wurden. Es gibt folgende Unterschiede:

- Zeilentrennzeichen
  - Auf UNIX basierende Textdateien verwenden ein Zeilenvorschubzeichen.
  - Nicht auf UNIX basierende Textdateien verwenden eine Folge aus Rücklauf- und Zeilenvorschubzeichen.
- Dateiendezeichen
  - Auf UNIX basierende Textdateien verwenden kein Dateiendezeichen.
  - Nicht auf UNIX basierende Textdateien verwenden ein Dateiendezeichen (X'1A').

Da DEL-Exportdateien Textdateien sind, können sie von einem Betriebssystem auf ein anderes übertragen werden. Dateiübertragungsprogramme können betriebssystemabhängige Dateien unterschiedlich verarbeiten, wenn Sie die Dateien im Textmodus übertragen. Zeilentrennzeichen und Dateiendezeichen werden im Binärmodus nicht umgesetzt.

**Anmerkung:** Wenn die Zeichendatenfelder Zeilentrennzeichen enthalten, werden diese ebenfalls während der Dateiübertragung umgesetzt. Diese Umsetzung führt zu unvorhersehbaren Änderungen der Daten. Aus diesem Grund sollten Sie keine DEL-Exportdateien verwenden, um Daten zwischen Plattformen zu versetzen. Verwenden Sie statt dessen das Dateiformat PC/IXF.

### WSF-Dateiformat

Numerische Daten in Dateien im WSF-Format werden im Intel-Maschinenformat gespeichert. Dieses Format ermöglicht es, Lotus-WSF-Dateien in andere Lotus-Betriebsumgebungen (z. B. auf Intel und auf UNIX basierende Systeme) zu übertragen und dort zu verwenden.

Als Folge dieser Konsistenz der internen Formate können WSF-Dateien, die aus DB2-Produkten exportiert wurden, von Lotus 1-2-3 oder Symphony auf anderen Plattformen verwendet werden. Außerdem können DB2-Produkte auch WSF-Dateien importieren, die auf anderen Plattformen erstellt wurden.

Verwenden Sie bei der Übertragung von WSF-Dateien zwischen Betriebssystemen stets den Binärmodus, nicht den Textmodus.

**Anmerkung:** Verwenden Sie das WSF-Dateiformat nicht zur Übertragung von Daten zwischen DB2-Datenbanken auf verschiedenen Plattformen, da dies zu Datenverlust führen kann. Verwenden Sie statt dessen das Dateiformat PC/IXF.

## Versetzen von Daten mit dem Tool db2move

### Versetzen von Daten mit dem Tool db2move

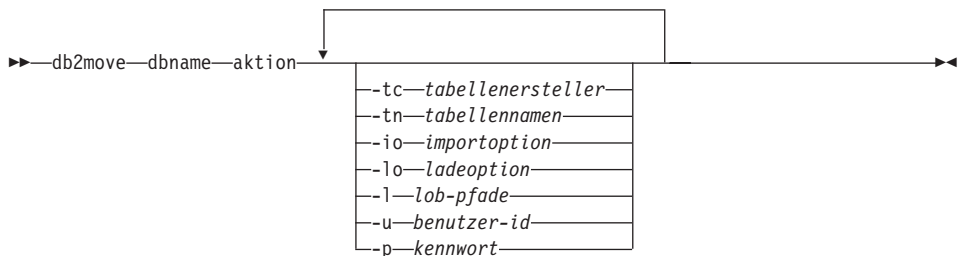
Dieses Tool kann Sie bei der Übertragung großer Mengen von Tabellen zwischen DB2-Datenbanken auf Workstations unterstützen. Das Tool fragt die Systemkatalogtabellen nach einer bestimmten Datenbank ab und stellt eine Liste aller Benutzertabellen zusammen. Anschließend exportiert es diese Tabellen im PC/IXF-Format. Die PC/IXF-Dateien können in andere lokale DB2-Datenbanken auf demselben System importiert oder geladen werden oder auf eine andere Workstation-Plattform übertragen und auf dieser Plattform in eine DB2-Datenbank importiert oder geladen werden.

**Anmerkung:** Tabellen mit Spalten strukturierten Typs werden bei Verwendung dieses Tools nicht versetzt.

### Berechtigung

Dieses Tool ruft abhängig von der durch den Benutzer angeforderten Aktion die DB2-API EXPORT, IMPORT oder LOAD auf. Daher muß die anfordernde Benutzer-ID über die richtigen Berechtigungen für die APIs verfügen, sonst schlägt die Anforderung fehl.

### Befehlsyntax



### Befehlsparameter

#### **dbname**

Der Name der Datenbank.

**aktion** Muß EXPORT, IMPORT oder LOAD sein.

**-tc** Tabellenersteller. Standardmäßig werden alle Ersteller ausgewählt.

Die Option gilt nur für die Aktion EXPORT. Wenn sie angegeben wird, werden nur die von den mit dieser Option angegebenen Erstellern erstellten Tabellen exportiert. Wird sie nicht angegeben, werden standardmäßig alle Ersteller verwendet. Bei Angabe mehrerer Ersteller muß jede Angabe durch ein Komma getrennt werden. Leerzeichen sind zwischen Ersteller-IDs nicht zulässig. Die maximale Anzahl von Erstellern, die angegeben werden kann, ist 10. Diese Option kann



zusammen mit der Option „-tn“ (Tabellennamen) zur Auswahl der zu exportierenden Tabellen verwendet werden.

Der Stern (\*) kann an einer beliebigen Stelle in der Zeichenfolge als Platzhalterzeichen angegeben werden.

- tn Tabellennamen. Standardmäßig werden alle Benutzertabellen ausgewählt.

Die Option gilt nur für die Aktion EXPORT. Wenn sie angegeben wird, werden nur die Tabellen exportiert, deren Namen exakt mit denen in der angegebenen Zeichenfolge übereinstimmen. Wird sie nicht angegeben, werden standardmäßig alle Benutzertabellen verwendet. Bei der Angabe mehrerer Tabellennamen muß jede Angabe durch ein Komma getrennt werden. Leerzeichen sind zwischen Tabellennamen nicht zulässig. Die maximale Anzahl von Tabellennamen, die angegeben werden kann, ist 10. Diese Option kann zusammen mit der Option „-tc“ (Tabellenersteller) zur Auswahl der zu exportierenden Tabellen verwendet werden. **db2move** exportiert nur die Tabellen, deren Namen mit den angegebenen Tabellennamen und deren Ersteller mit den angegebenen Erstellern übereinstimmen.

Der Stern (\*) kann an einer beliebigen Stelle in der Zeichenfolge als Platzhalterzeichen angegeben werden.

- io Importoption. Der Standardwert ist REPLACE\_CREATE.  
Gültige Werte sind INSERT, INSERT\_UPDATE, REPLACE, CREATE und REPLACE\_CREATE.
- lo Ladeoption. Der Standardwert ist INSERT.  
Gültige Optionen sind INSERT und REPLACE.
- l LOB-Pfade. Der Standardwert ist das aktuelle Verzeichnis.

Diese Option gibt die absoluten Pfadnamen an, in denen LOB-Dateien (bei EXPORT) erstellt bzw. (bei IMPORT oder LOAD) gesucht werden. Bei der Angabe mehrerer LOB-Pfade muß jede Angabe durch ein Komma getrennt werden. Leerzeichen sind zwischen LOB-Pfaden nicht zulässig. Wenn im ersten Pfad kein Speicherplatz mehr verfügbar ist (bei EXPORT) oder die Dateien nicht gefunden werden (bei IMPORT oder LOAD), wird der zweite Pfad verwendet usw.

Wenn als Aktion EXPORT angegeben wurde und LOB-Pfade angegeben werden, werden alle Dateien in den LOB-Pfadverzeichnissen gelöscht, die Verzeichnisse entfernt und neue Verzeichnisse erstellt. Wenn diese Option nicht angegeben wird, wird das aktuelle Verzeichnis als LOB-Pfad verwendet.

## Versetzen von Daten mit dem Tool db2move

- u** Benutzer-ID. Der Standardwert ist die angemeldete Benutzer-ID.

Sowohl die Benutzer-ID als auch das Kennwort sind wahlfrei. Wenn jedoch eines von beiden angegeben wird, müssen beide angegeben werden. Wenn der Befehl auf einem Client ausgeführt wird, der eine Verbindung zu einem fernen Server herstellt, sollten die Benutzer-ID und das Kennwort angegeben werden.
- p** Kennwort. Standardwert ist das bei der Anmeldung verwendete Kennwort.

Sowohl die Benutzer-ID als auch das Kennwort sind wahlfrei. Wenn jedoch eines von beiden angegeben wird, müssen beide angegeben werden. Wenn der Befehl auf einem Client ausgeführt wird, der eine Verbindung zu einem fernen Server herstellt, sollten die Benutzer-ID und das Kennwort angegeben werden.

### Beispiele

- `db2move sample export`

Mit diesem Befehl werden alle Tabellen in der Beispieldatenbank SAMPLE exportiert. Für alle Optionen werden die Standardwerte verwendet.
- `db2move sample export -tc userid1,us*rid2 -tn tname1,*tname2`

Mit diesem Befehl werden alle Tabellen exportiert, die von „userid1“ oder Benutzer-IDs wie „us%rid2“ und mit dem Namen „tname1“ oder Tabellennamen wie „%tname2“ erstellt wurden.
- `db2move sample import -l D:\LOBPATH1,C:\LOBPATH2`

Dieses Beispiel gilt nur für OS/2 oder die Windows-Betriebssysteme. Mit diesem Befehl werden alle Tabellen in SAMPLE importiert, zugehörige LOB-Dateien werden in den LOB-Pfaden „D:\LOBPATH1“ und „C:\LOBPATH2“ gesucht.
- `db2move sample load -l /home/userid/lobpath,/tmp`

Dieses Beispiel gilt nur für auf UNIX basierende Systeme. Mit diesem Befehl werden alle Tabellen in die Beispieldatenbank SAMPLE geladen, zugehörige LOB-Dateien werden im Unterverzeichnis /home/userid/lobpath und im Unterverzeichnis tmp gesucht.
- `db2move sample import -io replace -u userid -p password`

Mit diesem Befehl werden alle Tabellen in die Beispieldatenbank SAMPLE im Modus REPLACE importiert, und die angegebene Kombination aus Benutzer-ID und Kennwort wird verwendet.

### Anmerkungen zur Verwendung

Dieses Tool exportiert, importiert oder lädt von Benutzern erstellte Tabellen. Wenn die Datenbank von einem Betriebssystem in ein anderes Betriebssystem dupliziert werden soll, vereinfacht **db2move** das Versetzen der Tabellen. Sie müssen auch daran denken, alle anderen Objekte, die den Tabellen zugeordnet sind, zu versetzen: Aliasnamen, Sichten, Auslöser, benutzerdefinierte Funktionen usw. Das Programm **db2look** (DB2 Statistics and DLL Extraction Tool; siehe Handbuch *Command Reference*) kann Ihnen beim Versetzen einiger dieser Objekte helfen, indem es die DDL-Anweisungen (Data Definition Language - Datendefinitionssprache) aus der Datenbank extrahiert.

Wenn die APIs EXPORT, IMPORT oder LOAD von **db2move** aufgerufen werden, wird der Parameter `FileTypeMod` auf `lobsinfile` gesetzt. Das heißt, LOB-Daten werden in Dateien gespeichert, die von PC/IXF-Dateien getrennt sind. Es sind 26.000 Dateinamen für LOB-Dateien verfügbar.

Die Aktion LOAD muß lokal auf der Maschine ausgeführt werden, auf der sich die Datenbank und die Datendatei befinden. Wenn die API LOAD von **db2move** aufgerufen wird, wird der Parameter `CopyTargetList` auf den Wert NULL gesetzt, d. h. es erfolgt kein Kopiervorgang. Wenn der Parameter *logretain* aktiviert ist, kann die Ladeoperation später nicht aktualisierend wiederhergestellt werden. Der Tabellenbereich, in dem die geladene Tabelle gespeichert ist, wird in den Status „Sicherung anstehend“ versetzt und ist für den Zugriff nicht verfügbar. Es ist eine vollständige Datenbanksicherung oder eine Tabellenbereichssicherung erforderlich, um den Status „Sicherung anstehend“ für den Tabellenbereich aufzuheben.

Wird dieses Tool auf einem Client der Version 5.2 für eine Datenbank der Version 6 verwendet, werden Tabellen- oder Spaltennamen, die länger als 18 Zeichen sind, nicht unterstützt.

#### Bei Verwendung von EXPORT erforderliche/generierte Dateien:

- Eingabe: Keine.
- Ausgabe:

**EXPORT.out** Das zusammengefaßte Ergebnis der Aktion EXPORT.

**db2move.lst** Die Liste der Originaltabellenamen, ihrer entsprechenden PC/IXF-Dateinamen (`tabnnn.ixf`) sowie der Nachrichtendateinamen (`tabnnn.msg`). Diese Liste, die exportierten PC/IXF-Dateien und die LOB-Dateien (`tabnnnc.yyy`) werden als Eingabe für die Aktion IMPORT oder LOAD von **db2move** verwendet.

## Versetzen von Daten mit dem Tool db2move

<b>tabnnn.ixf</b>	Die exportierte PC/IXF-Datei einer bestimmten Tabelle.
<b>tabnnn.msg</b>	Die exportierte Nachrichtendatei der entsprechenden Tabelle.
<b>tabnnnc.yyy</b>	Die exportierten LOB-Dateien einer bestimmten Tabelle. „nnn“ ist die Tabellenummer. „c“ ist ein Buchstabe des Alphabets. „yyy“ ist eine Ziffer aus dem Bereich von 001 bis 999.  Diese Dateien werden nur erstellt, sofern die Tabelle, die exportiert wird, LOB-Daten enthält. Wenn sie erstellt werden, werden diese LOB-Dateien in den LOB-Pfadverzeichnissen („lobpath“) gespeichert. Es sind 26.000 Dateinamen für LOB-Dateien verfügbar.
<b>system.msg</b>	Die Nachrichtendatei mit den Systemnachrichten zu den Befehlen für das Erstellen oder Löschen von Dateien oder Verzeichnissen. Diese Datei wird nur verwendet, sofern die Aktion EXPORT ist und ein LOB-Pfad angegeben wird.

### Bei Verwendung von IMPORT erforderliche/generierte Dateien:

- Eingabe:
  - db2move.lst** Eine Ausgabedatei der Aktion EXPORT.
  - tabnnn.ixf** Eine Ausgabedatei der Aktion EXPORT.
  - tabnnnc.yyy** Eine Ausgabedatei der Aktion EXPORT.
- Ausgabe:
  - IMPORT.out** Das zusammengefaßte Ergebnis der Aktion IMPORT.
  - tabnnn.msg** Die Datei mit IMPORT-Nachrichten für die entsprechende Tabelle.

### Bei Verwendung von LOAD erforderliche/generierte Dateien:

- Eingabe:
  - db2move.lst** Eine Ausgabedatei der Aktion EXPORT.
  - tabnnn.ixf** Eine Ausgabedatei der Aktion EXPORT.
  - tabnnnc.yyy** Eine Ausgabedatei der Aktion EXPORT.
- Ausgabe:
  - LOAD.out** Das zusammengefaßte Ergebnis der Aktion LOAD.
  - tabnnn.msg** Die Datei mit LOAD-Nachrichten für die entsprechende Tabelle.

## Versetzen von Daten mit DB2 Connect

Wenn Sie in einer komplexen Umgebung arbeiten, in der Sie Daten zwischen einem Host-Datenbanksystem und einer Workstation versetzen müssen, können Sie DB2 Connect, den Gateway für die Datenübertragung vom Host zur Workstation und umgekehrt, verwenden (siehe Abb. 9).

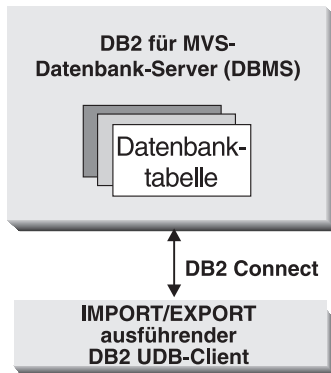


Abbildung 9. Import/Export über DB2 Connect

Im folgenden Abschnitt wird der Export und der Import von Daten mit DB2 Connect erörtert.

### Verwenden der Dienstprogramme EXPORT und IMPORT

Mit den DB2-Dienstprogrammen EXPORT und IMPORT können Daten von einer DRDA-Server-Datenbank in eine Datei auf der DB2 Connect-Workstation oder umgekehrt versetzt werden. Anschließend können diese Daten mit jeder anderen Anwendung oder jedem anderen Verwaltungssystem für relationale Datenbanken verwendet werden, von der/dem das betreffende Import-/Exportformat unterstützt wird. Beispielsweise können Sie Daten aus DB2 für MVS/ESA in eine ASCII-Datei mit begrenzter Satzlänge exportieren und sie später in eine Datenbank von DB2 für OS/2 importieren.

Sie können Export- und Importfunktionen von einem Datenbank-Client oder von der DB2 Connect-Workstation aus ausführen.

#### Anmerkungen:

1. Die zu importierenden oder exportierenden Daten dürfen die Beschränkungen der beiden Datenbanken hinsichtlich der Größe und Datentypen nicht verletzen.

## Versetzen von Daten mit DB2 Connect

2. Zur Verbesserung der Leistung beim Importieren können zusammengesetzte SQL-Anweisungen verwendet werden. Geben Sie beim Dienstprogramm IMPORT den Dateitypwert `compound` an, um eine angegebene Anzahl von SQL-Anweisungen zu einem Block zusammenzufassen (siehe „Dateitypwerte (IMPORT)“ auf Seite 63). Hierdurch können der Systemaufwand für das Netzwerk verringert und die Antwortzeiten verbessert werden.
3. Weitere Informationen zur Syntax der Dienstprogramme EXPORT und IMPORT finden Sie in „EXPORT - Befehl“ auf Seite 9 und „IMPORT - Befehl“ auf Seite 41.

### Versetzen von Daten von einer Workstation auf einen DRDA-Server

Gehen Sie wie folgt vor, um Daten in eine DRDA-Server-Datenbank zu versetzen:

1. Exportieren Sie die Daten aus einer DB2-Tabelle in eine PC/IXF-Datei.
2. Importieren Sie die PC/IXF-Datei mit der Option `INSERT` in eine kompatible Tabelle in der DRDA-Server-Datenbank.

### Versetzen von Daten von einem DRDA-Server auf eine Workstation

Gehen Sie wie folgt vor, um Daten von einer DRDA-Server-Datenbank zu versetzen:

1. Exportieren Sie die Daten aus der Tabelle der DRDA-Server-Datenbank in eine PC/IXF-Datei.
2. Importieren Sie die PC/IXF-Datei in eine DB2-Tabelle.

### Einschränkungen

Bei DB2 Connect müssen Export- und Importoperationen die folgenden Bedingungen erfüllen:

- Der Dateityp muß PC/IXF sein.
- Es muß eine Tabelle vorhanden sein, deren Attribute mit denen der Daten kompatibel sind, bevor der Importvorgang gestartet werden kann. Beim Import durch DB2 Connect kann keine Tabelle erstellt werden, weil `INSERT` die einzige unterstützte Option ist.
- Beim Import darf kein Intervall für die Festschreibezählung angegeben werden.

Wenn eine dieser Bedingungen nicht erfüllt ist, schlägt die Operation fehl, und eine Fehlermeldung wird zurückgegeben.

**Anmerkung:** Indexdefinitionen werden weder beim Export gespeichert noch beim Import verwendet.

### Gemischte SBCS- und DBCS-Daten

Beim Importieren und Exportieren gemischter Daten (d. h. Spalten, die sowohl SBCS- als auch DBCS-Daten enthalten) muß folgendes beachtet werden:

- Auf Systemen, die Daten im EBCDIC-Format speichern (MVS, OS/390, OS/400, VM und VSE), markieren DBCS-Startzeichen und DBCS-Endezeichen den Anfang und das Ende der DBCS-Daten. Bei der Definition von Spaltenlängen für die Datenbanktabellen muß daher genügend Platz für diese Zeichen gelassen werden.
- Wenn die Daten in einer Spalte kein einheitliches Muster haben, werden Spalten für Zeichendaten variabler Länge empfohlen.

---

### Versetzen von Daten zwischen typisierten Tabellen

Mit den DB2-Dienstprogrammen EXPORT und IMPORT können Daten in typisierte Tabellen bzw. aus typisierten Tabellen heraus versetzt werden. Typisierte Tabellen können sich in einer Hierarchie befinden. Das Versetzen von Daten zwischen Hierarchien kann folgendes beinhalten:

- Versetzen von Daten aus einer Hierarchie in eine identische Hierarchie
- Versetzen von Daten aus einer Hierarchie in einen Teilbereich einer größeren Hierarchie
- Versetzen von Daten aus einem Teilbereich einer großen Hierarchie in eine separate Hierarchie

Die Option IMPORT CREATE ermöglicht es, sowohl die Tabellen- als auch die Typhierarchie zu erstellen.

Die Kennzeichnung von Typen innerhalb einer Hierarchie ist datenbankabhängig. Dies bedeutet, daß derselbe Typ in verschiedenen Datenbanken unterschiedliche Kennungen hat. Darum muß beim Versetzen von Daten zwischen diesen Datenbanken ein Typenabgleich erfolgen, um sicherzustellen, daß die Daten korrekt versetzt werden.

Bevor die einzelnen typisierten Zeilen bei einer Exportoperation herausgeschrieben werden, wird eine Kennung in einen Indexwert umgesetzt. Dieser Indexwert kann eine beliebige Zahl von 1 bis zur Anzahl der relevanten Typen in der Hierarchie sein. Indexwerte werden gebildet, indem beim Durchgang durch die Hierarchie in einer bestimmten Reihenfolge alle Typen durchnumeriert werden. Die Reihenfolge wird als „Traversierfolge“ bezeichnet.

## Versetzen von Daten zwischen typisierten Tabellen

Dabei erfolgt der Durchgang durch alle übergeordneten und untergeordneten Tabellen in der Hierarchie von oben nach unten und von links nach rechts. Die Traversierfolge ist wichtig beim Versetzen von Daten zwischen Tabellenhierarchien, weil dabei im Verhältnis zu anderen Daten ermittelt wird, wohin die Daten versetzt werden.

Eine Methode ist der Durchgang von der obersten Stufe der Hierarchie (oder der Stammtabelle) durch die untergeordneten Stufen (untergeordnete Tabellen) bis zur letzten untergeordneten Tabelle; anschließend zurück zu ihrer übergeordneten Tabelle, weiter durch die nächste(n) „unmittelbar rechts angrenzende(n)“ untergeordnete(n) Tabelle(n); danach zurück zur nächsthöheren übergeordneten Tabelle, weiter durch deren untergeordnete Tabellen usw.

Die folgende Abbildung zeigt eine Hierarchie mit vier Traversierfolgen:

- Person, Employee, Manager, Architect, Student
- Person, Student, Employee, Manager, Architect (diese Traversierfolge ist mit einer gestrichelten Linie gekennzeichnet)
- Person, Employee, Architect, Manager, Student
- Person, Student, Employee, Architect, Manager

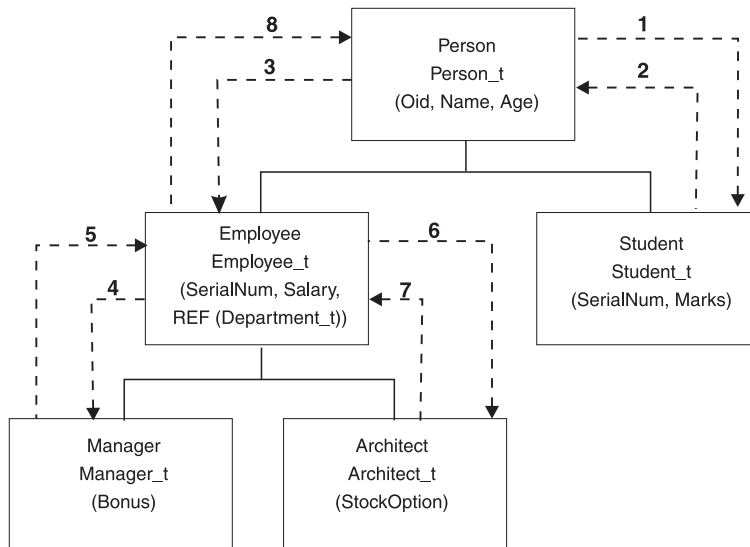


Abbildung 10.



### Traversierfolge

Es gibt eine Standardtraversierfolge, in der alle relevanten Typen ab einem bestimmten Startpunkt in der Hierarchie auf alle erreichbaren Typen der Hierarchie verweisen. Die Standardreihenfolge schließt alle Tabellen in der Hierarchie ein, und jede Tabelle wird nach dem Schema sortiert, das im erweiterten Sortiervergleichselement verwendet wird. Außerdem gibt es eine benutzerdefinierte Traversierfolge, bei der die zu verwendenden relevanten Typen vom Benutzer in einer Traversierfolgeliste definiert werden. Beim Arbeiten mit den Dienstprogrammen EXPORT und IMPORT muß dieselbe Traversierfolge verwendet werden.

Beachten Sie beim Angeben der Traversierfolge, daß der Durchgang durch die untergeordneten Tabellen der vorgegebenen Anordnung (PRE-ORDER) folgen muß (d. h. ein neuer Zweig der hierarchischen Struktur wird erst angefangen, nachdem der vorangegangene Zweig bis zum Ende abgearbeitet ist).

#### Standardtraversierfolge

Die Standardtraversierfolge verhält sich bei verschiedenen Dateiformaten unterschiedlich. In den nachfolgend dargestellten Fällen wird vorausgesetzt, daß identische Tabellenhierarchien und Typenbeziehungen vorliegen.

Beim Export von Daten in das PC/IXF-Dateiformat wird ein Datensatz mit allen relevanten Typen, ihren Definitionen und den relevanten Tabellen erzeugt. EXPORT nimmt außerdem die Zuordnung von Indexwerten für die einzelnen Tabellen vor. Beim Import wird anhand dieser Zuordnung sichergestellt, daß die Daten korrekt in die Zieldatenbank versetzt werden. Beim Arbeiten mit dem Dateiformat PC/IXF ist es ratsam, die Standardtraversierfolge zu verwenden.

Beim Arbeiten mit den Dateiformaten ASC, DEL und WSF kann die Erstellungsreihenfolge der typisierten Zeilen und Tabellen unterschiedlich sein, obwohl die Quellen- und Zielhierarchien dieselbe Struktur aufweisen. Dies würde zu Zeitdifferenzen führen, die die Standardtraversierfolge beim Durchgang durch die Hierarchien feststellen würde. Die Erstellungszeit jedes Typs bestimmt, in welcher Reihenfolge der Durchgang durch die Quellen- und die Zielhierarchie erfolgt, wenn die Standardtraversierfolge verwendet wird. Sie müssen sicherstellen, daß sowohl die Reihenfolge der Erstellung der einzelnen Typen in Quellen- und Zielhierarchie als auch die hierarchische Struktur von Quelle und Ziel identisch sind. Wählen Sie eine benutzerdefinierte Standardtraversierfolge aus, wenn diese Bedingungen nicht erfüllt werden können.

## Versetzen von Daten zwischen typisierten Tabellen

### **Benutzerdefinierte Traversierfolge**

Wenn Sie die Traversierfolge für die Hierarchien steuern möchten, müssen Sie sicherstellen, daß für die Dienstprogramme EXPORT und IMPORT die gleiche Traversierfolge verwendet wird. Angenommen, die folgenden Bedingungen sind erfüllt:

- Eine identische Definition von untergeordneten Tabellen in den Quellen- und Zieldatenbanken
- Eine identische hierarchische Beziehung zwischen untergeordneten Tabellen in den Quellen- und Zieldatenbanken
- Eine identische Traversierfolge

In diesem Fall ist die exakte Versetzung von Daten in die Zieldatenbank durch das Dienstprogramm IMPORT sichergestellt.

Beim Definieren der Traversierfolge legen Sie zwar den Startpunkt und den Pfad durch die Hierarchie fest, aber jede Verzweigung der Hierarchie muß zunächst komplett verarbeitet werden, bevor die nächste Verzweigung in Angriff genommen werden kann. Die Dienstprogramme EXPORT und IMPORT erkennen die Nichteinhaltung dieser Bedingung innerhalb der angegebenen Traversierfolge und melden gegebenenfalls einen Fehler.

### **Datenauswahl beim Versetzen von Daten**

Das Versetzen von Daten aus einer hierarchischen Struktur mit typisierten Tabellen in eine andere erfolgt entsprechend einer bestimmten Traversierfolge und durch Erstellung einer temporären Flachdatei. Über das Dienstprogramm EXPORT (in Kombination mit der Traversierfolge) kann gesteuert werden, welche Daten aus der Quelldatenbank in die temporäre Datei übernommen werden. Sie brauchen nur den Namen der Zieltabelle zusammen mit der WHERE-Klausel anzugeben. Das Dienstprogramm EXPORT verwendet diese Auswahlkriterien, um eine entsprechende temporäre Datei zu erstellen.

Das Dienstprogramm IMPORT steuert, was in die Zieltabelle gestellt wird. Sie können, falls gewünscht, am Ende jedes Namens einer untergeordneten Tabelle eine Attributliste angeben, um festzulegen, daß nur bestimmte Attribute in die Zieldatenbank versetzt werden sollen. Wenn keine Attributliste angegeben ist, werden alle Spalten der jeweiligen untergeordneten Tabelle versetzt.

Das Dienstprogramm IMPORT steuert Größe und Position der zu versetzenden Hierarchie mit den Parametern CREATE, INTO tabellenname, UNDER und AS ROOT TABLE. Weitere Informationen zu den Parametern des Befehls IMPORT finden Sie in „IMPORT - Befehl“ auf Seite 41.

## Beispiele zum Versetzen von Daten zwischen typisierten Tabellen

Die Beispiele in diesem Abschnitt basieren auf der folgenden hierarchischen Struktur:

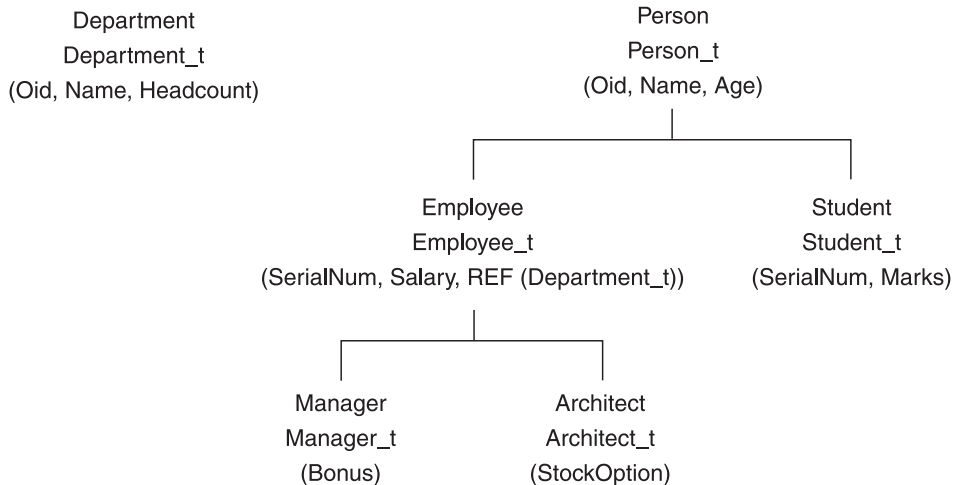


Abbildung 11.

### Beispiel 1

Importieren einer vollständigen Hierarchie und erneutes Erstellen dieser Hierarchie durch eine Importoperation:

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.ixf OF IXF HIERARCHY STARTING Person
DB2 CONNECT TO Target_db
DB2 IMPORT FROM entire_hierarchy.ixf OF IXF CREATE INTO
HIERARCHY STARTING Person AS ROOT TABLE
```

Alle in der Hierarchie enthaltenen Typen werden erstellt, wenn sie noch nicht vorhanden sind. Wenn diese Typen bereits vorhanden sind, müssen Sie in der Zieldatenbank die gleichen Definitionen aufweisen wie in der Quelldatenbank. Wenn die Definitionen nicht identisch sind, wird ein SQL-Fehler (SQL20013N) zurückgegeben. Da eine neue Hierarchie erstellt wird, gilt außerdem, daß keine der untergeordneten Tabellen, die in der in die Zieldatenbank (Target\_db) zu versetzenden Datendatei definiert sind, vorhanden sein darf. Alle angegebenen Tabellen aus der Hierarchie der Quelldatenbank werden erstellt. Zuletzt werden die Daten aus der Quelldatenbank in die richtigen untergeordneten Tabellen der Zieldatenbank importiert.

## Versetzen von Daten zwischen typisierten Tabellen

### Beispiel 2

In einem etwas komplexeren Beispiel soll die gesamte Hierarchie der Quelldatenbank exportiert und anschließend in die Zieldatenbank importiert werden. Außerdem werden zwar alle Daten zu Personen, die über 20 Jahre alt sind, aus der Quelldatenbank exportiert, aber es werden nur bestimmte ausgewählte Daten in die Zieldatenbank importiert.

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO entire_hierarchy.del OF DEL HIERARCHY (Person,
Employee, Manager, Architect, Student) WHERE Age>=20
DB2 CONNECT TO Target_db
DB2 IMPORT FROM entire_hierarchy.del OF DEL INSERT INTO (Person,
Employee(Salary), Architect) IN HIERARCHY (Person, Employee,
Manager, Architect, Student)
```

Die Zieltabellen Person, Employee und Architect müssen vorhanden sein. Daten werden in die untergeordneten Tabellen Person, Employee und Architect importiert. Es wird also folgendes importiert:

- Alle Spalten aus der Tabelle Person in die Tabelle Person
- Alle Spalten aus Person und Salary der Tabelle Employee in die Tabelle Employee
- Alle Spalten aus Person und Salary der Tabelle Employee sowie alle Spalten der Tabelle Architect in die Tabelle Architect

Die Spalten SerialNum und REF(Employee\_t) werden weder in die Tabelle Employee noch in die dazugehörigen untergeordneten Tabellen importiert (hier ist Architect die einzige untergeordnete Tabelle, in die Daten importiert werden).

**Anmerkung:** Da Architect eine untergeordnete Tabelle von Employee ist und als einzige Importspalte für Employee die Spalte Salary angegeben ist, wird Salary als einzige spezifische Spalte der Tabelle Employee in die Tabelle Architect importiert. Dies bedeutet, daß weder die Spalte SerialNum noch die Spalte REF(Employee\_t) in die Zeilen Employee oder Architect importiert werden.

Es werden keine Daten in die Tabellen Manager und Student importiert.

### Beispiel 3

In diesem Beispiel sollen Daten aus einer regulären Tabelle als eine einzige untergeordnete Tabelle in eine Hierarchie importiert werden. Der Befehl EXPORT verarbeitet eine reguläre (nicht typisierte) Tabelle, d. h. die Daten-datei enthält keine Spalte Type\_id. Der Wert no\_type\_id gibt diese Information an das Dienstprogramm IMPORT weiter, damit es nicht als erste Spalte die Spalte Type\_id erwartet.

```
DB2 CONNECT TO Source_db
DB2 EXPORT TO Student_sub_table.del OF DEL SELECT * FROM
Regular Student
DB2 CONNECT TO Target_db
DB2 IMPORT FROM Student_sub_table.del OF DEL METHOD P(1,2,3,5,4)
MODIFIED BY NO_TYPE_ID INSERT INTO HIERARCHY (Student)
```

In diesem Beispiel muß die Zieltabelle Student vorhanden sein. Da es sich bei Student um eine untergeordnete Tabelle handelt, wird durch den Wert no\_type\_id angegeben, daß die erste Spalte keine Type\_id enthält. Sie müssen jedoch sicherstellen, daß die Tabelle Student zusätzlich zu allen anderen Attributen der Tabelle eine Spalte Object\_id enthält. Außerdem wird Object\_id als erste Spalte in jeder Zeile erwartet, die in die Tabelle Student importiert wird. Die Klausel METHOD kehrt die Reihenfolge der beiden letzten Attribute um.

---

## Verwenden der Replikation zum Versetzen von Daten

Die Replikation ermöglicht Ihnen, Daten regelmäßig auf mehrere ferne Datenbanken zu kopieren. Wenn Sie Aktualisierungen einer Hauptdatenbank automatisch auf andere Datenbanken kopieren müssen, können Sie die DB2-Replikationseinrichtungen verwenden, um anzugeben, welche Daten kopiert werden sollen, in welche Datenbanktabellen die Daten zu kopieren sind und wie oft die Aktualisierungen kopiert werden sollen. Die DB2-Replikationseinrichtungen sind Teil einer größeren IBM Lösung für die Replikation von Daten in kleinen und großen Unternehmen.

Die IBM Replikations-Tools sind eine Gruppe von DB2 DataPropagator-Programmen und DB2 Universal Database-Tools, die Daten zwischen Verwaltungssystemen für verteilte relationale Datenbanken kopieren:

- Zwischen DB2 Universal Database-Plattformen
- Zwischen DB2 Universal Database-Plattformen und Host-Datenbanken, die die DRDA-Konnektivität (Distributed Relational Database Architecture - Architektur einer verteilten relationalen Datenbank) unterstützen
- Zwischen Host-Datenbanken, die die DRDA-Konnektivität unterstützen

Mit Hilfe von DB2 DataJoiner können Daten auch auf Verwaltungssysteme für relationale Datenbanken von anderen Herstellern als IBM repliziert werden.

## Verwenden von Replikation zum Versetzen von Daten

Sie können die IBM Replikations-Tools verwenden, um Kopieroperationen von einem einzigen Steuerpunkt aus für Daten im gesamten Unternehmen zu definieren, zu synchronisieren, zu automatisieren und zu verwalten. Die Replikations-Tools in DB2 Universal Database ermöglichen eine Replikation zwischen relationalen Datenbanken. Außerdem arbeiten sie mit IMS DataPropagator (vormals DPropNR) zusammen, um IMS- und VSAM-Daten zu replizieren, sowie mit Lotus NotesPump, um Replikationen an und von Lotus Notes-Datenbanken vorzunehmen.

Die Replikation versetzt Sie in die Lage, Endbenutzern und Anwendungen den Zugriff auf Produktionsdaten zu ermöglichen, ohne die Produktionsdatenbank weiter zu belasten. Sie können die Daten auf eine Datenbank kopieren, die für einen Benutzer oder eine Anwendung lokal ist, so daß sie nicht fern auf die Daten zugreifen müssen. Ein typisches Replikationsszenario wäre eine Quellentabelle mit Kopien in mindestens einer fernen Datenbank, wie es zum Beispiel bei der Zentrale einer Bank und ihren Zweigstellen vorstellbar ist. Zu vordefinierten Zeiten erfolgen automatische Aktualisierungen der DB2-Datenbanken, und alle Änderungen an der Quellendatenbank werden in die Zieldatenbanktabellen kopiert.

Die Replikations-Tools ermöglichen Ihnen, die Struktur der Kopiertabelle anzupassen. Sie können beim Kopieren in die Zieldatenbank SQL verwenden, um die zu kopierenden Daten auszuwählen, zu bearbeiten oder auf andere Weise aufzubereiten. Sie können Kopien für den Lesezugriff erstellen, die die Quellentabelle duplizieren, und haben die Möglichkeit, Daten zu einem bestimmten Zeitpunkt zu erfassen, ein Protokoll der Änderungen bereitzustellen oder Daten, die zu weiteren Zieltabellen kopiert werden sollen, zwischenspeichern. Darüber hinaus können Sie Kopien für den Schreib-/Lesezugriff erstellen, die von Endbenutzern oder Anwendungen aktualisiert werden können und deren Änderungen zurück in die Haupttabelle repliziert werden. Sie können Sichten von Quellentabellen und Sichten von Kopien replizieren. Ereignisgesteuerte Replikation ist ebenfalls möglich.

Die Replikation von Daten zwischen DB2-Datenbanken ist auf den folgenden Plattformen möglich: AIX, AS/400, HP-UX, Linux, Microsoft Windows 95, 98, 2000 und NT, OS/2, OS/390, SCO UnixWare, Sun Solaris, Sequent, VM und VSE. Außerdem können (mit DB2 DataJoiner) Daten zwischen DB2 und den folgenden anderen Datenbankprodukten repliziert werden: Informix, Microsoft Jet, Microsoft SQL Server, Oracle, Sybase und Sybase SQLAnywhere. In Kombination mit anderen IBM Produkten können Sie DB2-Daten auf bzw. von IMS, VSAM oder Lotus Notes replizieren. Schließlich können Sie Daten ferner an DB2 Everywhere auf Einheiten mit dem Windows CE- oder Palm-Betriebssystem replizieren.

### Die IBM Replikations-Tools nach Komponenten

Die IBM Replikations-Tool-Lösung setzt sich aus zwei Komponenten zusammen: dem Capture-Programm und dem Apply-Programm. Sie können diese Komponenten mit der DB2-Steuerzentrale einrichten. Der Betrieb und die Überwachung dieser Komponenten erfolgt außerhalb der Steuerzentrale.

Das Capture-Programm erfaßt Änderungen, die an den Quellentabellen vorgenommen wurden. Eine Quellentabelle kann folgendes sein:

- Eine externe Tabelle, die SQL-Daten aus einem Dateisystem oder einem nichtrelationalen Datenbankmanager enthält, die außerhalb von DB2 Data-Propagator geladen werden.
- Eine vorhandene Tabelle in der Datenbank.
- Eine Tabelle, die zuvor vom Apply-Programm aktualisiert wurde, das das Kopieren von Änderungen zurück in die Quellentabelle oder in andere Zieltabellen ermöglicht.

Die Änderungen werden in eine CD-Tabelle kopiert, in der sie gespeichert werden, bis das Zielsystem bereit ist, sie zu kopieren. Anschließend ruft das Programm DPROPR Apply die Änderungen aus der CD-Tabelle ab und kopiert sie in die Zieltabellen.

Verwenden Sie die Steuerzentrale für folgende Zwecke:

- Einrichten der Replikationsumgebung
- Definieren von Quellen- und Zieltabellen
- Festlegen des zeitlichen Ablaufs der automatisierten Kopiervorgänge
- Angeben von SQL-Erweiterungen für die Daten
- Definieren von Beziehungen zwischen den Quellen- und Zieltabellen

Weitere Informationen finden Sie im *Replikation Benutzer- und Referenzhandbuch*.

### Versetzen von Daten mit der Data Warehouse-Zentrale

Mit Hilfe der Data Warehouse-Zentrale können Daten aus Betriebsdatenbanken in eine Warehouse-Datenbank versetzt werden, die von Benutzern als Entscheidungshilfe abgefragt werden kann. Außerdem können Sie die Data Warehouse-Zentrale verwenden, um die Struktur der Betriebsdatenbanken, den sogenannten *Quellen* zu definieren. Anschließend können Sie angeben, wie die Betriebsdaten versetzt und für das Warehouse umgesetzt werden sollen. Sie können die Struktur der Tabellen in der Warehouse-Datenbank (die sogenannten *Ziele*) modellieren oder die Tabellen automatisch im Rahmen des Definitionsprozesses für die Operationen der Datenversetzung erstellen.

Die Data Warehouse-Zentrale setzt die folgenden DB2-Funktionen zum Versetzen und Umsetzen von Daten ein:

- **SQL**

Mit Hilfe von SQL können Sie Daten aus Quellen auswählen und die Daten in Ziele einfügen. Außerdem können Sie SQL zum Umsetzen der Daten in das entsprechende Warehouse-Format verwenden. SQL können Sie mit der Data Warehouse-Zentrale generieren. Alternativ können Sie auch Ihr eigenes SQL schreiben.

- **Dienstprogramme LOAD und EXPORT**

Mit diesen DB2-Dienstprogrammen können Sie Daten aus einer Quelle exportieren und die Daten anschließend in ein Ziel laden. Diese Dienstprogramme sind nützlich, wenn Sie große Datenmengen versetzen müssen. Die Data Warehouse-Zentrale unterstützt die folgenden Arten von Lade- und Exportoperationen:

**DB2 Daten exportieren**

Exportiert Daten aus einer lokalen DB2-Datenbank in eine begrenzte Datei.

**Daten mit ODBC exportieren**

Wählt Daten aus einer Tabelle in einer Datenbank, die in ODBC registriert ist, aus und schreibt die Daten anschließend in eine begrenzte Datei.

**DB2 Laden**

Lädt Daten aus einer begrenzten Datei in eine DB2-Tabelle.

**DB2 Laden in DB2 UDB EEE-Datenbank (nur AIX)**

Lädt Daten aus einer begrenzten Datei in eine Datenbank von DB2 Universal Database Enterprise - Extended Edition und ersetzt hierbei in einer Tabelle vorhandene Daten durch neue Daten. Diese Operation fordert die Zielpartitionierungszuordnung für die Datenbank an, teilt die Eingabedatei so, daß jede Datei auf eine Datenbankpartition geladen werden kann, und führt anschließend für alle Partitionen eine ferne Ladeoperation durch.



## Verwenden von Replikation zum Versetzen von Daten

- Replikation

Zum Kopieren großer Datenmengen von Warehouse-Quellen in ein Warehouse-Ziel können Sie auch die Replikation einsetzen und anschließend alle nachfolgenden Änderungen der Quelldaten erfassen. Die Data Warehouse-Zentrale unterstützt die folgenden Replikationsarten:

**Basisergebnistabelle**

Erstellt eine Zieltabelle, die Ergebnisdaten für eine Benutzertabelle enthält und in bestimmten Intervallen angehängt wird.

**CA-Tabelle**

Erstellt eine Zieltabelle, die Ergebnisdaten enthält und auf den Änderungen basiert, die für eine Quellentabelle aufgezeichnet werden.

**Zeitpunkt**

Erstellt eine Zieltabelle, die mit der Quellentabelle identisch ist, und fügt der Zieltabelle eine Zeitmarkenspalte hinzu.

**Zwischenspeichertabelle**

Erstellt eine CDD-Tabelle, die als Quelle für aktualisierte Daten, die für mehrere Zieltabellen bestimmt sind, verwendet werden kann.

**Benutzerkopie**

Erstellt eine Zieltabelle, die mit der Quellentabelle zum Zeitpunkt der Kopiererstellung identisch ist.

Diese Operationen werden in jeder Workstation-Betriebsumgebung von DB2 Universal Database - DB2 Universal Database für OS/390, DB2 für AS/400 und DataJoiner - unterstützt.

- Umsetzungsprogramm für gespeicherte Prozeduren

Mit der Data Warehouse-Zentrale können Sie Daten in eine OLAP-Datenbank (Online Analytical Processing - Online-Analyseverarbeitung) versetzen. Nachdem die Daten in das Warehouse versetzt wurden, können Sie das Umsetzungsprogramm für gespeicherte Prozeduren verwenden, um die Daten zu bereinigen und anschließend in Fakt- und Dimensionstabellen zusammenfassen. Außerdem können Sie mit den Umsetzungsprogrammen statistische Daten generieren. Nachdem die Daten bereinigt und umgesetzt wurden, können Sie sie in OLAP-Kuben laden oder auf Abteilungs-Server replizieren, die manchmal auch als *Datamarts* bezeichnet werden. Die Umsetzungsprogramme sind nicht in allen DB2-Angeboten enthalten. Weitere Informationen erhalten Sie bei Ihrem IBM Ansprechpartner.

Weitere Informationen über die Data Warehouse-Zentrale finden Sie im Handbuch *Data Warehouse-Zentrale Verwaltung*.

## Verwenden von Replikation zum Versetzen von Daten

---

## Anhang A. Lesen von Syntaxdiagrammen

Ein Syntaxdiagramm zeigt, wie ein Befehl angegeben werden muß, damit das Betriebssystem die Eingabe richtig auswerten kann.

Lesen Sie ein Syntaxdiagramm von links nach rechts und von oben nach unten. Folgen Sie dabei der horizontalen Linie (dem Hauptpfad). Wenn eine Linie mit einer Pfeilspitze endet, wird die Befehlssyntax fortgesetzt, und die nächste Linie beginnt mit einer Pfeilspitze. Ein vertikaler Balken kennzeichnet das Ende der Befehlssyntax.

Bei der Eingabe von Daten eines Syntaxdiagramms müssen Sie auch Zeichen wie Anführungszeichen und Gleichheitszeichen eingeben.

Parameter sind als Schlüsselwörter oder Variablen klassifiziert:

- Schlüsselwörter sind Konstanten und werden in Großbuchstaben dargestellt. An der Eingabeaufforderung können sie jedoch in beliebiger Schreibweise eingegeben werden. Ein Befehlsname ist ein Beispiel für ein Schlüsselwort.
- Variablen sind Namen oder Werte, die vom Benutzer angegeben werden, und werden in Kleinbuchstaben dargestellt. An der Eingabeaufforderung können sie jedoch in beliebiger Schreibweise eingegeben werden, sofern dies nicht ausdrücklich anders angegeben ist. Ein Dateiname ist ein Beispiel für eine Variable.

Ein Parameter kann eine Kombination aus einem Schlüsselwort und einer Variablen sein.

Erforderliche Parameter werden im Hauptpfad angegeben:



Wahlfreie Parameter werden unterhalb des Hauptpfads angegeben:

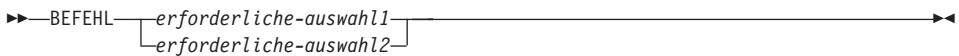


## Lesen von Syntaxdiagrammen

Der Standardwert eines Parameters wird oberhalb des Pfad angegeben:



Ein Stapel von Parametern (erster Parameter im Hauptpfad angegeben) gibt an, daß einer der Parameter gewählt werden muß:



Ein Stapel von Parametern, bei dem der erste Parameter unterhalb des Hauptpfads angegeben ist, gibt an, daß einer der Parameter gewählt werden kann:

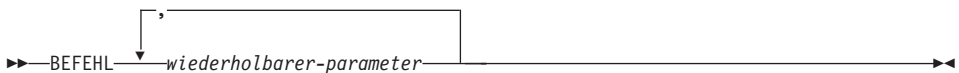


Ein nach links zurückweisender Pfeil oberhalb des Pfads gibt an, daß Elemente gemäß den folgenden Konventionen wiederholt werden können:

- Wenn der Pfeil durchgängig ist, kann das Element in einer Liste wiederholt werden, wobei die Elemente durch Leerstellen zu trennen sind:

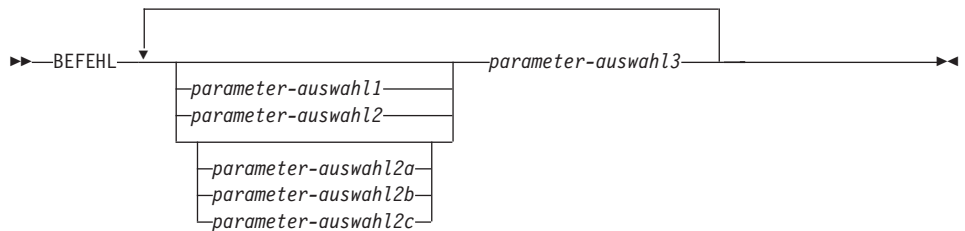


- Wenn der Pfeil ein Komma enthält, kann das Element in einer Liste wiederholt werden, wobei die Elemente durch Kommas zu trennen sind:

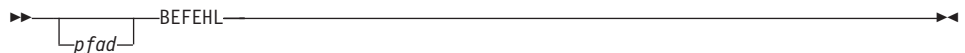


Elemente aus Parameterstapeln können gemäß den Stapelkonventionen für die zuvor erläuterten erforderlichen und wahlfreien Parameter wiederholt werden.

Manche Syntaxdiagramme enthalten Parameterstapel innerhalb von anderen Parameterstapeln. Elemente aus Stapeln können nur gemäß den zuvor erläuterten Konventionen wiederholt werden. Das heißt, wenn über einem inneren Stapel kein nach links zurückweisender Pfeil ist, über einem äußeren Stapel jedoch schon, kann nur ein einziger Parameter aus dem inneren Stapel gewählt und mit einem beliebigen Parameter des äußeren Stapels kombiniert werden; diese Kombination kann dann wiederholt werden. Das folgende Diagramm zeigt beispielsweise, daß der Parameter *parameter-auswahl2a* mit dem Parameter *parameter-auswahl2* kombiniert und diese Kombination dann wiederholt werden kann (*parameter-auswahl2* plus *parameter-auswahl2a*):



Einigen Befehlen ist ein wahlfreier Pfadparameter vorangestellt:



Wenn dieser Parameter nicht angegeben ist, sucht das System den Befehl im aktuellen Verzeichnis. Wenn der Befehl nicht gefunden wird, sucht das System weiterhin in allen Verzeichnissen in den Pfaden, die in der Datei `.profile` bzw. in der Anweisung `PATH` angegeben sind.

Einige Befehle weisen syntaktische Varianten auf, die funktionell identisch sind:



## Lesen von Syntaxdiagrammen

## Anhang B. Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD

Die folgende Tabelle faßt wichtige Unterschiede zwischen den DB2-Dienstprogrammen LOAD und IMPORT zusammen.

Dienstprogramm IMPORT	Dienstprogramm LOAD
Langsam beim Versetzen von großen Datenmengen.	Schneller als das Dienstprogramm IMPORT beim Versetzen großer Datenmengen, da das Dienstprogramm LOAD formatierte Seiten direkt in die Datenbank schreibt.
Partitionsinterne Parallelität wird beschränkt ausgenutzt.	Partitionsinterne Parallelität wird ausgenutzt. Dazu sind in der Regel symmetrische Mehrprozessormaschinen (SMP-Maschinen) erforderlich.
FASTPARSE wird nicht unterstützt.	FASTPARSE wird unterstützt und bietet eine reduzierte Datenprüfung der vom Benutzer bereitgestellten Daten.
CODEPAGE wird nicht unterstützt.	CODEPAGE wird unterstützt, und Zeichendaten (sowie numerische Daten, die in Zeichen angegeben sind) werden während der Ladeoperation von dieser Codepage in die Codepage der Datenbank umgesetzt.
Hierarchische Daten werden unterstützt.	Hierarchische Daten werden nicht unterstützt.
Das Erstellen von Tabellen, Hierarchien und Indizes wird beim PC/IXF-Format unterstützt.	Tabellen und Indizes müssen vorhanden sein.
Das Importieren von Daten in Übersichtstabellen wird nicht unterstützt.	Das Laden von Daten in Übersichtstabellen wird unterstützt.
WSF-Format wird unterstützt.	WSF-Format wird nicht unterstützt.
BINARYNUMERICS wird nicht unterstützt.	BINARYNUMERICS wird unterstützt.
PACKEDDECIMAL wird nicht unterstützt.	PACKEDDECIMAL wird unterstützt.
ZONEDDECIMAL wird nicht unterstützt.	ZONEDDECIMAL wird unterstützt.

## Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD

Dienstprogramm IMPORT	Dienstprogramm LOAD
Das Überschreiben von Spalten, die als GENERATED ALWAYS definiert sind, ist nicht möglich.	Das Überschreiben von Spalten, die als GENERATED ALWAYS definiert sind, ist durch Verwendung der Dateitypwerte GENERATEDIGNORE und IDENTITYIGNORE möglich.
Das Importieren von Daten in Tabellen und Sichten wird unterstützt.	Daten können nur in Tabellen geladen werden.
Die Tabellenbereiche, in denen sich die Tabelle und die Indizes befinden, sind während der Dauer der Importoperation online.	Die Tabellenbereiche, in denen sich die Tabelle und die Indizes befinden, sind während der Dauer der Ladeoperation offline.
Alle Zeilen werden protokolliert.	Es erfolgt nur eine minimale Protokollierung.
Auslöser werden unterstützt.	Auslöser werden nicht unterstützt.
Wenn die Importoperation unterbrochen wird und ein Wert für <i>commitcount</i> angegeben wurde, ist die Tabelle verwendbar und enthält die Zeilen, die bis zur letzten COMMIT-Operation geladen wurden. Der Benutzer kann dann die Importoperation erneut starten oder die Tabelle so akzeptieren, wie sie vorliegt.	Wenn die Ladeoperation unterbrochen wird und ein Wert für <i>savecount</i> angegeben wurde, verbleibt die Tabelle im Status „Laden anstehend“ und kann nicht verwendet werden, bis die Ladeoperation erneut gestartet und fortgesetzt wird oder der Tabellenbereich anhand eines vor der Ladeoperation erstellten Sicherungsabbilds wiederhergestellt wird.
Der erforderliche Speicherbereich entspricht in etwa dem größten Index plus 10%. Dieser Speicherbereich wird den temporären Tabellenbereichen in der Datenbank entnommen.	Der erforderliche Speicherbereich entspricht in etwa der Summe der Größen aller Indizes, die für die Tabelle definiert sind, und kann u. U. bis doppelt so groß werden. Dieser Speicherbereich wird dem temporären Speicherplatz in der Datenbank entnommen.
Alle Integritätsbedingungen werden während der Importoperation ausgewertet.	Während der Ladeoperation erfolgt eine Prüfung auf Eindeutigkeit, aber alle anderen Integritätsbedingungen müssen mit der Anweisung SET INTEGRITY überprüft werden.
Die Schlüsselwerte werden während einer Importoperation einzeln in den Index eingefügt.	Nach dem Laden der Daten werden die Schlüsselwerte sortiert, und der Index wird erstellt.
Wenn aktualisierte Statistikdaten erforderlich sind, muß das Dienstprogramm RUNSTATS nach einer Importoperation ausgeführt werden.	Statistikdaten können während der Ladeoperation gesammelt werden, wenn die Daten in der Tabelle ersetzt (REPLACE) werden.



## Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD

Dienstprogramm IMPORT	Dienstprogramm LOAD
Über DB2 Connect können Sie Daten in eine Host-Datenbank importieren.	Es ist nicht möglich, Daten in eine Host-Datenbank zu laden.
Importdateien müssen sich auf dem Knoten befinden, auf dem das Dienstprogramm IMPORT aufgerufen wird.	In einer Umgebung mit partitionierten Datenbanken müssen sich die zu ladenden Dateien oder Pipes auf dem Knoten befinden, der die Datenbank enthält. In einer Umgebung ohne partitionierte Datenbanken können sich die zu ladenden Dateien oder Pipes auf dem Knoten befinden, der die Datenbank enthält, oder auf dem Client mit Fernverbindung, von dem aus das Dienstprogramm LOAD aufgerufen wurde.
Ein Sicherungsabbild ist nicht erforderlich. Da vom Dienstprogramm IMPORT SQL-Einfügungen verwendet werden, protokolliert DB2 die Aktivität, und es sind keine Sicherungen erforderlich, um diese Operationen bei Fehlern wiederherzustellen.	Ein Sicherungsabbild kann während der Ladeoperation erstellt werden.

## Unterschiede zwischen den Dienstprogrammen IMPORT und LOAD

---

## Anhang C. Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD

Es werden vier Betriebssystemdateiformate beschrieben, die von den DB2-Dienstprogrammen EXPORT, IMPORT und LOAD unterstützt werden:

**DEL** ASCII-Format ohne universelle Zeilenbegrenzer. Dieses Format ermöglicht den Datenaustausch zwischen einer Vielzahl von Datenbankmanagern und Dateimanagern. Bei diesem allgemeinen Ansatz zum Speichern von Daten werden spezielle Zeichenbegrenzer verwendet, um Spaltenwerte zu trennen.

**ASC** ASCII-Format mit universellen Zeilenbegrenzern. Dieses Format dient zum Importieren oder Laden von Daten anderer Anwendungen, die Flachdateien mit ausgerichteten Spaltendaten erstellen.

### **PC/IXF**

PC-Version des IXF-Formats (Integrated Exchange Format). Dies ist das bevorzugte Format für den Austausch von Daten innerhalb des Datenbankmanagers. PC/IXF ist eine strukturierte Beschreibung einer Datenbanktabelle, die eine externe Darstellung der internen Tabelle enthält.

**WSF** Arbeitsblattformat. Dieses Format ermöglicht den Datenaustausch mit Produkten wie Lotus 1-2-3 and Symphony. Dieses Dateiformat wird vom Dienstprogramm LOAD nicht unterstützt.

Bei Verwendung der Datendateiformate DEL, WSF oder ASC müssen Sie die Tabelle, einschließlich der Spaltennamen und Datentypen, definieren, bevor Sie die Datei importieren. Die Datentypen in den Feldern der Betriebssystemdatei werden in den entsprechenden Datentyp in der Datenbanktabelle umgesetzt. Das Dienstprogramm IMPORT akzeptiert Daten mit geringfügigen Inkompatibilitätsproblemen, einschließlich Zeichendaten, die beim Import möglicherweise aufgefüllt bzw. abgeschnitten werden, und numerischer Daten, die in unterschiedliche Typen numerischer Felder importiert werden.

Bei Verwendung des Datendateiformats PC/IXF muß die Tabelle beim Starten der Importoperation noch nicht vorhanden sein. Benutzerdefinierte einzigartige Typen (UDTs - User-defined Types) werden nicht in die Spaltentypen der neuen Tabelle aufgenommen. Statt dessen wird der Basistyp verwendet. Auf ähnliche Weise werden UDTs beim Exportieren in das Datendateiformat PC/IXF in der PC/IXF-Datei als Basisdatentypen gespeichert.

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

---

### ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

Eine DEL-Datei ist eine sequentielle ASCII-Datei mit Zeilen- und Spaltenbegrenzungszeichen. Jede DEL-Datei ist ein Datenstrom aus ASCII-Zeichen, der sich aus Zellenwerten zusammensetzt, die zuerst nach Zeilen und dann nach Spalten geordnet sind. Die Zeilen im Datenstrom werden durch Zeilenbegrenzungszeichen getrennt. Die einzelnen Zellenwerte innerhalb einer jeden Zeile werden durch Spaltenbegrenzungszeichen getrennt.

Die folgende Tabelle beschreibt das Format von DEL-Dateien, die importiert oder als Ergebnis einer Exportaktion generiert werden können.

DEL-Datei ::= Zeile 1 Daten || Zeilenbegrenzer ||  
              Zeile 2 Daten || Zeilenbegrenzer ||  
              .  
              .  
              .  
              Zeile n Daten || Wahlfreier Zeilenbegrenzer

Zeile i Daten ::= Zellenwert(i,1) || Spaltenbegrenzer ||  
                  Zellenwert(i,2) || Spaltenbegrenzer ||  
                  .  
                  .  
                  .  
                  Zellenwert(i,m)

Zeilenbegrenzer ::= ASCII-Zeilenvorschubfolge<sup>a</sup>

Spaltenbegrenzer ::= Standardwert: ASCII-Komma (,)<sup>b</sup>

Zellenwert(i,j) ::= Führende Leerzeichen  
                  || ASCII-Darstellung eines numerischen Werts  
                  ( Integer, Dezimal oder Gleitkomma)  
                  || Begrenzte Zeichenfolge  
                  || Nicht begrenzte Zeichenfolge  
                  || Nachgestellte Leerzeichen

Nicht begrenzte Zeichenfolge ::= Ein Satz beliebiger Zeichen außer eines Zeilen- oder Spaltenbegrenzers

Begrenzte Zeichenfolge ::= Ein Zeichenfolgebegrenzer ||  
                          Eine erweiterte Zeichenfolge ||  
                          Ein Zeichenfolgebegrenzer ||  
                          Nicht verwertbarer Rest

Nicht verwertbarer Rest ::= Ein Satz beliebiger Zeichen außer eines Zeilen- oder Spaltenbegrenzers

Zeichenfolgebegrenzer ::= Standardwert: Doppelte ASCII-Anführungszeichen (")<sup>c</sup>

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

Erweiterte Zeichenfolge ::= || Ein Satz beliebiger Zeichen außer eines Zeilen- oder Zeichenfolgebegrenzers, wenn der Wert NODOUBLEDEL angegeben ist  
|| Ein Satz beliebiger Zeichen außer eines Zeilen- oder Zeichenfolgebegrenzers, wenn die Zeichenfolge nicht Teil zweier aufeinanderfolgender Zeichenfolgebegrenzer ist  
|| Ein Satz beliebiger Zeichen außer eines Zeichenfolgebegrenzers, wenn der Zeichenfolgebegrenzer nicht Teil zweier aufeinanderfolgender Zeichenfolgebegrenzer ist und der Wert DELPRIORITYCHAR angegeben ist

Dateiendezeichen ::= Hex '1A' (nur OS/2 oder Windows-Betriebssysteme)

ASCII-Darstellung eines numerischen

Werts<sup>d</sup> ::= Wahlfreies

Vorzeichen '+' oder '-'

|| 1 bis 31 Dezimalziffern mit wahlfreiem Dezimalzeichen vor,  
nach oder zwischen zwei Ziffern

|| Wahlfreier Exponent

Exponent ::= Zeichen 'E' oder 'e'

|| Wahlfreies Zeichen '+' oder '-'

|| 1 bis 3 Dezimalziffern ohne Dezimalzeichen

Dezimalziffer ::= Beliebiges Zeichen '0', '1', ... '9'

Dezimalzeichen ::= Standardwert: ASCII-Punkt (.)<sup>e</sup>

- <sup>a</sup> Als Satzbegrenzer wird das Zeilenvorschubzeichen, ASCII-Wert x0A, angenommen. Unter OS/2 oder den Windows-Betriebssystemen generierte Daten können den 2-Byte-Standardwert für Rücklauf/Zeilenvorschub, 0x0D0A, verwenden. Daten in EBCDIC-Codepages müssen das EBCDIC-Zeichen für Zeilenvorschub (0x25) als Satzbegrenzer verwenden (EBCDIC-Daten können mit der Option CODEPAGE des Befehls LOAD geladen werden).
- <sup>b</sup> Das Spaltenbegrenzungszeichen kann mit der Option COLDEL angegeben werden.
- <sup>c</sup> Das Zeichenfolgebegrenzungszeichen kann mit der Option CHARDEL angegeben werden.

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

**Anmerkung:** Die Standardpriorität der Begrenzer ist:

1. Satzbegrenzer
2. Zeichenbegrenzer
3. Spaltenbegrenzer

Siehe dazu auch die Beschreibung des Wertes `delprioritychar` in Tabelle 8 auf Seite 140.

- <sup>d</sup> Wenn die ASCII-Darstellung eines numerischen Wertes einen Exponenten aufweist, handelt es sich um eine FLOAT-Konstante. Wenn sie ein Dezimalzeichen, jedoch keinen Exponenten aufweist, handelt es sich um eine DECIMAL-Konstante. Wenn sie kein Dezimalzeichen und keinen Exponenten aufweist, handelt es sich um eine INTEGER-Konstante.
- <sup>e</sup> Das Dezimalzeichen kann mit der Option DECPT angegeben werden.

### Beispiel für DEL-Datei

Es folgt ein Beispiel für eine DEL-Datei. Jede Zeile endet mit einer Zeilenvorschubfolge (unter OS/2 oder den Windows-Betriebssystemen endet jede Zeile mit einer Folge aus einem Rücklauf- und einem Zeilenvorschubzeichen).

```
"Smith, Bob",4973,15.46  
"Jones, Bill",12345,16.34  
"Williams, Sam",452,193.78
```

Das folgende Beispiel zeigt die Verwendung von nicht begrenzten Zeichenfolgen. Das Spaltenbegrenzungszeichen wurde durch ein Semikolon ersetzt, weil die Zeichendaten ein Komma enthalten.

```
Smith, Bob;4973;15.46  
Jones, Bill;12345;16.34  
Williams, Sam;452;193.78
```

### Anmerkungen:

1. Ein Leerzeichen (X'20') ist als Begrenzer nie zulässig.
2. Leerzeichen vor dem ersten oder nach dem letzten Zeichen eines Zellenwertes werden beim Import gelöscht. Leerzeichen, die in einen Zellenwert eingebettet sind, werden nicht gelöscht.
3. Ein Punkt (.) ist kein gültiges Zeichenfolgebegrenzungszeichen, da dies zu Konflikten mit den Punkten in den Werten für Zeitmarken führen würde.
4. Bei reinem DBCS (Grafik), gemischtem DBCS und EUC sind Begrenzer auf den Bereich von x00 bis x3F (jeweils einschließlich) beschränkt.

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

5. Bei DEL-Daten, die in einer EBCDIC-Codepage angegeben werden, dürfen die Begrenzer nicht mit den DBCS-Startzeichen und -Endezeichen identisch sein.
6. Unter OS/2 oder den Windows-Betriebssystemen gibt das erste Dateieindezeichen (X'1A'), das sich nicht innerhalb von Zeichenbegrenzern befindet, das Dateieinde an. Darauf folgende Daten werden nicht importiert.
7. Ein Nullwert wird durch das Fehlen eines Zellenwerts an einer Stelle, an der sich normalerweise ein Zellenwert befände, oder durch eine Folge aus Leerzeichen gekennzeichnet.
8. Da einige andere Produkte die Länge von Zeichenfeldern auf 254 oder 255 Byte begrenzen, gibt das Dienstprogramm EXPORT eine Warnung aus, wenn eine Zeichenspalte mit einer Länge von mehr als 254 Zeichen für den Export ausgewählt wird. Das Dienstprogramm IMPORT verarbeitet Felder, die so lang wie die längste LONG VARCHAR- und LONG VARGRAPHIC-Spalte sind.

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

### Beschreibungen von DEL-Datentypen

Tabelle 9. Gültige Datentypformate für das Dateiformat DEL

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	Eine INTEGER-Konstante im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807.	ASCII-Darstellung eines numerischen Wertes im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.
BLOB, CLOB	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
BLOB_FILE, CLOB_FILE	Die Zeichendaten für jede BLOB-/CLOB-Spalte werden in einzelnen Dateien gespeichert, und der Dateiname ist in Zeichenbegrenzer eingeschlossen.	Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.
CHAR	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird nach Bedarf abgeschnitten oder mit Leerzeichen (X'20') aufgefüllt und so an die Breite der Datenbankspalte angeglichen.
DATE	<i>jjjmmtt</i> (Jahr Monat Tag) ohne Spaltenbegrenzer. Beispiel: 19931029  Alternativ kann die Option DATESISO verwendet werden, um festzulegen, daß alle Datumswerte im ISO-Format exportiert werden sollen.	Eine begrenzte oder nicht begrenzte Zeichenfolge, die einen Datumswert in einem ISO-Format enthält, das mit dem Landescode der Zieldatenbank konsistent ist, oder eine nicht begrenzte Zeichenfolge im Format <i>jjjmmtt</i> .



## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

Tabelle 9. Gültige Datentypformate für das Dateiformat DEL (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
DBCLOB (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
DBCLOB_FILE (nur DBCS)	Die Zeichendaten für jede DBCLOB-Spalte werden in einzelnen Dateien gespeichert, und der Dateiname ist in Zeichenbegrenzer eingeschlossen.	Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.
DECIMAL	Eine DECIMAL-Konstante mit der Genauigkeit und der Anzahl der Kommastellen des Feldes, das exportiert wird. Die Option DECPLUSBLANK kann verwendet werden, um anzugeben, daß positiven Dezimalwerten ein Leerzeichen statt eines Pluszeichens (+) vorangestellt werden soll.	ASCII-Darstellung eines numerischen Werts, der nicht außerhalb des Bereichs der Datenbankspalte liegt, in die das Feld importiert wird. Wenn der Eingabewert mehr Ziffern nach dem Dezimalzeichen enthält, als in der Datenbankspalte Platz finden, werden die überzähligen Ziffern abgeschnitten.
FLOAT(long)	Eine FLOAT-Konstante im Bereich -10E307 bis 10E307.	ASCII-Darstellung eines numerischen Wertes im Bereich -10E307 bis 10E307.
GRAPHIC (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird nach Bedarf abgeschnitten oder mit Doppelbyteleerzeichen (X'8140') aufgefüllt und so an die Breite der Datenbankspalte angeglichen.

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

Tabelle 9. Gültige Datentypformate für das Dateiformat DEL (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
INTEGER	Eine INTEGER-Konstante im Bereich -2 147 483 648 bis 2 147 483 647.	ASCII-Darstellung eines numerischen Werts im Bereich -2 147 483 648 bis 2 147 483 647. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.
LONG VARCHAR	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
LONG VARGRAPHIC (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird als Wert der Datenbankspalte verwendet.
SMALLINT	Eine INTEGER-Konstante im Bereich -32 768 bis 32 767.	ASCII-Darstellung eines numerischen Werts im Bereich -32 768 bis 32 767. Dezimal- und Gleitkommazahlen werden ganzzahlig abgeschnitten.
TIME	ss.mm.ss (Stunde Minuten Sekunden). Ein Zeitwert im ISO-Format, in Zeichenbegrenzer eingeschlossen. Beispiel: "09.39.43"	Eine begrenzte oder nicht begrenzte Zeichenfolge, die einen Zeitwert in einem Format enthält, das mit dem Landescode der Zieldatenbank konsistent ist.
TIMESTAMP	jjjj-mm-tt-ss.mm.ss.nnnnnn (Jahr Monat Tag Stunde Minuten Sekunden Mikrosekunden). Eine Zeichenfolge, die ein Datum und eine Uhrzeit darstellt, in Zeichenbegrenzer eingeschlossen.	Eine begrenzte oder nicht begrenzte Zeichenfolge, die einen Zeitmarkenwert enthält, dessen Speicherung in einer Datenbank zulässig ist.

## ASCII-Format ohne universelle Zeilenbegrenzer (DEL)

Tabelle 9. Gültige Datentypformate für das Dateiformat DEL (Forts.)

<b>Datentyp</b>	<b>Format in vom Dienstprogramm EXPORT erstellten Dateien</b>	<b>Für das Dienstprogramm IMPORT zulässiges Format</b>
VARCHAR	Zeichendaten, die in Zeichenbegrenzer (z. B. doppelte Anführungszeichen) eingeschlossen sind.	Eine begrenzte oder nicht begrenzte Zeichenfolge. Die Zeichenfolge wird nach Bedarf abgeschnitten und so an die maximal zulässige Breite der Datenbankspalte angeglichen.
VARGRAPHIC (nur DBCS)	Grafikdaten werden als begrenzte Zeichenfolge exportiert.	Eine begrenzte oder nicht begrenzte Zeichenfolge (Länge ist eine gerade Anzahl Byte). Die Zeichenfolge wird nach Bedarf abgeschnitten und so an die maximal zulässige Breite der Datenbankspalte angeglichen.

## ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC)

---

### ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC)

Eine ASC-Datei ist eine sequentielle ASCII-Datei mit Zeilenbegrenzungszeichen. Sie kann für den Austausch von Daten mit beliebigen ASCII-Produkten verwendet werden, die Daten im Spaltenformat verarbeiten können (einschließlich Textverarbeitungsprogrammen). Jede ASC-Datei ist ein Datenstrom aus ASCII-Zeichen, der sich aus Datenwerten zusammensetzt, die in Zeilen und Spalten angeordnet sind. Die Zeilen innerhalb des Datenstroms werden durch Zeilenbegrenzungszeichen voneinander getrennt. Jede Spalte innerhalb einer Zeile wird durch ein Wertepaar bestehend aus Anfangsposition-Endposition (mit IMPORT-Parametern angegeben) definiert. Jedes Paar stellt Speicherpositionen innerhalb einer Zeile dar, die als Bytepositionen angegeben werden. Die erste Position innerhalb einer Zeile ist Byteposition 1. Das erste Element jedes Positionspaares ist das Byte in der Zeile, bei dem die Spalte beginnt, und das zweite Element ist das Byte, bei dem die Spalte endet. Die Spalten können sich überlappen. Jede Zeile in einer ASC-Datei hat die gleiche Spaltendefinition.

Eine ASC-Datei ist durch folgendes definiert:

```
ASC-Datei ::= Zeile 1 Daten || Zeilenbegrenzer ||  
            Zeile 2 Daten || Zeilenbegrenzer ||  
            .  
            .  
            Zeile n Daten
```

```
Zeile i Daten ::= ASCII-Zeichen || Zeilenbegrenzer
```

```
Zeilenbegrenzer ::= ASCII-Zeilenvorschubfolgea
```

- <sup>a</sup> Als Satzbegrenzer wird das Zeilenvorschubzeichen, ASCII-Wert x0A, angenommen. Unter OS/2 oder den Windows-Betriebssystemen generierte Daten können den 2-Byte-Standardwert für Rücklauf/Zeilenvorschub, 0x0D0A, verwenden. Daten in EBCDIC-Codepages müssen das EBCDIC-Zeichen für Zeilenvorschub (0x25) als Satzbegrenzer verwenden (EBCDIC-Daten können mit der Option CODEPAGE des Befehls LOAD geladen werden). Der Satzbegrenzer wird nie als Teil eines Feldes mit Daten gewertet.

### Beispiel für ASC-Datei

Es folgt ein Beispiel für eine ASC-Datei. Jede Zeile endet mit einer Zeilenvorschubfolge (unter OS/2 oder den Windows-Betriebssystemen endet jede Zeile mit einer Folge aus einem Rücklauf- und einem Zeilenvorschubzeichen).

Smith, Bob	4973	15.46
Jones, Suzanne	12345	16.34
Williams, Sam	452123	193.78

### Anmerkungen:

1. Bei ASC-Dateien wird davon ausgegangen, daß sie keine Spaltennamen enthalten.
2. Zeichenfolgen sind *nicht* in Begrenzer eingeschlossen. Der Datentyp einer Spalte in der ASC-Datei wird durch den Datentyp der Zielspalte in der Datenbanktabelle festgelegt.
3. NULL wird unter folgenden Bedingungen in einer Datenbankspalte, die Nullwerte enthalten darf, importiert:
  - Ein Feld mit Leerzeichen ist für eine numerische Datenbankspalte oder eine Datenbankspalte des Typs DATE, TIME oder TIMESTAMP bestimmt.
  - Es wird ein Feld ohne Anfangs- und Endpositionspaar angegeben.
  - Es wird ein Positionspaar mit Anfangs- und Endpositionen gleich Null angegeben.
  - Eine Datenzeile ist zu kurz, um einen gültigen Wert für die Zielspalte zu enthalten.
  - Die LOAD-Option NULL INDICATORS wird verwendet und die Nullanzeigerspalte enthält N (oder einen anderen vom Benutzer angegebenen Wert).
4. Wenn die Zielspalte keinen Nullwert enthalten darf, führt der Versuch, ein Feld mit Leerzeichen in eine numerische Spalte oder eine Spalte des Typs DATE, TIME oder TIMESTAMP zu importieren, zum Zurückweisen der Zeile.
5. Wenn die Eingabedaten nicht mit der Zielspalte kompatibel sind und diese Spalte Nullwerte enthalten darf, wird eine Null importiert oder die Zeile zurückgewiesen, je nachdem, wo der Fehler erkannt wird. Wenn die Spalte keine Nullwerte enthalten darf, wird die Zeile zurückgewiesen. Nachrichten mit Angaben zu den festgestellten Inkompatibilitäten werden in die Nachrichtendatei geschrieben.

## ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC)

### Beschreibungen von ASC-Datentypen

Tabelle 10. Gültige Datentypformate für das Dateiformat ASC

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807 liegen. Dezimalzahlen werden ganzzahlig abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
BLOB/CLOB	<p>Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen. Wenn die ASC-Option zum <i>Abschneiden von Leerzeichen</i> wirksam ist, werden abschließende Leerzeichen von der ursprünglichen oder der abgeschnittenen Zeichenfolge entfernt.</p>
BLOB_FILE, CLOB_FILE, DBCLOB_FILE (nur DBCS)	<p>Der begrenzte oder nicht begrenzte Name der Datei, die die Daten enthält.</p>
CHAR	<p>Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten oder mit Leerzeichen aufgefüllt und so an die Breite der Zielspalte angeglichen.</p>
DATE	<p>Eine Zeichenfolge, die einen Datumswert in einem Format enthält, das mit dem Landescape der Zieldatenbank konsistent ist.</p> <p>Die Anfangs- und Endposition müssen eine Feldbreite festlegen, die im Bereich der externen Darstellung eines Datums liegt.</p>
DBCLOB (nur DBCS)	<p>Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen.</p>

## ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC)

Table 10. Gültige Datentypformate für das Dateiformat ASC (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
DECIMAL	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich der Datenbankspalte liegen, in die sie importiert werden. Wenn der Eingabewert mehr Ziffern nach dem Dezimalzeichen enthält, als in der Datenbankspalte Platz finden, werden die überzähligen Ziffern abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
FLOAT(long)	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Alle Werte sind gültig. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Ein E wird unabhängig von der Groß-/Kleinschreibung als Anfang des Exponenten einer FLOAT-Konstante akzeptiert.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>
GRAPHIC (nur DBCS)	<p>Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten oder mit Doppelbyteleerzeichen (0x8140) aufgefüllt und so an die maximal zulässige Länge der Zielspalte angeglichen.</p>
INTEGER	<p>Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -2 147 483 648 bis 2 147 483 647 liegen. Dezimalzahlen werden ganzzahlig abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.</p> <p>Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.</p>

## ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC)

Tabelle 10. Gültige Datentypformate für das Dateiformat ASC (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
LONG VARCHAR	Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen. Wenn die ASC-Option zum <i>Abschneiden von Leerzeichen</i> wirksam ist, werden abschließende Leerzeichen von der ursprünglichen oder der abgeschnittenen Zeichenfolge entfernt.
LONG VARGRAPHIC (nur DBCS)	Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen.
SMALLINT	Eine Konstante eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -32 768 bis 32 767 liegen. Dezimalzahlen werden ganzzahlig abgeschnitten. Ein Komma, Punkt oder Doppelpunkt gilt als Dezimalzeichen. Tausendertrennzeichen sind nicht zulässig.  Die Anfangs- und Endposition müssen ein Feld festlegen, dessen Breite 50 Byte nicht überschreitet. Ganze Zahlen, Dezimalzahlen und die Mantissen von Gleitkommazahlen dürfen nicht mehr als 31 Stellen haben. Die Exponenten von Gleitkommazahlen dürfen nicht mehr als 3 Stellen haben.
TIME	Eine Zeichenfolge, die einen Zeitwert in einem Format enthält, das mit dem Landescode der Zieldatenbank konsistent ist.  Die Anfangs- und Endposition müssen eine Feldbreite festlegen, das im Bereich der externen Darstellung eines Zeitwerts liegt.
TIMESTAMP	Eine Zeichenfolge, die einen Zeitmarkenwert enthält, dessen Speicherung in einer Datenbank zulässig ist.  Die Anfangs- und Endposition müssen eine Feldbreite festlegen, die im Bereich der externen Darstellung einer Zeitmarke liegt.
VARCHAR	Eine Zeichenfolge. Die Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen. Wenn die ASC-Option zum <i>Abschneiden von Leerzeichen</i> wirksam ist, werden abschließende Leerzeichen von der ursprünglichen oder der abgeschnittenen Zeichenfolge entfernt.



## ASCII-Dateiformat mit universellen Zeilenbegrenzern (ASC)

Tabelle 10. Gültige Datentypformate für das Dateiformat ASC (Forts.)

Datentyp	Für das Dienstprogramm IMPORT zulässiges Format
VARGRAPHIC (nur DBCS)	Eine Zeichenfolge mit einer geraden Anzahl Byte. Eine Zeichenfolge mit einer ungeraden Anzahl Byte ist ungültig und wird abgelehnt. Eine gültige Zeichenfolge wird nach Bedarf rechts abgeschnitten und so an die maximal zulässige Länge der Zielspalte angeglichen.

---

### PC-Version des IXF-Dateiformats

Die PC-Version des IXF-Dateiformats (PC/IXF) ist eine Datenbankmanageranpassung der IXF-Datenaustauscharchitektur (Integration Exchange Format). Die IXF-Architektur ist speziell auf den Austausch von Strukturen und Daten relationaler Datenbanken ausgelegt. Die PC/IXF-Architektur ermöglicht es dem Datenbankmanager, eine Datenbank zu exportieren, ohne daß die Anforderungen und Eigenarten eines Zielprodukts bekannt sein müssen. Auch ein Produkt, in das eine PC/IXF-Datei importiert wird, muß nur die PC/IXF-Architektur beherrschen. Die Merkmale des Produkts, von dem die Datei exportiert wurde, sind nicht von Bedeutung. Die PC/IXF-Dateiarchitektur wahrt die Unabhängigkeit sowohl der exportierenden als auch der importierenden Datenbanksysteme.

Die IXF-Architektur stellt ein generisches Austauschformat für relationale Datenbanken dar, das eine Vielzahl von Typen relationaler Daten unterstützt, zu denen auch Typen zählen, die von bestimmten Produkten für relationale Datenbanken möglicherweise nicht unterstützt werden. Das PC/IXF-Dateiformat wahrt diese Flexibilität. Beispielsweise unterstützt die PC/IXF-Architektur sowohl SBCS-Datentypen (Einzelbytezeichenfolge) als auch DBCS-Datentypen (Doppelbytezeichenfolge). Nicht alle Implementierungen unterstützen alle PC/IXF-Datentypen. Selbst eingeschränkte Implementierungen erlauben jedoch die Erkennung und Disposition von nicht unterstützten Datentypen beim Import.

Im allgemeinen besteht eine PC/IXF-Datei aus einer ununterbrochenen Folge von Datensätzen mit variabler Länge. Die Datei enthält die folgenden Datensatztypen in der gezeigten Reihenfolge:

- Einen Kennsatz der Satzart H
- Einen Tabellensatz der Satzart T
- Mehrere Spaltendeskriptorsätze der Satzart C (ein Datensatz für jede Spalte in der Tabelle)
- Mehrere Datensätze der Satzart D (jede Zeile in der Tabelle wird durch einen oder mehrere D-Datensätze dargestellt)

## PC-Version des IXF-Dateiformats

Eine PC/IXF-Datei kann an einer beliebigen Stelle hinter dem H-Satz auch Anwendungsdatensätze der Satzart A enthalten. Diese Datensätze sind in PC/IXF-Dateien zulässig, so daß eine Anwendung zusätzliche Daten, die im PC/IXF-Format nicht definiert sind, in eine PC/IXF-Datei einfügen kann. A-Datensätze werden von jedem Programm ignoriert, das eine PC/IXF-Datei liest und keine Informationen zum Datenformat und Inhalt besitzt, das bzw. der durch die Anwendungskennung im A-Datensatz impliziert wird.

Jeder Datensatz in einer PC/IXF-Datei beginnt mit einem Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große rechtsbündige Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Programme, die PC/IXF-Dateien lesen, müssen diese Satzlängen verwenden, um das Ende des aktuellen Datensatzes und den Anfang des nächsten Datensatzes zu bestimmen. H-, T- und C-Datensätze müssen groß genug für alle definierten Felder sein, und die Satzlängenfelder müssen selbstverständlich mit den Ist-Längen übereinstimmen. Wenn jedoch zusätzliche Daten (z. B. ein *neues* Feld) am Ende eines dieser Datensätze angefügt wird, müssen vorhandene Programme, die PC/IXF-Dateien lesen, die zusätzlichen Daten ignorieren und lediglich eine Warnung generieren. Programme hingegen, die PC/IXF-Dateien schreiben, müssen H-, T- und C-Datensätze mit genau der für alle definierten Felder erforderlichen Länge schreiben.

PC/IXF-Dateisätze bestehen aus Feldern, die Zeichendaten enthalten. Die Dienstprogramme IMPORT und EXPORT interpretieren diese Zeichendaten mit Hilfe der CPGID der Zieldatenbank, wobei es zwei Ausnahmen gibt:

- Das IXFADATA-Feld von A-Datensätzen.  
Die Codepage-Umgebung von Zeichendaten in einem IXFADATA-Feld wird von der Anwendung festgelegt, die einen bestimmten A-Datensatz erstellt und verarbeitet. Das bedeutet, daß die Umgebung je nach Implementierung variiert.
- Das IXFDCOLS-Feld von D-Datensätzen.  
Die Codepage-Umgebung von Zeichendaten in einem IXFDCOLS-Feld ist eine Funktion der im C-Datensatz enthaltenen Informationen. Dieser Datensatz definiert eine bestimmte Spalte und ihre Daten.

Numerische Felder in H-, T- und C-Datensätzen sowie im Präfixteil von D- und A-Datensätzen müssen rechtsbündige Einzelbytezeichendarstellungen von ganzzahligen Werten sein, die mit führenden Nullen oder Leerzeichen aufgefüllt sind. Nullwerte müssen mit mindestens einem (rechtsbündigen) Nullzeichen, nicht mit Leerzeichen, angegeben werden. Wenn eines dieser numerischen Felder, z. B. IXFCLENG, dessen Länge vom Datentyp impliziert wird, nicht verwendet wird, muß es mit Leerzeichen aufgefüllt werden. Diese numerischen Felder sind:

IXFHRECL, IXFTRECL, IXFCRECL, IXFDRECL, IXFARECL,  
 IXFHCNT, IXFHSBCP, IXFHDBCP, IXFTCCNT, IXFTNAML,  
 IXFCLENG, IXFCDRID, IXFCPOSN, IXFCNAML, IXFCTYPE,  
 IXFCSBCP, IXFCDBCP, IXFCNDIM, IXFCDSIZ, IXFDRID

**Anmerkung:** Das PC/IXF-Dateiformat des Datenbankmanagers ist nicht identisch mit dem IXF-Format von IBM System /370 (siehe „Unterschiede zwischen PC/IXF und Version 0 System /370 IXF“ auf Seite 308).

### PC/IXF-Satzarten

Es gibt fünf PC/IXF-Basissatzarten:

- Kennsatz
- Tabellensatz
- Spaltendeskriptorsatz
- Datensatz
- Anwendungssatz

Außerdem werden von DB2 UDB sechs Unterarten von Anwendungssätzen verwendet:

- Indexsatz
- Hierarchiesatz
- Satz für untergeordnete Tabelle
- Fortsetzungssatz
- Beendigungssatz
- Identitätssatz

Jede PC/IXF-Satzart ist als eine Folge von Feldern definiert. Diese Felder sind erforderlich und müssen in der gezeigten Reihenfolge vorkommen.

#### KENNSATZ

FELDNAME	LÄNGE	ART	KOMMENTARE
-----	-----	-----	-----
IXFHRECL	06 BYTE	CHARACTER	Satzlänge
IXFHRECT	01 BYTE	CHARACTER	Satzart = 'H'
IXFHID	03 BYTE	CHARACTER	IXF-Kennung
IXFHVERS	04 BYTE	CHARACTER	IXF-Version
IXFHPROD	12 BYTE	CHARACTER	Produkt
IXFHDATE	08 BYTE	CHARACTER	Erstellungsdatum
IXFHTIME	06 BYTE	CHARACTER	Erstellungszeit
IXFHCNT	05 BYTE	CHARACTER	Anzahl vorhergehender Datensätze
IXFHSBCP	05 BYTE	CHARACTER	Einzelbyte-Codepage
IXFHDBCP	05 BYTE	CHARACTER	Doppelbyte-Codepage
IXFHFIL1	02 BYTE	CHARACTER	reserviert

## PC-Version des IXF-Dateiformats

### **IXFHRECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der H-Datensatz muß lang genug sein für alle darin definierten Felder.

### **IXFHRECT**

Die IXF-Satzart, die für diesen Datensatz auf H gesetzt ist.

### **IXFHID**

Die Dateiformatkennung, die für diese Datei auf IXF gesetzt ist.

### **IXFHVERS**

Die Version des PC/IXF-Formats, die bei der Erstellung der Datei verwendet wurde, ist auf '0002' gesetzt.

### **IXFHPROD**

Ein Feld, das von dem Programm, das die Datei erstellt, zu Identifikationszwecken verwendet werden kann. Wenn dieses Feld ausgefüllt wird, dienen die ersten sechs Byte dazu, das Produkt anzugeben, von dem die Datei erstellt wurde. Die letzten sechs Byte dienen dazu, die Version bzw. das Release des Produkts anzugeben. Der Datenbankmanager verwendet dieses Feld, um das Vorhandensein von datenbankmanagerspezifischen Daten anzuzeigen.

### **IXFHDATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*.

### **IXFHTIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld ist wahlfrei und kann leergelassen werden.

### **IXFHHCNT**

Die Anzahl der H-, T- und C-Datensätze in dieser Datei, die vor dem ersten Datensatz kommen. A-Datensätze sind bei diesem Wert nicht berücksichtigt.

### **IXFHSBCP**

Einzelbyte-Codepage-Feld, das eine Einzelbytezeichendarstellung einer SBCS-CPGID oder '00000' enthält.

Das Dienstprogramm EXPORT setzt dieses Feld mit der SBCS-CPGID der exportierten Datenbanktabelle gleich. Beispiel: Wenn die SBCS-CPGID den Wert 850 hat, enthält dieses Feld '00850'.

**IXFHDBCP**

Doppelbyte-Codepage-Feld, das eine Einzelbytezeichendarstellung einer DBCS-CPGID oder '00000' enthält.

Das Dienstprogramm EXPORT setzt dieses Feld mit der DBCS-CPGID der exportierten Datenbanktabelle gleich. Beispiel: Wenn die DBCS-CPGID den Wert 301 hat, enthält dieses Feld '00301'.

**IXFHFIL1**

Zusatzfeld, das zwei Leerzeichen enthält. Dient als Entsprechung zu einem reservierten Feld in Host-IXF-Dateien.

TABELLENSATZ

FELDDNAME	LÄNGE	ART	KOMMENTARE
-----	-----	-----	-----
IXFTRECL	006 BYTE	CHARACTER	Satzlänge
IXFTRECT	001 BYTE	CHARACTER	Satzart = 'T'
IXFTNAML	003 BYTE	CHARACTER	Namenslänge
IXFTNAME	256 BYTE	CHARACTER	Name der Daten
IXFTQULL	003 BYTE	CHARACTER	Länge Qualifikationsmerkmal
IXFTQUAL	256 BYTE	CHARACTER	Qualifikationsmerkmal
IXFTSRC	012 BYTE	CHARACTER	Datenquelle
IXFTDATA	001 BYTE	CHARACTER	Datenkonvention = 'C'
IXFTFORM	001 BYTE	CHARACTER	Datenformat = 'M'
IXFTMFRM	005 BYTE	CHARACTER	Maschinenformat = 'PC'
IXFTLOC	001 BYTE	CHARACTER	Datenspeicherposition = 'I'
IXFTCCNT	005 BYTE	CHARACTER	'C'-Satzzählung
IXFTFIL1	002 BYTE	CHARACTER	reserviert
IXFTDESC	030 BYTE	CHARACTER	Datenbeschreibung
IXFTPKNM	257 BYTE	CHARACTER	Primärschlüsselname
IXFTDSPC	257 BYTE	CHARACTER	reserviert
IXFTISPC	257 BYTE	CHARACTER	reserviert
IXFTLSPC	257 BYTE	CHARACTER	reserviert

Die folgenden Felder sind im Tabellensatz enthalten:

**IXFTRECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der T-Datensatz muß lang genug sein für alle darin definierten Felder.

**IXFTRECT**

Die IXF-Satzart, die für diesen Datensatz auf T gesetzt ist.

**IXFTNAML**

Die Länge des Tabellennamens im IXFTNAME-Feld in Byte.

### IXFTNAME

Der Name der Tabelle. Wenn jede Datei nur eine Tabelle enthält, dient dieses Feld nur zur Information. Dieses Feld wird vom Datenbankmanager beim Importieren von Daten nicht verwendet. Beim Schreiben einer PC/IXF-Datei schreibt der Datenbankmanager den DOS-Dateinamen (und möglicherweise Pfadinformationen) in dieses Feld.

### IXFTQULL

Die Länge des Qualifikationsmerkmals für den Tabellennamen im IXFTQUAL-Feld in Byte.

### IXFTQUAL

Qualifikationsmerkmal für den Tabellennamen, das den Ersteller einer Tabelle in einem relationalen System angibt. Dieses Feld dient nur zur Information. Wenn ein Programm, das eine Datei schreibt, keine Daten in dieses Feld schreiben muß, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen.

### IXFTSRC

Wird verwendet, um die ursprüngliche Quelle der Daten anzugeben. Dieses Feld dient nur zur Information. Wenn ein Programm, das eine Datei schreibt, keine Daten in dieses Feld schreiben muß, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen.

### IXFTDATA

Konvention, die zum Beschreiben der Daten verwendet wird. Dieses Feld muß für den Import und Export auf C gesetzt sein, um anzugeben, daß einzelne Spaltenattribute in den folgenden Spaltendeskriptorsätzen (C) beschrieben werden und daß Daten den PC/IXF-Konventionen entsprechen.

### IXFTFORM

Konvention, die zum Speichern von numerischen Daten verwendet wird. Dieses Feld muß auf M gesetzt sein, um anzugeben, daß numerische Daten in den Datensätzen (D) in dem (internen) Maschinenformat gespeichert sind, das im IXFTMFRM-Feld angegeben ist.

### IXFTMFRM

Das Format beliebiger Maschinendaten in der PC/IXF-Datei. Der Datenbankmanager liest oder schreibt Dateien nur, wenn dieses Feld auf PC*bbb* gesetzt ist, wobei *b* ein Leerzeichen ist und PC angibt, daß Daten in der PC/IXF-Datei in einem IBM PC-Maschinenformat vorliegen.

**IXFTLOC**

Die Speicherposition der Daten. Der Datenbankmanager unterstützt nur den Wert I, was bedeutet, daß sich die Daten in dieser Datei befinden.

**IXFTCCNT**

Die Anzahl von C-Datensätzen in dieser Tabelle. Hierbei handelt es sich um eine rechtsbündige Zeichendarstellung eines ganzzahligen Wertes.

**IXFTFIL1**

Zusatzfeld, das zwei Leerzeichen enthält. Dient als Entsprechung zu einem reservierten Feld in Host-IXF-Dateien.

**IXFTDESC**

Beschreibende Daten zur Tabelle. Dieses Feld dient nur zur Information. Wenn ein Programm, das eine Datei schreibt, keine Daten in dieses Feld schreiben muß, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen. Dieses Feld enthält NOT NULL WITH DEFAULT, wenn für die Spalte NOT NULL WITH DEFAULT angegeben war und der Tabellename nicht aus einer Workstation-Datenbank stammt.

**IXFTPKNM**

Der Name des für die Tabelle definierten Primärschlüssels (sofern vorhanden). Der Name wird in Form einer auf auf Null endenden Zeichenfolge gespeichert.

**IXFTDSPC**

Dieses Feld ist zur zukünftigen Verwendung reserviert.

**IXFTISPC**

Dieses Feld ist zur zukünftigen Verwendung reserviert.

**IXFTLSPC**

Dieses Feld ist zur zukünftigen Verwendung reserviert.

**SPALTENDESKRIPTORSATZ**

FELDDNAME	LÄNGE	ART	KOMMENTARE
-----	-----	-----	-----
IXFCRECL	006 BYTE	CHARACTER	Satzlänge
IXFTRECT	001 BYTE	CHARACTER	Satzart = 'C'
IXFCNAML	003 BYTE	CHARACTER	Länge des Spaltennamens
IXFCNAME	256 BYTE	CHARACTER	Spaltenname
IXFCNULL	001 BYTE	CHARACTER	Spalte läßt Nullen zu
IXFCDEF	001 BYTE	CHARACTER	Spalte hat Standardwerte
IXFCSLCT	001 BYTE	CHARACTER	Spaltenauswahlmarkierung
IXFCKPOS	002 BYTE	CHARACTER	Position im Primärschlüssel
IXFCCLAS	001 BYTE	CHARACTER	Datenklasse

## PC-Version des IXF-Dateiformats

IXFCTYPE	003	BYTE	CHARACTER	Datentyp
IXFCSBCP	005	BYTE	CHARACTER	Einzelbyte-Codepage
IXFCDBC	005	BYTE	CHARACTER	Doppelbyte-Codepage
IXFCLENG	005	BYTE	CHARACTER	Spaltendatenlänge
IXFCDRID	003	BYTE	CHARACTER	Satzkennung 'D'
IXFCPOSN	006	BYTE	CHARACTER	Spaltenposition
IXFCDESC	030	BYTE	CHARACTER	Spaltenbeschreibung
IXFCLOBL	020	BYTE	CHARACTER	Länge der LOB-Spalte
IXFCUDTL	003	BYTE	CHARACTER	Länge des UDT-Namens
IXFCUDTN	256	BYTE	CHARACTER	UDT-Name
IXFCDEFL	003	BYTE	CHARACTER	Länge des Standardwerts
IXFCDEFV	254	BYTE	CHARACTER	Standardwert
IXFCDLPR	010	BYTE	CHARACTER	DATALINK-Merkmale
IXFCREF	001	BYTE	CHARACTER	Verweisart
IXFCNDIM	002	BYTE	CHARACTER	Anzahl Dimensionen
IXFCDSIZ	variabel		CHARACTER	Größe jeder Dimension

Die folgenden Felder sind in Spaltendeskiptorsätzen enthalten:

### **IXFCRECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der C-Datensatz muß lang genug sein für alle darin definierten Felder.

### **IXFCRECT**

Die IXF-Satzart, die für diesen Datensatz auf C gesetzt ist.

### **IXFCNAML**

Die Länge des Spaltennamens im IXFTNAME-Feld in Byte.

### **IXFCNAME**

Der Name der Spalte.

### **IXFCNULL**

Gibt an, ob Nullen in dieser Spalte zulässig sind. Gültige Werte sind Y oder N.

### **IXFCDEF**

Gibt an, ob für dieses Feld ein Standardwert definiert ist. Gültige Werte sind Y oder N.

### **IXFCSLCT**

Ein veraltetes Feld, das die Auswahl einer Untermenge von Spalten in der Tabelle ermöglichen sollte. Programme, die PC/IXF-Dateien schreiben, müssen in diesem Feld stets Y speichern. Programme, die PC/IXF-Dateien lesen, müssen dieses Feld ignorieren.

### **IXFCKPOS**

Die Position der Spalte als Bestandteil des Primärschlüssels. Gültige Werte sind 01 bis 16 bzw. N, falls die Spalte nicht Bestandteil des Primärschlüssels ist.



### **IXFCLAS**

Die Klasse der Datentypen, die im IXFCTYPE-Feld verwendet werden sollen. Der Datenbankmanager unterstützt nur relationale Typen (R).

### **IXFCTYPE**

Der Datentyp der Spalte. Weitere Informationen zu Datentypen finden Sie in „PC/IXF-Datentypen“ auf Seite 277.

### **IXFCSBCP**

Enthält eine Einzelbytezeichendarstellung einer SBCS-CPGID. Dieses Feld legt die CPGID für Einzelbytezeichendaten fest, die im IXFDCOLS-Feld der D-Datensätze für diese Spalte vorkommen.

Die Semantik dieses Feldes variiert mit dem Datentyp für die Spalte (im IXFCTYPE-Feld angegeben).

- Für eine Zeichenfolgespalte sollte dieses Feld normalerweise einen Wert ungleich Null enthalten, der mit dem Wert des IXFHSSBCP-Feldes im H-Datensatz übereinstimmt. Andere Werte sind jedoch erlaubt. Wenn dieser Wert Null ist, wird angenommen, daß die Spalte Bitfolgendaten enthält.
- Für eine numerische Spalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf Null gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Datums- oder Uhrzeitspalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf den Wert des Feldes IXFHSSBCP gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Grafikspalte muß dieses Feld Null sein.

Siehe auch Tabelle 12 auf Seite 284.

### **IXFCDBCP**

Enthält eine Einzelbytezeichendarstellung einer DBCS-CPGID. Dieses Feld legt die CPGID für Doppelbytezeichendaten fest, die im IXFDCOLS-Feld der D-Datensätze für diese Spalte vorkommen.

Die Semantik dieses Feldes variiert mit dem Datentyp für die Spalte (im IXFCTYPE-Feld angegeben).

- Für eine Zeichenfolgespalte muß dieses Feld Null sein oder einen Wert enthalten, der mit dem Wert des IXFHDBCP-Feldes im H-Datensatz übereinstimmt. Andere Werte sind jedoch erlaubt. Wenn der Wert im IXFCSBCP-Feld Null ist, muß der Wert in diesem Feld Null sein.
- Für eine numerische Spalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf Null gesetzt und vom Dienstprogramm IMPORT ignoriert.

## PC-Version des IXF-Dateiformats

- Für eine Datums- oder Uhrzeitspalte hat dieses Feld keine Bedeutung. Es wird vom Dienstprogramm EXPORT auf Null gesetzt und vom Dienstprogramm IMPORT ignoriert.
- Für eine Grafikspalte muß dieses Feld einen Wert haben, der mit dem Wert des IXFHDBCP-Feldes identisch ist.

Siehe auch Tabelle 12 auf Seite 284.

### **IXFCLENG**

Bietet Informationen zur Größe der beschriebenen Spalte. Für einige Datentypen wird dieses Feld nicht verwendet und muß Leerzeichen enthalten. Für andere Datentypen enthält dieses Feld die rechtsbündige Zeichendarstellung einer ganzen Zahl, die die Spaltenlänge festlegt. Für andere Datentypen ist dieses Feld in zwei Teilfenster unterteilt: 3 Byte für die Genauigkeit und 2 Byte für die Anzahl der Kommastellen. Beide Teilfelder sind rechtsbündige Zeichendarstellungen von ganzen Zahlen.

### **IXFCDRID**

Die D-Satzkennung. Dieses Feld enthält die rechtsbündige Zeichendarstellung eines ganzzahligen Wertes. Es ist möglich, mehrere D-Datensätze für die einzelnen Datenzeilen in der PC/IXF-Datei zu verwenden. Dieses Feld gibt an, welcher D-Datensatz (einer von mehreren D-Datensätzen, die eine Datenzeile bilden) die Daten für die Spalte enthält. Der Wert Eins (z. B. 001) gibt an, daß sich die Daten für eine Spalte im ersten D-Datensatz in einer Datenzeile befinden. Der erste C-Datensatz muß den IXFCDRID-Wert Eins haben. Alle nachfolgenden C-Datensätze müssen einen IXFCDRID-Wert haben, der mit dem Wert des vorherigen C-Datensatzes übereinstimmt oder um Eins größer ist.

### **IXFCPOSN**

Der Wert dieses Feldes wird verwendet, um die Daten für die Spalte innerhalb eines der D-Datensätze zu lokalisieren, die eine Zeile der Tabellendaten darstellen. Es handelt sich um den Anfangspunkt der Daten für diese Spalte innerhalb des IXFDCOLS-Feldes des D-Datensatzes. Wenn die Spalte Nullwerte enthalten darf, zeigt IXFCPOSN auf den Nullanzeiger, andernfalls auf die Daten selbst. Wenn eine Spalte Daten mit variabler Länge enthält, beginnen die Daten selbst mit dem aktuellen Längenzeiger. Der IXFCPOSN-Wert für das erste Byte im IXFDCOLS-Feld des D-Datensatzes ist Eins (nicht Null). Wenn sich eine Spalte in einem neuen D-Datensatz befindet, muß der Wert von IXFCPOSN Eins sein. Andernfalls müssen die IXFCPOSN-Werte von Spalte zu Spalte so zunehmen, daß sich die Datenwerte nicht überschneiden.

### **IXFCDESC**

Beschreibende Informationen zu der Spalte. Dieses Feld dient nur zur Information. Wenn ein Programm, das in eine Datei schreibt, keine Daten in dieses Feld schreiben muß, wird es vorzugsweise mit Leerzeichen gefüllt. Programme, die eine Datei lesen, können dieses Feld drucken oder anzeigen oder es in einem Informationsfeld speichern. Vom Inhalt dieses Feldes dürfen jedoch keine Berechnungen abhängen.

### **IXFCLOBL**

Die Länge des in dieser Spalte definierten großen Objekts (LOB) in Byte. Handelt es sich nicht um eine LOB-Spalte, lautet der Wert in diesem Feld 000.

### **IXFCUDTL**

Die Länge des Namens des benutzerdefinierten Typs (UDT) im IXFCUDTN-Feld in Byte. Handelt es sich nicht um eine Spalte des Typs UDT, hat dieses Feld den Wert 000.

### **IXFCUDTN**

Der Name des benutzerdefinierten Typs, der als Datentyp für diese Spalte verwendet wird.

### **IXFCDEFL**

Die Länge des Standardwerts im IXFCDEFV-Feld in Byte. Wenn diese Spalte keinen Standardwert hat, lautet der Wert in diesem Feld 000.

### **IXFCDEFV**

Gibt den Standardwert für diese Spalte an, sofern ein solcher definiert wurde.

### **IXFCDLPR**

Handelt es sich bei der Spalte um eine DATALINK-Spalte, beschreibt dieses Feld die folgenden Merkmale:

- Das erste Zeichen steht für die Verbindungsart. Es hat den Wert U.
- Das zweite Zeichen stellt den Verbindungssteuerungstyp dar. Gültige Werte sind N (= keine Steuerung) und F (= Dateisteuerung).
- Das dritte Zeichen stellt die Integritätsstufe dar und hat den Wert A. Hiermit wird angegeben, daß der Datenbankmanager alle DATALINK-Werte steuert.
- Das vierte Zeichen steht für die Leseberechtigung. Gültige Werte sind D (= Berechtigung wird durch die Datenbank festgestellt) und F (= Berechtigung wird durch Dateisystem festgestellt).
- Das fünfte Zeichen gibt die Schreibberechtigung an. Gültige Werte sind B (= geblockter Zugriff) und F (Berechtigung wird durch Dateisystem festgestellt).

## PC-Version des IXF-Dateiformats

- Das sechste Zeichen stellt die Wiederherstellungsoptionen dar. Gültige Werte sind Y (= DB2 unterstützt Zeitpunktwiederherstellung von Dateien, auf die in dieser Spalte verwiesen wird) und N (= keine Unterstützung).
- Das siebte Zeichen stellt die Aktion dar, die ausgeführt werden soll, wenn die Verbindung der Datendatei aufgehoben wird. Gültige Werte sind R (= Wiederherstellen) und D (= Datei löschen).

### IXFCREF

Wenn die Spalte Teil einer Hierarchie ist, gibt dieses Feld an, ob es sich bei der Spalte um eine Datenspalte (D) oder um eine Referenzspalte (R) handelt.

### IXFCNDIM

Die Anzahl der Dimensionen in der Spalte. Matrizen werden in dieser Version von PC/IXF nicht unterstützt. Dieses Feld muß daher eine Zeichendarstellung eines ganzzahligen Nullwerts enthalten.

### IXFCDSIZ

Die Größe bzw. der Bereich jeder Dimension. Die Länge dieses Feldes beträgt 5 Byte pro Dimension. Da Matrizen nicht unterstützt werden (d. h. die Anzahl der Dimensionen Null sein muß), hat dieses Feld die Länge Null und ist eigentlich gar nicht vorhanden.

### DATENSATZ

FELDNAME	LÄNGE	ART	KOMMENTARE
IXFDRECL	06 BYTE	CHARACTER	Satzlänge
IXFDRECT	01 BYTE	CHARACTER	Satzart = 'D'
IXFDRID	03 BYTE	CHARACTER	Satzkennung 'D'
IXDFIL1	04 BYTE	CHARACTER	reserviert
IXFDCOLS	variabel	variabel	Spaltendaten

Die folgenden Felder sind in Datensätzen enthalten:

### IXFDRECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Jeder D-Datensatz muß lang genug für alle signifikanten Daten für das aktuelle Vorkommen der letzten im Datensatz gespeicherten Datenspalte sein.

### IXFDRECT

Die IXF-Satzart, die für diesen Datensatz auf D gesetzt ist, um anzugeben, daß der Datensatz Datenwerte für die Tabelle enthält.

**IXFDRID**

Die Satzkennung, die einen bestimmten D-Datensatz in der Folge von mehreren D-Datensätzen angibt, die eine Datenzeile bilden. Für den ersten D-Datensatz in einer Datenzeile enthält dieses Feld den Wert Eins. Für den zweiten D-Datensatz in einer Datenzeile enthält dieses Feld den Wert Zwei usw. In jeder Datenzeile müssen alle D-Satzkennungen, die in den C-Datensätzen gesendet werden, auch vorhanden sein.

**IXFDFIL1**

Zusatzfeld, das vier Leerzeichen enthält. Dient als Entsprechung zu reservierten Feldern in Host-IXF-Dateien und bietet Platz für ein mögliches DBCS-Startzeichen.

**IXFDCOLS**

Der Bereich für Spaltendaten. Der Datenbereich eines Datensatzes (D-Satzes) besteht aus mindestens einem Spalteneintrag. Es gibt einen Spalteneintrag für jeden Spaltendeskriptorsatz, der die gleiche D-Satzkennung hat wie der D-Satz. Im D-Satz wird die Anfangsposition der Spalteneinträge durch den IXFCPOSN-Wert in C-Sätzen angegeben.

Das Format der Spalteneingabedaten hängt davon ab, ob die Spalte Nullwerte enthalten darf:

- Wenn die Spalte Nullwerte enthalten darf (das IXFCNULL-Feld auf Y gesetzt ist), enthalten die Spalteneingabedaten einen Nullanzeiger. Wenn die Spalte nicht Null ist, folgen auf den Anzeiger datentypspezifische Informationen, einschließlich des eigentlichen Datenbankwertes. Der Nullanzeiger ist ein 2 Byte großer Wert, der für nicht Null auf x'0000' und für Null auf x'FFFF' gesetzt wird.
- Wenn die Spalte keine Nullwerte enthalten darf, enthalten die Spalteneingabedaten nur datentypspezifische Informationen, einschließlich des tatsächlichen Datenbankwertes.

Für Datentypen mit variabler Länge schließen die datentypspezifischen Informationen einen aktuellen Längenanzeiger ein. Die aktuellen Längenanzeiger sind 2 Byte große ganze Zahlen in einem Format, das vom IXFTMFRM-Feld festgelegt wird.

## PC-Version des IXF-Dateiformats

Die Länge des Datenbereichs eines D-Datensatzes darf 32 771 Byte nicht überschreiten.

### ANWENDUNGSDATENSATZ

FELDNAM	LÄNGE	ART	KOMMENTARE
IXFARECL	06 BYTE	CHARACTER	Satzlänge
IXFARECT	01 BYTE	CHARACTER	Satzart = 'A'
IXFAAPPID	12 BYTE	CHARACTER	Anwendungskennzeichen
IXFADATA	variabel	variabel	anwendungsspezifische Daten

Die folgenden Felder sind in Anwendungsdatensätzen enthalten:

#### **IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAAPPID-Feld.

#### **IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

#### **IXFAAPPID**

Das Anwendungskennzeichen, das die Anwendung angibt, die den A-Datensatz erstellt. Vom Datenbankmanager erstellte PC/IXF-Dateien können A-Datensätze enthalten, in denen die ersten sechs Zeichen dieses Feldes auf eine Konstante gesetzt sind, die den Datenbankmanager angibt, während die letzten 6 Zeichen das Release oder die Version des Datenbankmanagers oder einer anderen Anwendung angeben, der bzw. die den A-Datensatz schreibt.

#### **IXFADATA**

Dieses Feld enthält anwendungsabhängige Ergänzungsdaten, deren Format und Inhalt nur dem Programm bekannt ist, das den A-Datensatz erstellt, sowie anderen Anwendungen, die den A-Datensatz wahrscheinlich verarbeiten.

DB2-INDEXSATZ

FELDNAME	LÄNGE	ART	KOMMENTARE
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satzart = 'A'
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAITYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'I'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFANDXL	002 BYTE	SHORT INT	Länge des Indexnamens
IXFANDXN	256 BYTE	CHARACTER	Name des Indexes
IXFANCL	002 BYTE	SHORT INT	Länge des Namens des Indexerstellers
IXFANCN	256 BYTE	CHARACTER	Name des Indexerstellers
IXFATABL	002 BYTE	SHORT INT	Länge des Tabellennamens
IXFATABN	256 BYTE	CHARACTER	Tabellenname
IXFATCL	002 BYTE	SHORT INT	Länge des Namens des Tabellenerstellers
IXFATCN	256 BYTE	CHARACTER	Name des Tabellenerstellers
IXFAUNIQ	001 BYTE	CHARACTER	Eindeutige Regel
IXFACNT	002 BYTE	CHARACTER	Spaltenanzahl
IXFAREVS	001 BYTE	CHARACTER	Markierung für Rückwärtsdurchsuchen zulassen
IXFAPCTF	002 BYTE	CHARACTER	Wert für PCTFREE
IXFAPCTU	002 BYTE	CHARACTER	Wert für MINPCTUSED (Prozentsatz des auf Indexseiten freizuhaltenen Minimums an verwendetem Speicherplatz)
IXFAEXTI	001 BYTE	CHARACTER	reserviert
IXFACNML	002 BYTE	SHORT INT	Länge des Namens der Spalten
IXFACOLN	variabel	CHARACTER	Name der Spalten im Index

Für jeden benutzerdefinierten Index wird 1 Datensatz diesen Typs angegeben. Die Position dieses Datensatzes befindet sich nach allen C-Datensätzen für die Tabelle. DB2-Indexsätze enthalten die folgenden Felder:

**IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAPPID-Feld.

**IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

## PC-Version des IXF-Dateiformats

### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### **IXFAITYP**

Gibt an, daß dies die Subart „I“ von DB2-Anwendungssätzen ist.

### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muß denselben Wert wie das IXFHDATE-Feld aufweisen.

### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muß denselben Wert wie das IXFHTIME-Feld aufweisen.

### **IXFANDXL**

Die Länge des Indexnamens im IXFANDXN-Feld in Byte.

### **IXFANDXN**

Der Name des Indexes.

### **IXFANCL**

Die Länge des Namens des Indexerstellers im IXFANCN-Feld in Byte.

### **IXFANCN**

Der Name des Indexerstellers.

### **IXFATABL**

Die Länge des Tabellennamens im IXFATABN-Feld in Byte.

### **IXFATABN**

Der Name der Tabelle.

### **IXFATCL**

Die Länge des Namens des Tabellenerstellers im IXFATCN-Feld in Byte.

### **IXFATCN**

Der Name des Tabellenerstellers.

### **IXFAUNIQ**

Gibt die Indexart an. Gültige Werte sind P (= Primärschlüssel), U (= eindeutiger Index) und D (= nicht eindeutiger Index).

### **IXFACCNT**

Gibt die Anzahl der Spalten in der Indexdefinition an.



**IXFAREVS**

Gibt an, ob das Rückwärtsdurchsuchen dieses Indexes zulässig ist. Gültige Werte sind Y (= zulässig) und N (= nicht zulässig).

**IXFAPCTF**

Gibt den Prozentsatz der freizuhaltenen Indexseiten an. Gültige Werte liegen zwischen -1 und 99. Wenn der Wert -1 oder Null angegeben wird, wird der Systemstandardwert verwendet.

**IXFAPCTU**

Gibt den Mindestprozentsatz für den Speicherplatz an, der auf Indexseiten freigehalten werden muß, bevor zwei Indexseiten zusammengeführt werden können. Gültige Werte liegen zwischen 00 und 99.

**IXFAEXTI**

Zur zukünftigen Verwendung reserviert.

**IXFACNML**

Die Länge der Spaltennamen im IXFACOLN-Feld in Byte.

**IXFACOLN**

Die Namen der Spalten, die Bestandteil dieses Indexes sind. Gültige Werte haben das Format *+name-name...* Hierbei gibt + eine aufsteigende Sortierung der Spalte und - eine absteigende Sortierung der Spalte an.

DB2-HIERARCHIESATZ

FELDDNAME	LÄNGE	ART	KOMMENTARE
-----	-----	-----	-----
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satzart = 'A'
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAXTYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'X'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFAYCNT	010 BYTE	CHARACTER	'Y'-Satzzählung für diese Hierarchie
IXFAYSTR	010 BYTE	CHARACTER	Anfangsspalte dieser Hierarchie

## PC-Version des IXF-Dateiformats

Zur Beschreibung einer Hierarchie wird 1 Datensatz diesen Typs verwendet. Alle Sätze für untergeordnete Tabellen (siehe unten) müssen unmittelbar auf den Hierarchiesatz folgen. Die Hierarchiesätze müssen auf die C-Datensätze für die Tabelle folgen. DB2-Hierarchiesätze enthalten die folgenden Felder:

### **IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAPPID-Feld.

### **IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### **IXFAXTYP**

Gibt an, daß dies die Subart „X“ von DB2-Anwendungssätzen ist.

### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjmmmtt*. Dieses Feld muß denselben Wert wie das IXFHDATE-Feld aufweisen.

### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muß denselben Wert wie das IXFHTIME-Feld aufweisen.

### **IXFAYCNT**

Gibt die Anzahl der Sätze für untergeordnete Tabellen an, die nach diesem Hierarchiesatz erwartet werden.

**IXFAYSTR**

Gibt den Index der Sätze für untergeordnete Tabellen am Anfang der exportierten Daten an. Wurde der Export einer Hierarchie von einer untergeordneten Tabelle aus gestartet, die nicht die Stammtabelle ist, werden alle übergeordneten Tabellen dieser untergeordneten Tabelle exportiert. Die Position dieser untergeordneten Tabelle innerhalb der IXF-Datei wird ebenfalls in diesem Feld gespeichert. Der erste X-Datensatz stellt die Spalte mit einem Index 0 dar.

DB2-SATZ FÜR UNTERGEORDNETE TABELLE

FELDNAME	LÄNGE	ART	KOMMENTARE
-----	-----	-----	-----
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satzart = 'A'
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAYTYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'Y'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFASCHL	003 BYTE	CHARACTER	Länge des Typschemennamens
IXFASCHN	256 BYTE	CHARACTER	Typschemenname
IXFATYPL	003 BYTE	CHARACTER	Länge des Typnamens
IXFATYPN	256 BYTE	CHARACTER	Typname
IXFATABL	003 BYTE	CHARACTER	Länge des Tabellennamens
IXFATABN	256 BYTE	CHARACTER	Tabellenname
IXFAPNDX	010 BYTE	CHARACTER	Index der untergeordnete Tabelle für übergeordnete Tabelle
IXFASNDX	005 BYTE	CHARACTER	Beginn Spaltenindex der aktuellen Tabelle
IXFAENDX	005 BYTE	CHARACTER	Ende Spaltenindex der aktuellen Tabelle

Zur Beschreibung einer untergeordneten Tabelle als Teil einer Hierarchie wird 1 Datensatz diesen Typs verwendet. Alle Sätze für untergeordnete Tabellen, die zu einer Hierarchie gehören, müssen zusammen gespeichert werden und unmittelbar auf den entsprechenden Hierarchiesatz folgen. Eine untergeordnete Tabelle besteht aus einer oder mehreren Spalten. Jede Spalte wird in einem Spaltensatz beschrieben. Jede Spalte in einer untergeordneten Tabelle muß durch eine Reihe von aufeinanderfolgenden C-Datensätzen beschrieben werden.

## PC-Version des IXF-Dateiformats

DB2-Sätze für untergeordnete Tabellen enthalten die folgenden Felder:

### **IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAPPID-Feld.

### **IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### **IXFAYTYP**

Gibt an, daß dies die Subart „Y“ von DB2-Anwendungssätzen ist.

### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muß denselben Wert wie das IXFHDATE-Feld aufweisen.

### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muß denselben Wert wie das IXFHTIME-Feld aufweisen.

### **IXFASCHL**

Die Länge des Schemennamens für die untergeordnete Tabelle im IXFASCHN-Feld in Byte.

### **IXFASCHN**

Der Name des Schemas für die untergeordnete Tabelle.

### **IXFATYPL**

Die Länge des Namens der untergeordneten Tabelle im IXFATYPN-Feld in Byte.

### **IXFATYPN**

Der Name der untergeordneten Tabelle.

### **IXFATABL**

Die Länge des Tabellennamens im IXFATABN-Feld in Byte.

### **IXFATABN**

Der Name der Tabelle.

**IXFAPNDX**

Der Eintragsindex für die untergeordnete Tabelle der übergeordneten Tabelle. Bildet diese untergeordnete Tabelle den Stamm einer Hierarchie, enthält dieses Feld den Wert -1.

**IXFASNDX**

Startindex der Spaltensätze, aus denen diese untergeordnete Tabelle besteht.

**IXFAENDX**

Endindex der Spaltensätze, aus denen diese untergeordnete Tabelle besteht.

DB2-FORTSETZUNGSSATZ

FELDDNAME	LÄNGE	ART	KOMMENTARE
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satzart = 'A'
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFACTYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'C'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit
IXFALAST	002 BYTE	SHORT INT	Datenträgernummer der ersten Diskette
IXFATHIS	002 BYTE	SHORT INT	Datenträgernummer dieser Diskette
IXFANEXT	002 BYTE	SHORT INT	Datenträgernummer der nächsten Diskette

Dieser Datensatz befindet sich am Ende jeder Datei, die Bestandteil einer auf mehrere Datenträger verteilten IXF-Datei ist, sofern es sich bei dem aktuellen Datenträger nicht um den letzten Datenträger handelt. Außerdem ist dieser Datensatz am Beginn jeder Datei zu finden, die Bestandteil einer auf mehrere Datenträger verteilten IXF-Datei ist, sofern es sich bei dem aktuellen Datenträger nicht um den ersten Datenträger handelt. Der Zweck dieses Datensatzes ist die Einhaltung der Dateireihenfolge. DB2-Fortsetzungssätze enthalten die folgenden Felder:

**IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAPPID-Feld.

**IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

## PC-Version des IXF-Dateiformats

### IXFAPPID

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### IXFACTYP

Gibt an, daß dies die Subart „C“ von DB2-Anwendungssätzen ist.

### IXFADATE

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muß denselben Wert wie das IXFHDATE-Feld aufweisen.

### IXFATIME

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muß denselben Wert wie das IXFHTIME-Feld aufweisen.

### IXFALAST

Dieses Feld ist ein Feld für Binärdaten im Little-Endian-Format. Der Wert sollte um 1 kleiner als der Wert im IXFATHIS-Feld sein.

### IXFATHIS

Dieses Feld ist ein Feld für Binärdaten im Little-Endian-Format. Der Wert in diesem Feld sollte auf Fortsetzungsdatenträgern ebenfalls fortlaufend sein. Der erste Datenträger hat den Wert 1.

### IXFANEXT

Dieses Feld ist ein Feld für Binärdaten im Little-Endian-Format. Der Wert sollte um 1 größer als der Wert im IXFATHIS-Feld sein, es sei denn, der Datensatz befindet sich am Dateianfang. In diesem Fall sollte der Wert 0 lauten.

### DB2-BEENDIGUNGSSATZ

FELDDNAME	LÄNGE	ART	KOMMENTARE
IXFARECL	006 BYTE	CHARACTER	Satzlänge
IXFARECT	001 BYTE	CHARACTER	Satzart = 'A'
IXFAPPID	012 BYTE	CHARACTER	Anwendungskennzeichen = 'DB2 02.00'
IXFAETYP	001 BYTE	CHARACTER	Anwendungsspezifischer Datentyp = 'E'
IXFADATE	008 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Daten
IXFATIME	006 BYTE	CHARACTER	aus dem H-Datensatz geschriebene Zeit

Dieser Datensatz ist das Dateiendezeichen, das sich am Ende einer IXF-Datei befindet. DB2-Beendigungssätze enthalten die folgenden Felder:

### IXFARECL

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAPPID-Feld.

**IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

**IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

**IXFAETYP**

Gibt an, daß dies die Subart „E“ von DB2-Anwendungssätzen ist.

**IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muß denselben Wert wie das IXFHDATE-Feld aufweisen.

**IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muß denselben Wert wie das IXFHTIME-Feld aufweisen.

DB2-IDENTITÄTSSATZ

FELDNAME	LÄNGE	ART	KOMMENTARE
-----	-----	-----	-----
IXFARECL	06 BYTE	CHARACTER	Satzlänge
IXFARECT	01 BYTE	CHARACTER	Satzart = 'A'
IXFAPPID	12 BYTE	CHARACTER	Anwendungskennzeichen
IXFATYPE	01 BYTE	CHARACTER	anwendungsspezifische Satzart = 'S'
IXFADATE	08 BYTE	CHARACTER	Erstellungsdatum des Anwendungssatzes
IXFATIME	06 BYTE	CHARACTER	Erstellungszeit des Anwendungssatzes
IXFACOLN	06 BYTE	CHARACTER	Spaltennummer der Identitätsspalte
IXFAITYP	01 BYTE	CHARACTER	GENERATED ALWAYS ('Y' oder 'N')
IXFASTRT	33 BYTE	CHARACTER	START AT-Wert für Identitätsspalte
IXFAINCR	33 BYTE	CHARACTER	INCREMENT BY-Wert für Identitätsspalte
IXFACACH	10 BYTE	CHARACTER	CACHE-Wert für Identitätsspalte

DB2-Identitätssätze enthalten die folgenden Felder:

**IXFARECL**

Der Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des Teils des PC/IXF-Datensatzes in Byte angibt, der auf den Satzlängenanzeiger folgt, d. h. die gesamte Satzgröße minus 6 Byte. Der A-Datensatz muß lang genug sein für das gesamte IXFAPPID-Feld.

**IXFARECT**

Die IXF-Satzart, die für diesen Datensatz auf A gesetzt ist, um anzugeben, daß es sich um einen Anwendungsdatensatz handelt. Diese

## PC-Version des IXF-Dateiformats

Datensätze werden von Programmen ignoriert, die keine Kenntnis des Inhalts und Formats der Daten haben, die vom Anwendungskennzeichen impliziert werden.

### **IXFAPPID**

Das Anwendungskennzeichen, das DB2 als die Anwendung angibt, die diesen A-Datensatz erstellt.

### **IXFATYPE**

Gibt die anwendungsspezifische Satzart an. Dieses Feld sollte immer den Wert „S“ haben.

### **IXFADATE**

Das Datum, an dem die Datei geschrieben wurde, im Format *jjjjmmtt*. Dieses Feld muß denselben Wert wie das IXFHDATE-Feld aufweisen.

### **IXFATIME**

Die Uhrzeit, zu der die Datei geschrieben wurde, im Format *ssmmss*. Dieses Feld muß denselben Wert wie das IXFHTIME-Feld aufweisen.

### **IXFACOLN**

Die Spaltennummer der Identitätsspalte in der Tabelle.

### **IXFAITYP**

Der Typ der Identitätsspalte. Der Wert „Y“ gibt an, daß die Identitätsspalte als GENERATED ALWAYS definiert ist, d. h. in jedem Fall generiert wird. Alle anderen Werte werden so interpretiert, daß die Spalte als GENERATED BY DEFAULT definiert ist.

### **IXFASTRT**

Der START AT-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde.

### **IXFAINCR**

Der INCREMENT BY-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde.

### **IXFACACH**

Der CACHE-Wert für die Identitätsspalte, der zum Zeitpunkt der Tabellenerstellung an die Anweisung CREATE TABLE übergeben wurde. Der Wert „1“ entspricht der Option NO CACHE.



## PC/IXF-Datentypen

Tabelle 11. PC/IXF-Datentypen

Name	IXFCTYPE-Wert	Beschreibung
BIGINT	492	Eine 8 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Stellt eine ganze Zahl zwischen -9 223 372 036 854 775 808 und 9 223 372 036 854 775 807 dar. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. IXFCLENG wird nicht verwendet und muß Leerzeichen enthalten.
BLOB, CLOB	404, 408	<p>Eine Zeichenfolge variabler Länge. Die maximale Länge der Zeichenfolge ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten, und sie darf 32 767 Byte nicht überschreiten. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 4 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Byte angibt. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist.</p> <p>Folgendes gilt nur für BLOBs: Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.</p> <p>Folgendes gilt nur für CLOBs: Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist.</p>

## PC-Version des IXF-Dateiformats

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
BLOB_FILE, CLOB_FILE, DBCLOB_FILE	916, 920, 924	<p>Ein Feld fester Länge, das eine SQLFILE-Struktur enthält, bei der die Felder <i>name_length</i> und <i>name</i> ausgefüllt sind. Die Länge der Struktur ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten und darf 255 Byte nicht überschreiten. Der Dateiname liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann der Dateiname auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht der Dateiname aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.</p> <p>Da die Länge der Struktur in IXFCLENG gespeichert ist, geht die eigentliche Länge des ursprünglichen LOBs verloren. IXF-Dateien mit Spalten des Typs BLOB_FILE, CLOB_FILE oder DBCLOB_FILE dürfen zur erneuten Erstellung des LOB-Feldes nicht verwendet werden, da das LOB-Feld mit der Länge <i>sql_lobfile_len</i> erstellt wird.</p>
CHAR	452	<p>Eine Zeichenfolge mit fester Länge. Die Länge der Zeichenfolge ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten, und sie darf 254 Byte nicht überschreiten. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.</p>

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
DATE	384	<p>Ein Zeitpunkt gemäß gregorianischem Kalender. Jedes Datum ist eine 10 Byte große Zeichenfolge im ISO-Format: <i>jjj-mm-tt</i>. Der Wertebereich für das Jahr liegt zwischen 0001 und 9999. Der Wertebereich für den Monat liegt zwischen 1 und 12. Der Bereich für den Tag ist 01 bis <i>n</i>, wobei <i>n</i> vom Monat abhängt. Dabei gelten die üblichen Regeln für die Tage eines Monats und Schaltjahre. Führende Nullen können bei keinem Teil weggelassen werden. IXFCLENG wird nicht verwendet und muß Leerzeichen enthalten. Gültige Zeichen für DATE sind in allen PC-ASCII-Codepages gleich. Daher sind IXFCSBCP und IXFCDBCP nicht signifikant und müssen Null sein.</p>
DBCLOB	412	<p>Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen in der Zeichenfolge an, die 16 383 nicht überschreiten darf. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 4 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Doppelbytezeichen angibt (d. h. der Wert dieser ganzen Zahl ist die halbe Länge der Zeichenfolge in Byte). Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muß IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.</p>

## PC-Version des IXF-Dateiformats

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
DECIMAL	484	Eine gepackte Dezimalzahl mit der Genauigkeit P (wie durch die ersten drei Byte von IXFLENG im Spaltendeskriptorsatz festgelegt) und der Anzahl S der Kommastellen (wie durch die letzten beiden Byte von IXFLENG festgelegt). Die Länge einer gepackten Dezimalzahl in Byte ist $(P+2)/2$ . Die Genauigkeit muß eine ungerade Zahl zwischen 1 und 31 (jeweils einschließlich) sein. Die gepackte Dezimalzahl liegt im internen Format vor, das mit IXFTMFRM angegeben wird, während die gepackte Dezimalzahl für den PC so definiert ist, daß sie mit der gepackten Dezimalzahl für das IBM System /370 identisch ist. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein.
FLOATING POINT	480	Entweder eine lange (8 Byte) oder kurze (4 Byte) Gleitkommazahl. Dies hängt davon ab, ob IXFLENG auf 8 oder 4 gesetzt ist. Die Daten liegen im internen Maschinenformat vor, wie mit IXFTMFRM angegeben. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. 4 Byte große Gleitkommazahlen werden vom Datenbankmanager nicht unterstützt.
GRAPHIC	468	Eine Zeichenfolge fester Länge aus Doppelbytezeichen. Das IXFLENG-Feld im Spaltendeskriptorsatz gibt die Anzahl von Doppelbytezeichen in der Zeichenfolge an und darf maximal den Wert 127 enthalten. Die eigentliche Länge der Zeichenfolge ist das Doppelte des Werts des IXFLENG-Feldes in Byte. Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muß IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
INTEGER	496	Eine 4 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Stellt eine ganze Zahl zwischen -2 147 483 648 und +2 147 483 647 dar. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. IXFCLENG wird nicht verwendet und muß Leerzeichen enthalten.
LONGVARCHAR	456	Eine Zeichenfolge variabler Länge. Die maximale Länge der Zeichenfolge ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten, und sie darf 32 767 Byte nicht überschreiten. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Byte angibt. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.

## PC-Version des IXF-Dateiformats

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
LONG VARGRAPHIC	472	Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen in der Zeichenfolge an, die 16 383 nicht überschreiten darf. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Doppelbytezeichen angibt (d. h. der Wert dieser ganzen Zahl ist die halbe Länge der Zeichenfolge in Byte). Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muß IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.
SMALLINT	500	Eine 2 Byte große ganze Zahl in dem von IXFTMFRM festgelegten Format. Stellt eine ganze Zahl zwischen -32 768 und +32 767 dar. IXFCSBCP und IXFCDBCP sind nicht signifikant und müssen Null sein. IXFLENG wird nicht verwendet und muß Leerzeichen enthalten.
TIME	388	Ein Zeitpunkt gemäß gregorianischem Kalender. Jede Zeitangabe ist eine 8 Byte große Zeichenfolge im ISO-Format: <i>ss.mm.ss</i> . Der Wertebereich für die Stunde liegt zwischen 0 und 24 und der Wertebereich für Minuten und Sekunden zwischen 0 und 59. Bei Stunde 24 sind die Minuten- und Sekundenangaben gleich Null. Der kleinste Zeitwert ist 00.00.00, der größte 24.00.00. Führende Nullen können bei keinem Teil weggelassen werden. IXFLENG wird nicht verwendet und muß Leerzeichen enthalten. Gültige Zeichen für TIME sind in allen PC-ASCII-Codepages gleich. Daher sind IXFCSBCP und IXFCDBCP nicht signifikant und müssen Null sein.

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
TIMESTAMP	392	Das Datum und die Uhrzeit mit der Genauigkeit von Mikrosekunden. Jede Zeitmarke ist eine Zeichenfolge im Format <i>jjjj-mm-tt-ss.mm.ss.nnnnnn</i> (Jahr Monat Tag Stunde Minuten Sekunden Mikrosekunden). IXFCLENG wird nicht verwendet und muß Leerzeichen enthalten. Gültige Zeichen für TIMESTAMP sind in allen PC-ASCII-Codepages gleich. Daher sind IXFCSBCP und IXFCDBCP nicht signifikant und müssen Null sein.
VARCHAR	448	Eine Zeichenfolge variabler Länge. Die maximale Länge der Zeichenfolge in Byte ist im IXFCLENG-Feld des Spaltendeskriptorsatzes enthalten und darf 254 Byte nicht überschreiten. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Byte angibt. Die Zeichenfolge liegt in der Codepage vor, die in IXFCSBCP angegeben ist. Wenn IXFCDBCP ungleich Null ist, kann die Zeichenfolge auch Doppelbytezeichen in der Codepage enthalten, die in IXFCDBCP angegeben ist. Wenn IXFCSBCP Null ist, besteht die Zeichenfolge aus Bitdaten und darf von keinem Umsetzungsprogramm umgesetzt werden.

## PC-Version des IXF-Dateiformats

Tabelle 11. PC/IXF-Datentypen (Forts.)

Name	IXFCTYPE-Wert	Beschreibung
VARGRAPHIC	464	Eine Zeichenfolge variabler Länge aus Doppelbytezeichen. Das IXFCLENG-Feld im Spaltendeskriptorsatz gibt die maximale Anzahl von Doppelbytezeichen in der Zeichenfolge an, die 127 nicht überschreiten darf. Der Zeichenfolge selbst wird ein aktueller Längenanzeiger vorangestellt, bei dem es sich um eine 2 Byte große ganze Zahl handelt, die die Länge der Zeichenfolge in Doppelbytezeichen angibt (d. h. der Wert dieser ganzen Zahl ist die halbe Länge der Zeichenfolge in Byte). Die Zeichenfolge liegt in der DBCS-Codepage vor, wie mit IXFCDBCP im C-Datensatz angegeben. Da die Zeichenfolge nur Doppelbytezeichen enthält, muß IXFCSBCP Null sein. Es gibt keine DBCS-Startzeichen oder DBCS-Endezeichen.

Nicht alle Kombinationen von IXFCSBCP- und IXFCDBCP-Werten für die PC/IXF-Zeichen- oder -Grafikspalten sind gültig. Eine PC/IXF-Zeichen- oder -Grafikspalte mit einer ungültigen Kombination aus IXFCSBCP und IXFCDBCP ist ein ungültiger Datentyp.

Tabelle 12. Gültige PC/IXF-Datentypen

PC/IXF-Datentyp	Gültige Paare (IXFCSBCP,IXFCDBCP)	Ungültige Paare (IXFCSBCP,IXFCDBCP)
CHAR, VARCHAR oder LONG VARCHAR	(0,0), (x,0) oder (x,y)	(0,y)
BLOB	(0,0)	(x,0), (0,y) oder (x,y)
CLOB	(x,0), (x,y)	(0,0), (0,y)
GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC oder DBCLOB	(0,y)	(0,0), (x,0) oder (x,y)
<b>Anmerkung:</b> x und y sind nicht 0.		



## Beschreibung von PC/IXF-Datentypen

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
BIGINT	Eine BIGINT-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -9 223 372 036 854 775 808 bis 9 223 372 036 854 775 807 liegen.
BLOB	Eine PC/IXF-Spalte des Typs BLOB wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spalten-deskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs CHAR, VARCHAR, LONG VARCHAR, BLOB oder BLOB_FILE ist in folgenden Fällen zulässig: <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte. Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC BLOB ist ebenfalls zulässig. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.</li> </ul>

## PC-Version des IXF-Dateiformats

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
CHAR	Eine PC/IXF-Spalte des Typs CHAR wird erstellt. Die Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den PC/IXF-Spaltendeskriptorsatz kopiert.	<p>Eine PC/IXF-Spalte des Typs CHAR, VARCHAR oder LONG VARCHAR ist in folgenden Fällen zulässig:</p> <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte.</li> </ul> <p>Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist ebenfalls zulässig, wenn die Datenbankspalte mit FOR BIT DATA markiert ist. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge in jedem Fall mit der Länge der Datenbankspalte kompatibel sein. Die Daten werden bei Bedarf auf der rechten Seite mit Einzelbyteleerzeichen (x'20') aufgefüllt. Siehe auch „Option FORCEIN“ auf Seite 298.</p>
CLOB	Eine PC/IXF-Spalte des Typs CLOB wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs CHAR, VARCHAR, LONG VARCHAR, CLOB oder CLOB_FILE ist zulässig, wenn der Einzelbyte-Codepage-Wert der PC/IXF-Spalte mit der SBCS-CPGID der Datenbankspalte übereinstimmt und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte Null oder der DBCS-CPGID der Datenbankspalte entspricht. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
DATE	Eine DATE-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	Eine PC/IXF-Spalte des Typs DATE ist die übliche Eingabe. Das Dienstprogramm IMPORT versucht auch, Spalten in einem beliebigen Zeichentyp anzunehmen, sofern sie keine inkompatiblen Längen aufweisen. Die Zeichenspalte in der PC/IXF-Datei muß Datumswerte in einem Format enthalten, das mit dem Landescode der Zieldatenbank konsistent ist.
DBCLOB	Eine PC/IXF-Spalte des Typs DBCLOB wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spalten-deskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC, LONG VARGRAPHIC, DBCLOB oder DBCLOB_FILE ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.
DECIMAL	Eine DECIMAL-Spalte, die mit der Datenbankspalte identisch ist, wird erstellt. Die Genauigkeit und die Anzahl der Kommastellen werden im Spaltendeskriptorsatz gespeichert.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich der DECIMAL-Spalte liegen, in die sie importiert werden.
FLOAT	Eine FLOAT-Spalte wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Alle Werte liegen innerhalb des Bereichs.

## PC-Version des IXF-Dateiformats

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
GRAPHIC (nur DBCS)	Eine PC/IXF-Spalte des Typs GRAPHIC wird erstellt. Die Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spalten-deskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge mit der Länge der Datenbankspalte kompatibel sein. Die Daten werden bei Bedarf auf der rechten Seite mit Doppelbyteleerzeichen (x'8140') aufgefüllt. Siehe auch „Option FORCEIN“ auf Seite 298.
INTEGER	Eine Spalte des Typs INTEGER wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -2 147 483 648 bis 2 147 483 647 liegen.
LONG VARCHAR	Eine PC/IXF-Spalte des Typs LONG VARCHAR wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spalten-deskriptorsatz kopiert.	<p>Eine PC/IXF-Spalte des Typs CHAR, VARCHAR oder LONG VARCHAR ist in folgenden Fällen zulässig:</p> <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte.</li> </ul> <p>Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist ebenfalls zulässig, wenn die Datenbankspalte mit FOR BIT DATA markiert ist. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge in jedem Fall mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.</p>

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
LONG VARGRAPHIC (nur DBCS)	Eine PC/IXF-Spalte des Typs LONG VARGRAPHIC wird erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spalten-deskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.
SMALLINT	Eine Spalte des Typs SMALLINT wird erstellt, die mit der Datenbankspalte identisch ist.	Eine Spalte eines beliebigen numerischen Typs (SMALLINT, INTEGER, BIGINT, DECIMAL oder FLOAT) wird akzeptiert. Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich -32 768 bis 32 767 liegen.
TIME	Eine Spalte des Typs TIME wird erstellt, die mit der Datenbankspalte identisch ist.	Eine PC/IXF-Spalte des Typs TIME ist die übliche Eingabe. Das Dienstprogramm IMPORT versucht auch, Spalten in einem beliebigen Zeichentyp anzunehmen, sofern sie keine inkompatiblen Längen aufweisen. Die Zeichenspalte in der PC/IXF-Datei muß Zeitwerte in einem Format enthalten, das mit dem Landescode der Zieldatenbank konsistent ist.
TIMESTAMP	Eine Spalte des Typs TIMESTAMP wird erstellt, die mit der Datenbankspalte identisch ist.	Eine PC/IXF-Spalte des Typs TIMESTAMP ist die übliche Eingabe. Das Dienstprogramm IMPORT versucht auch, Spalten in einem beliebigen Zeichentyp anzunehmen, sofern sie keine inkompatiblen Längen aufweisen. Die Zeichenspalte in der PC/IXF-Datei muß Daten im Eingabeformat für Zeitmarken enthalten.

## PC-Version des IXF-Dateiformats

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
VARCHAR	<p>Wenn die maximale Länge der Datenbankspalte <math>\leq 254</math> ist, wird eine PC/IXF-Spalte des Typs VARCHAR erstellt. Wenn die maximale Länge der Datenbankspalte <math>&gt; 254</math> ist, wird eine PC/IXF-Spalte des Typs LONG VARCHAR erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spaltendeskriptorsatz kopiert.</p>	<p>Eine PC/IXF-Spalte des Typs CHAR, VARCHAR oder LONG VARCHAR ist in folgenden Fällen zulässig:</p> <ul style="list-style-type: none"> <li>• Die Datenbankspalte weist die Markierung FOR BIT DATA auf.</li> <li>• Der Einzelbyte-Codepage-Wert der PC/IXF-Spalte stimmt mit der SBCS-CPGID der Datenbankspalte überein, und der Doppelbyte-Codepage-Wert der PC/IXF-Spalte entspricht Null oder der DBCS-CPGID der Datenbankspalte.</li> </ul> <p>Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist ebenfalls zulässig, wenn die Datenbankspalte mit FOR BIT DATA markiert ist. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge in jedem Fall mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.</p>

Tabelle 13. Gültige Datentypformate für das Dateiformat PC/IXF (Forts.)

Datentyp	Format in vom Dienstprogramm EXPORT erstellten Dateien	Für das Dienstprogramm IMPORT zulässiges Format
VARGRAPHIC (nur DBCS)	Wenn die maximale Länge der Datenbankspalte <= 127 ist, wird eine PC/IXF-Spalte des Typs VARGRAPHIC erstellt. Wenn die maximale Länge der Datenbankspalte > 127 ist, wird eine PC/IXF-Spalte des Typs LONG VARGRAPHIC erstellt. Die maximale Länge der Datenbankspalte, der SBCS-CPGID-Wert und der DBCS-CPGID-Wert werden in den Spalten-deskriptorsatz kopiert.	Eine PC/IXF-Spalte des Typs GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC ist zulässig, wenn der Doppelbyte-Codepage-Wert der PC/IXF-Spalte mit dem der Datenbankspalte übereinstimmt. Wenn die PC/IXF-Spalte eine feste Länge besitzt, muß ihre Länge mit der maximalen Länge der Datenbankspalte kompatibel sein. Siehe auch „Option FORCEIN“ auf Seite 298.

### Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken

Das Dienstprogramm IMPORT des Datenbankmanagers wendet beim Importieren einer PC/IXF-Datei in einer SBCS- oder DBCS-Umgebung die folgenden allgemeinen Regeln an:

- Das Dienstprogramm IMPORT akzeptiert nur Dateien im PC/IXF-Format (IXFHID = 'IXF'). IXF-Dateien anderer Formate können nicht importiert werden.
- Das Dienstprogramm IMPORT weist eine PC/IXF-Datei mit mehr als 1024 Spalten zurück.
- Der Wert von IXFHSBCP im PC/IXF-H-Datensatz muß mit der SBCS-CPGID übereinstimmen, oder es muß eine Umsetzungstabelle zwischen IXFHSBCP/IXFHDBCP und der SBCS/DBCS-CPGID der Zieldatenbank geben. Der Wert von IXFHDBCP muß entweder '00000' sein oder dem Wert der DBCS-CPGID der Zieldatenbank entsprechen. Wenn eine dieser Bedingungen nicht erfüllt ist, weist das Dienstprogramm IMPORT die PC/IXF-Datei zurück, sofern nicht die „Option FORCEIN“ auf Seite 298 angegeben ist.
- Ungültige Datentypen — neue Tabelle  
Der Import einer PC/IXF-Datei in eine *neue* Tabelle wird durch die Schlüsselwörter CREATE oder REPLACE\_CREATE des Befehls IMPORT angegeben. Wenn eine PC/IXF-Spalte mit einem ungültigen Datentyp (gültige Typen sind in „PC/IXF-Datentypen“ auf Seite 277 definiert) für den Import in eine neue Tabelle ausgewählt wird, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, es wird keine Tabelle erstellt, und es werden keine Daten importiert.
- Ungültige Datentypen — vorhandene Tabelle  
Der Import einer PC/IXF-Datei in eine *vorhandene* Tabelle wird durch die Schlüsselwörter INSERT oder INSERT\_UPDATE des Befehls IMPORT angegeben. Wenn eine PC/IXF-Spalte mit einem ungültigen Datentyp für den Import in eine vorhandene Tabelle ausgewählt wird, ist eine von zwei Aktionen möglich:
  - Wenn die Zieltabellenspalte Nullwerte enthalten darf, werden alle Werte der ungültigen PC/IXF-Spalte ignoriert, und die Tabellenspaltenwerte werden auf NULL gesetzt.
  - Wenn die Zieltabellenspalte keine Nullwerte enthalten darf, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, und es werden keine Daten importiert. Die vorhandene Tabelle bleibt unverändert.



- Beim Import in eine neue Tabelle bewirken PC/IXF-Spalten, die Nullwerte enthalten dürfen, die Generierung von Datenbankspalten, die Nullwerte enthalten dürfen. PC/IXF-Spalten, die keine Nullwerte enthalten dürfen, bewirken die Generierung von Datenbankspalten, die keine Nullwerte enthalten dürfen.
- Eine PC/IXF-Spalte, die keine Nullwerte enthalten darf, kann in eine Datenbankspalte, die Nullwerte enthalten darf, importiert werden.
- Eine PC/IXF-Spalte, die Nullwerte enthalten darf, kann in eine Datenbankspalte, die keine Nullwerte enthalten darf, importiert werden. Wenn ein NULL-Wert in der PC/IXF-Spalte festgestellt wird, weist das Dienstprogramm IMPORT die Werte aller Spalten in der PC/IXF-Zeile zurück, die den NULL-Wert enthält (die ganze Zeile wird zurückgewiesen), und die Verarbeitung wird bei der nächsten PC/IXF-Zeile fortgesetzt. Das bedeutet, daß keine Daten aus einer PC/IXF-Zeile, die einen NULL-Wert enthält, importiert werden, wenn eine Zieltabellenspalte keine Nullwerte enthalten darf.
- Inkompatible Spalten — neue Tabelle  
Wenn beim Import in eine *neue* Datenbanktabelle eine PC/IXF-Spalte ausgewählt wird, die mit der Zieldatenbankspalte inkompatibel ist, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, es wird keine Tabelle erstellt, und es werden keine Daten importiert.

**Anmerkung:** Die IMPORT-Option FORCEIN (siehe „Option FORCEIN“ auf Seite 298) erweitert den Bereich kompatibler Spalten.

- Inkompatible Spalten — vorhandene Tabelle  
Wenn beim Import in eine *vorhandene* Datenbanktabelle eine PC/IXF-Spalte ausgewählt wird, die zur Zieldatenbankspalte inkompatibel ist, ist eine von zwei Aktionen möglich:
  - Wenn die Zieltabellenspalte Nullwerte enthalten darf, werden alle Werte der PC/IXF-Spalte ignoriert, und die Tabellenspaltenwerte werden auf NULL gesetzt.
  - Wenn die Zieltabellenspalte keine Nullwerte enthalten darf, wird das Dienstprogramm IMPORT beendet. Die gesamte PC/IXF-Datei wird zurückgewiesen, und es werden keine Daten importiert. Die vorhandene Tabelle bleibt unverändert.

**Anmerkung:** Die IMPORT-Option FORCEIN (siehe „Option FORCEIN“ auf Seite 298) erweitert den Bereich kompatibler Spalten.

## PC-Version des IXF-Dateiformats

- Ungültige Werte

Wenn beim Import ein PC/IXF-Spaltenwert festgestellt wird, der für die Zieldatenbankspalte ungültig ist, weist das Dienstprogramm IMPORT die Werte aller Spalten in der PC/IXF-Zeile mit dem ungültigen Wert zurück, d. h. die gesamte Zeile wird zurückgewiesen, und die Verarbeitung wird bei der nächste PC/IXF-Zeile fortgesetzt.

- Das Importieren oder Laden von PC/IXF-Dateien, die DBCS-Daten enthalten, erfordert, daß die entsprechenden Umsetzungsdateien (in `sqllib\conv`) auf der Client-Maschine installiert sind. Die Namen dieser Umsetzungsdateien enthalten die Nummern der Quellen- und der Ziel-Codepage. Die Erweiterung ist stets `.cnv`. Die Datei `09320943.cnv` beispielsweise enthält die Tabelle zur Umsetzung von Codepage 932 in 943.

Wenn die Client-Maschine nicht über die entsprechenden Umsetzungsdateien verfügt, können sie von einer Server-Maschine in das Verzeichnis `sqllib\conv` auf der Client-Maschine kopiert werden. Achten Sie darauf, die Dateien von einer kompatiblen Plattform zu kopieren. Wird der Client beispielsweise unter einem UNIX-Betriebssystem ausgeführt, müssen Sie die Dateien von einem Server kopieren, der ebenfalls unter einem UNIX-Betriebssystem ausgeführt wird.

### Datentypspezifische Regeln für den Import von PC/IXF-Dateien in Datenbanken

- Eine gültige numerische PC/IXF-Spalte kann in jede beliebige kompatible numerische Datenbankspalte importiert werden. PC/IXF-Spalten, die 4 Byte große Gleitkommatdaten enthalten, werden nicht importiert, weil dies ein ungültiger Datentyp ist.
- Datum-/Uhrzeitspalten der Datenbank können Werte aus entsprechenden PC/IXF-Datums-/Uhrzeitspalten (DATE, TIME und TIMESTAMP) sowie aus anderen PC/IXF-Zeichenspalten (CHAR, VARCHAR und LONG VARCHAR) akzeptieren, wobei jedoch die Einschränkungen hinsichtlich der Spaltenlänge und der Wertekompatibilität gelten.
- Eine gültige PC/IXF-Zeichenspalte (CHAR, VARCHAR oder LONG VARCHAR) kann immer in eine *vorhandene* Datenbankzeichenspalte mit der Markierung FOR BIT DATA importiert werden. Ansonsten gilt folgendes:
  - IXFCSBCP und die SBCS-CPGID müssen übereinstimmen.
  - Es muß eine Umsetzungstabelle für IXFCSBCP/IXFCDBCP und SBCS/DBCS vorhanden sein.
  - Ein Satz muß ausschließlich aus Nullen bestehen (FOR BIT DATA).

Wenn IXFCSBCP nicht Null ist, muß der Wert von IXFCDBCP entweder Null sein oder mit der DBCS-CPGID der Zieldatenbankspalte übereinstimmen.

Wenn eine dieser Bedingungen nicht erfüllt ist, sind die PC/IXF-Spalten und die Datenbankspalten inkompatibel.

Beim Importieren einer gültigen PC/IXF-Zeichenspalte in eine *neue* Datenbanktabelle muß der Wert von IXFCSBCP Null sein oder mit der SBCS-CPGID der Datenbank übereinstimmen, oder es ist eine Umsetzungstabelle erforderlich. Wenn IXFCSBCP Null ist, muß auch IXFCDBCP Null sein (andernfalls hat die PC/IXF-Spalte einen ungültigen Datentyp). Von IMPORT wird eine Zeichenspalte mit der Markierung FOR BIT DATA in der neuen Tabelle erstellt. Wenn IXFCSBCP nicht Null ist und mit der SBCS-CPGID der Datenbank übereinstimmt, muß der Wert von IXFCDBCP entweder Null sein oder mit der DBCS-CPGID der Datenbank übereinstimmen. In diesem Fall wird vom Dienstprogramm eine Zeichenspalte in der neuen Tabelle erstellt, wobei die SBCS- und DBCS-CPGID-Werte mit denen der Datenbank übereinstimmen. Wenn diese Bedingungen nicht erfüllt sind, sind die PC/IXF-Spalten und die Datenbankspalten inkompatibel.

Die Option FORCEIN (siehe „Option FORCEIN“ auf Seite 298) kann verwendet werden, um Codepage-Gleichheitsüberprüfungen zu überschreiben. Eine PC/IXF-Zeichenspalte mit IXFCSBCP gleich Null und IXFCDBCP ungleich Null ist jedoch ein ungültiger Datentyp, der nicht importiert werden kann, auch dann nicht, wenn FORCEIN angegeben ist.

- Eine gültige PC/IXF-Grafikspalte (GRAPHIC, VARGRAPHIC oder LONG VARGRAPHIC) kann immer in eine *vorhandene* Datenbankzeichenspalte mit der Markierung FOR BIT DATA importiert werden, ist jedoch mit allen anderen Datenbankspalten inkompatibel. Die Option FORCEIN (siehe „Option FORCEIN“ auf Seite 298) kann verwendet werden, um die Einschränkung abzuschwächen. Eine PC/IXF-Grafikspalte mit IXFCSBCP ungleich Null und IXFCDBCP gleich Null ist jedoch ein ungültiger Datentyp, der nicht importiert werden kann, auch dann nicht, wenn FORCEIN angegeben ist.

Beim Import einer gültigen PC/IXF-Grafikspalte in eine Datenbankgrafikspalte muß der Wert von IXFCDBCP mit der DBCS-CPGID der Zieldatenbankspalte übereinstimmen (das heißt, die Doppelbyte-Codepages der beiden Spalten müssen übereinstimmen).

## PC-Version des IXF-Dateiformats

- Wenn beim Import einer PC/IXF-Datei in eine vorhandene Datenbanktabelle eine Zeichenfolgespalte mit fester Länge (CHAR oder GRAPHIC) ausgewählt wird, deren Länge die maximale Länge der Zielspalte überschreitet, sind die Spalten inkompatibel.
- Wenn beim Import einer PC/IXF-Datei in eine vorhandene Datenbanktabelle eine Zeichenfolgespalte mit variabler Länge (VARCHAR, LONG VARCHAR, VARGRAPHIC oder LONG VARGRAPHIC) ausgewählt wird, deren Länge die maximale Länge der Zielspalte überschreitet, *sind* die Spalten kompatibel. Einzelne Werte werden anhand der Kompatibilitätsregeln verarbeitet, die für die INSERT-Anweisung des Datenbankmanagers gelten. PC/IXF-Werte, die zu lang für die Zieldatenbankspalte sind, sind ungültig.
- PC/IXF-Werte, die in eine *Zeichenspalte* der Datenbank mit fester Länge (d. h. eine CHAR-Spalte) importiert werden, werden nach Bedarf rechts mit Einzelbyteleerzeichen (0x20) aufgefüllt, um Werte zu erhalten, deren Länge mit der der Datenbankspalte übereinstimmt. PC/IXF-Werte, die in eine *Grafikspalte* der Datenbank mit fester Länge (d. h. eine GRAPHIC-Spalte) importiert werden, werden nach Bedarf rechts mit Doppelbyteleerzeichen (0x8140) aufgefüllt, um Werte zu erhalten, deren Länge mit der der Datenbankspalte übereinstimmt.
- Da PC/IXF-Spalten des Typs VARCHAR maximal 254 Byte lang sein dürfen, muß eine VARCHAR-Spalte der Datenbank mit der maximalen Länge  $n$  (wobei  $254 < n < 4001$  gilt) in eine PC/IXF-Spalte LONG VARCHAR der maximalen Länge  $n$  exportiert werden.
- Obwohl PC/IXF-Spalten des Typs LONG VARCHAR maximal 32 767 Byte lang sein dürfen und LONG VARCHAR-Spalten der Datenbank maximal 32 700 Byte lang sein dürfen, sind PC/IXF-Spalten des Typs LONG VARCHAR, die länger sind als 32 700 Byte (aber kürzer als 32 768 Byte), gültig und können in LONG VARCHAR-Spalten der Datenbank importiert werden. Allerdings können Daten verlorengehen.
- Da PC/IXF-Spalten des Typs VARGRAPHIC maximal 127 Byte lang sein dürfen, muß eine VARGRAPHIC-Spalte der Datenbank mit der maximalen Länge  $n$  (wobei  $127 < n < 2001$  gilt) in eine PC/IXF-Spalte LONG VARGRAPHIC der maximalen Länge  $n$  exportiert werden.
- Obwohl PC/IXF-Spalten des Typs LONG VARGRAPHIC maximal 16 383 Byte lang sein dürfen und LONG VARGRAPHIC-Spalten der Datenbank maximal 16 350 Byte lang sein dürfen, sind PC/IXF-Spalten des Typs LONG VARGRAPHIC, die länger sind als 16 350 Byte (aber kürzer als 16 384 Byte), gültig und können in LONG VARGRAPHIC-Spalten der Datenbank importiert werden. Allerdings können Daten verlorengehen.

Tabelle 14 bietet eine Übersicht des Imports von PC/IXF-Dateien in neue oder vorhandene Datenbanktabellen ohne Verwendung der Option FORCEIN.

Tabelle 14. Übersicht des Imports von PC/IXF-Dateien ohne Verwendung der Option FORCEIN

PC/IXF- SPALTEN- DATENTYP	DATENBANKSPALTENDATENTYP											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>d</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
NUMERIC (Numerisch)												
-SMALLINT	N											
	V	V	V	V <sup>a</sup>	V							
-INTEGER		N										
	V <sup>a</sup>	V	V	V <sup>a</sup>	V							
-BIGINT			N									
	V <sup>a</sup>	V <sup>a</sup>	V	V <sup>a</sup>	V							
-DECIMAL				N								
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
-FLOAT					N							
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
CHARACTER (Zeichen)												
-(0,0)						N						
						V				V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS,0)							N	N				
						V	V	V		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS, DBCS)								N		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
						V		V				
GRAPH (Grafik)												
									N			
						V			V			
DATETIME (Datum und Uhrzeit)												
-DATE										N		
										V		
-TIME											N	
											V	
-TIME STAMP												N
												V

## PC-Version des IXF-Dateiformats

Tabelle 14. Übersicht des Imports von PC/IXF-Dateien ohne Verwendung der Option FORCEIN (Forts.)

PC/IXF- SPALTEN- DATENTYP	DATENBANKSPALTENDATENTYP											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>d</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
<b>Anmerkungen:</b>												
<p>1. Die Tabelle ist eine Matrix aller gültigen PC/IXF- und Datenbankmanagerdatentypen. Wenn eine PC/IXF-Spalte in eine Datenbankspalte importiert werden kann, wird in der Matrixzelle am Schnittpunkt der Matrixzeile des PC/IXF-Datentyps und der Matrixspalte des Datenbankmanager-Datentyps ein Buchstabe angezeigt. Ein 'N' gibt an, daß das Dienstprogramm eine neue Datenbanktabelle erstellt (es wird eine Datenbankspalte des angegebenen Datentyps erstellt). Ein 'V' gibt an, daß das Dienstprogramm Daten in eine vorhandene Datenbanktabelle importiert (eine Datenbankspalte des angegebenen Datentyps ist ein gültiges Ziel).</p> <p>2. Zeichenfolgedatentypen werden anhand von Codepage-Attributen unterschieden. Diese Attribute werden als geordnetes Paar (SBCS,DBCS) angezeigt, wobei folgendes gilt:</p> <ul style="list-style-type: none"> <li>• SBCS ist entweder Null oder gibt einen Wert ungleich Null eines Einzelbyte-Codepage-Attributs des Zeichendatentyps an.</li> <li>• DBCS ist entweder Null oder gibt einen Wert ungleich Null eines Doppelbyte-Codepage-Attributs des Zeichendatentyps an.</li> </ul> <p>3. Wenn die Tabelle angibt, daß eine PC/IXF-Zeichenspalte in eine Datenbankzeichenspalte importiert werden kann, entsprechen die Werte der jeweiligen Codepage-Attributepaare den Regeln, die für die Gleichheit von Codepages gelten.</p> <p><sup>a</sup> Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich für den numerischen Zieldatentyp liegen.</p> <p><sup>b</sup> Datentyp ist nur in DBCS-Umgebungen verfügbar.</p> <p><sup>c</sup> Einzelne Werte werden zurückgewiesen, wenn sie keine gültigen Datums- oder Uhrzeitwerte darstellen.</p> <p><sup>d</sup> Datentyp ist in DBCS-Umgebungen nicht verfügbar.</p>												

### Option FORCEIN

Die Option FORCEIN ermöglicht den Import einer PC/IXF-Datei trotz Codepage-Unterschieden zwischen Daten in der PC/IXF-Datei und der Zieldatenbank. Sie bietet zusätzliche Flexibilität bei der Definition kompatibler Spalten.

#### Allgemeine FORCEIN-Semantik

Für die Verwendung der Option FORCEIN in einer SBCS- oder DBCS-Umgebung gilt die folgende allgemeine Semantik:

- Die Option FORCEIN sollte mit Bedacht verwendet werden. Es ist zumeist ratsam, zu versuchen, einen Import ohne Aktivierung dieser Option durchzuführen. Aufgrund der generischen Beschaffenheit der PC/IXF-Datenaustauscharchitektur können manche PC/IXF-Dateien jedoch Datentypen oder Werte enthalten, die nicht ohne Eingriff importiert werden können.
- Der Import mit FORCEIN in eine *neue* Tabelle kann zu einem anderen Ergebnis führen als der Import in eine vorhandene Tabelle. Eine vorhandene Tabelle enthält vordefinierte Zieldatentypen für jeden PC/IXF-Datentyp.

- Wenn LOB-Daten mit der Option LOBSINFILE exportiert und die Dateien auf einen anderen Client mit einer anderen Codepage versetzt werden, werden die CLOBS und DBCLOBS in den separaten Dateien anders als andere Daten beim Importieren oder Laden in eine Datenbank nicht in die Client-Codepage umgesetzt.

### **FORCEIN-Codepage-Semantik**

Für die Verwendung der Option FORCEIN in einer SBCS- oder DBCS-Umgebung gilt die folgende Codepage-Semantik:

- Die Option FORCEIN inaktiviert alle Codepage-Vergleiche des Dienstprogramms IMPORT.  
Diese Regel gilt für Codepage-Vergleiche auf Spalten- und auch auf Dateiebene, wenn in eine neue oder vorhandene Datenbanktabelle importiert wird. Auf der Spaltenebene (z. B. Datentyp) gilt diese Regel nur für die folgenden Datenbankmanager- und PC/IXF-Datentypen: Zeichen (CHAR, VARCHAR und LONG VARCHAR) und Grafik (GRAPHIC, VARGRAPHIC und LONG VARGRAPHIC). Die Einschränkung ergibt sich daraus, daß die Codepage-Attribute anderer Datentypen für die Auswertung der Datentypwerte nicht von Bedeutung sind.
- Die Option FORCEIN inaktiviert nicht die Prüfung der Codepage-Attribute zur Ermittlung der Datentypen.  
Beispiel: Der Datenbankmanager erlaubt die Deklaration einer CHAR-Spalte mit dem Attribut FOR BIT DATA. Eine solche Deklaration setzt sowohl die SBCS-CPGID als auch die DBCS-CPGID der Spalte auf Null. Es ist der Nullwert dieser CPGIDs, der die Spaltenwerte als Bitfolgen (statt als Zeichenfolgen) angibt.
- Die Option FORCEIN impliziert keine Codepage-Umsetzung.  
Werte mit Datentypen, die von der Option FORCEIN betroffen sind, werden „unverändert“ kopiert. Es werden keine Codepunktzuordnungen angewendet, um eine Änderung der Codepage-Umgebungen zu berücksichtigen. Das Auffüllen des importierten Werts mit Leerzeichen kann im Falle von Zielspalten mit fester Länge erforderlich sein.
- Wenn Daten mit der Option FORCEIN in eine *vorhandene* Tabelle importiert werden, gilt folgendes:
  - Der Codepage-Wert der Zieldatenbanktabelle und -spalten ist immer vorrangig.
  - Der Codepage-Wert der PC/IXF-Datei und -Spalten wird ignoriert.

Diese Regel gilt unabhängig davon, ob die Option FORCEIN verwendet wird. Der Datenbankmanager läßt nach der Erstellung der Datenbank keine Änderungen an einem Codepage-Wert einer Datenbank oder Spalte zu.

## PC-Version des IXF-Dateiformats

- Beim Importieren in eine *neue* Tabelle unter Verwendung der Option FORCEIN gilt folgendes:
  - Der Codepage-Wert der Zieldatenbank ist immer vorrangig.
  - PC/IXF-Zeichenspalten mit IXFCSBCP = IXFCDBCP = 0 generieren Tabellenspalten mit dem Attribut FOR BIT DATA.
  - Alle anderen PC/IXF-Zeichenspalten generieren Tabellenzeichenspalten mit SBCS- und DBCS-CPGID-Werten, die mit denen der Datenbank identisch sind.
  - PC/IXF-Grafikspalten generieren Tabellengrafikspalten mit der SBCS-CPGID „undefiniert“ und einer DBCS-CPGID, die mit der der Datenbank identisch ist (nur DBCS-Umgebung).

### Beispiel für FORCEIN

Angenommen, Sie haben eine PC/IXF-Spalte des Typs CHAR mit IXFCSBCP = '00897' und IXFCDBCP = '00301'. Diese Spalte soll in eine Datenbankspalte des Typs CHAR mit SBCS-CPGID = '00850' und DBCS-CPGID = '00000' importiert werden. Ohne FORCEIN wird das Dienstprogramm beendet, und es werden keine Daten importiert, oder aber die PC/IXF-Spaltenwerte werden ignoriert und die Datenbankspalte enthält NULL-Werte (wenn die Datenbankspalte Nullwerte enthalten darf). Mit FORCEIN wird das Dienstprogramm fortgesetzt, und Codepage-Inkompatibilitäten werden ignoriert. Wenn keine weiteren Datentypinkompatibilitäten (wie beispielsweise die Länge) auftreten, werden die Werte der PC/IXF-Spalte „unverändert“ importiert und stehen zur Auswertung unter der Codepage-Umgebung der Datenbank zur Verfügung.

Die folgende Tabelle zeigt:

- Die Codepage-Attribute einer Spalte, die in einer *neuen* Datenbanktabelle erstellt wird, wenn aus der PC/IXF-Datei ein Datentyp mit angegebenen Codepage-Attributen importiert wird.
- Daß das Dienstprogramm IMPORT die PC/IXF-Datentypen zurückweist, wenn sie ungültig oder inkompatibel sind.



Tabelle 15. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (neue Tabelle). In dieser Tabelle wird davon ausgegangen, daß es keine Umsetzungstabelle zwischen a und x gibt. Gäbe es eine solche, würden die Elemente 3 und 4 auch ohne Verwendung der Option FORCEIN funktionieren.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER DATENBANK- TABELLENSPALTE	
	Ohne FORCEIN	Mit FORCEIN
SBCS		
(0,0)	(0,0)	(0,0)
(a,0)	(a,0)	(a,0)
(x,0)	zurückweisen	(a,0)
(x,y)	zurückweisen	(a,0)
(a,y)	zurückweisen	(a,0)
(0,y)	zurückweisen	(0,0)
DBCS		
(0,0)	(0,0)	(0,0)
(a,0)	(a,b)	(a,b)
(x,0)	zurückweisen	(a,b)
(a,b)	(a,b)	(a,b)
(x,y)	zurückweisen	(a,b)
(a,y)	zurückweisen	(a,b)
(x,b)	zurückweisen	(a,b)
(0,b)	(-,b)	(-,b)
(0,y)	zurückweisen	(-,b)

## PC-Version des IXF-Dateiformats

Tabelle 15. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (neue Tabelle) (Forts.). In dieser Tabelle wird davon ausgegangen, daß es keine Umsetzungstabelle zwischen a und x gibt. Gäbe es eine solche, würden die Elemente 3 und 4 auch ohne Verwendung der Option *FORCEIN* funktionieren.

CODEPAGE-ATTRIBUTE DES PC/IXF-DATENTYPS	CODEPAGE-ATTRIBUTE DER DATENBANK- TABELLENSPALTE	
	Ohne <i>FORCEIN</i>	Mit <i>FORCEIN</i>
<p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"> <li>Codepage-Attribute eines PC/IXF-Datentyps werden als geordnetes Paar gezeigt, wobei x für einen Einzelbyte-Codepage-Wert ungleich Null und y für einen Doppelbyte-Codepage-Wert ungleich Null steht. Ein '-' steht für einen nicht definierten Codepage-Wert.</li> <li>Die Verwendung verschiedener Buchstaben in verschiedenen Codepage-Attributpaaren ist willkürlich. Verschiedene Buchstaben implizieren verschiedene Werte. Beispiel: Wenn ein PC/IXF-Datentyp als (x,y) gezeigt wird und die Datenbankspalte als (a,y), entspricht x nicht a, die PC/IXF-Datei und die Datenbank haben jedoch den gleichen Doppelbyte-Codepage-Wert y.</li> <li>Nur Zeichen- und Grafikdatentypen sind von der <i>FORCEIN</i>-Codepage-Semantik betroffen.</li> <li>Es wird angenommen, daß die Datenbank, die die neue Tabelle enthält, die Codepage-Attribute (a,0) hat. Demzufolge müssen alle Zeichenspalten in der neuen Tabelle die Codepage-Attribute (0,0) oder (a,0) haben.  In einer DBCS-Umgebung wird angenommen, daß die Datenbank, die die neue Tabelle enthält, die Codepage-Attribute (a,b) hat. Demzufolge müssen alle Grafikspalten in der neuen Tabelle die Codepage-Attribute (-,b) und alle Zeichenspalten die Codepage-Attribute (a,b) haben. Die SBCS-CPGID wird als '-', gezeigt, weil sie für Grafikdatentypen nicht definiert ist.</li> <li>Der Datentyp des Ergebnisses wird durch die in „<i>FORCEIN</i>-Datentypsemantik“ auf Seite 305 beschriebenen Regeln bestimmt.</li> <li>Das Ergebnis zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen (siehe „Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 292).</li> </ol>		

Die folgende Tabelle zeigt:

- Daß das Dienstprogramm *IMPORT* PC/IXF-Datentypen mit verschiedenen Codepage-Attributen in einer *vorhandenen* Tabellenspalte (der *Zielspalte*) mit den angegebenen Codepage-Attributen akzeptiert.
- Daß das Dienstprogramm *IMPORT* den Import eines PC/IXF-Datentyps mit bestimmten Codepage-Attributen in eine *vorhandene* Tabellenspalte mit den gezeigten Codepage-Attributen nicht zuläßt. Das Dienstprogramm weist die PC/IXF-Datentypen zurück, wenn sie ungültig oder inkompatibel sind.

Tabelle 16. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (vorhandene Tabelle). In dieser Tabelle wird davon ausgegangen, daß es keine Umsetzungstabelle zwischen a und x gibt.

CODEPAGE- ATTRIBUTE DES PC/IXF- DATENTYPS	CODEPAGE- ATTRIBUTE DER ZIELDATEN- BANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne FORCEIN	Mit FORCEIN
SBCS			
(0,0)	(0,0)	akzeptieren	akzeptieren
(a,0)	(0,0)	akzeptieren	akzeptieren
(x,0)	(0,0)	akzeptieren	akzeptieren
(x,y)	(0,0)	akzeptieren	akzeptieren
(a,y)	(0,0)	akzeptieren	akzeptieren
(0,y)	(0,0)	akzeptieren	akzeptieren
(0,0)	(a,0)	Null oder zurückweisen	akzeptieren
(a,0)	(a,0)	akzeptieren	akzeptieren
(x,0)	(a,0)	Null oder zurückweisen	akzeptieren
(x,y)	(a,0)	Null oder zurückweisen	akzeptieren
(a,y)	(a,0)	Null oder zurückweisen	akzeptieren
(0,y)	(a,0)	Null oder zurückweisen	Null oder zurückweisen
DBCS			
(0,0)	(0,0)	akzeptieren	akzeptieren
(a,0)	(0,0)	akzeptieren	akzeptieren
(x,0)	(0,0)	akzeptieren	akzeptieren
(a,b)	(0,0)	akzeptieren	akzeptieren
(x,y)	(0,0)	akzeptieren	akzeptieren
(a,y)	(0,0)	akzeptieren	akzeptieren
(x,b)	(0,0)	akzeptieren	akzeptieren
(0,b)	(0,0)	akzeptieren	akzeptieren
(0,y)	(0,0)	akzeptieren	akzeptieren

## PC-Version des IXF-Dateiformats

Tabelle 16. Übersicht über die Codepage-Semantik des Dienstprogramms *IMPORT* (vorhandene Tabelle) (Forts.). In dieser Tabelle wird davon ausgegangen, daß es keine Umsetzungstabelle zwischen a und x gibt.

CODEPAGE- ATTRIBUTE DES PC/IXF- DATENTYPS	CODEPAGE- ATTRIBUTE DER ZIELDATEN- BANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne FORCEIN	Mit FORCEIN
(0,0)	(a,b)	Null oder zurückweisen	akzeptieren
(a,0)	(a,b)	akzeptieren	akzeptieren
(x,0)	(a,b)	Null oder zurückweisen	akzeptieren
(a,b)	(a,b)	akzeptieren	akzeptieren
(x,y)	(a,b)	Null oder zurückweisen	akzeptieren
(a,y)	(a,b)	Null oder zurückweisen	akzeptieren
(x,b)	(a,b)	Null oder zurückweisen	akzeptieren
(0,b)	(a,b)	Null oder zurückweisen	Null oder zurückweisen
(0,y)	(a,b)	Null oder zurückweisen	Null oder zurückweisen
(0,0)	(-,b)	Null oder zurückweisen	akzeptieren
(a,0)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(x,0)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(a,b)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(x,y)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(a,y)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(x,b)	(-,b)	Null oder zurückweisen	Null oder zurückweisen
(0,b)	(-,b)	akzeptieren	akzeptieren

Tabelle 16. Übersicht über die Codepage-Semantik des Dienstprogramms IMPORT (vorhandene Tabelle) (Forts.). In dieser Tabelle wird davon ausgegangen, daß es keine Umsetzungstabelle zwischen a und x gibt.

CODEPAGE- ATTRIBUTE DES PC/IXF- DATENTYPS	CODEPAGE- ATTRIBUTE DER ZIELDATEN- BANKSPALTE	ERGEBNISSE DES IMPORTS	
		Ohne FORCEIN	Mit FORCEIN
(0,y)	(-,b)	Null oder zurückweisen	akzeptieren
<p><b>Anmerkungen:</b></p> <ol style="list-style-type: none"> <li>1. Siehe Anmerkungen zu Tabelle 15 auf Seite 301.</li> <li>2. Das Ergebnis Null oder zurückweisen ist eine Folge der Regeln für ungültige oder inkompatible Datentypen (siehe „Allgemeine Regeln für den Import von PC/IXF-Dateien in Datenbanken“ auf Seite 292).</li> </ol>			

### FORCEIN-Datentypsemantik

Die Option FORCEIN ermöglicht den Import bestimmter PC/IXF-Spalten in Zieldatenbankspalten mit ungleichen oder anderweitig inkompatiblen Datentypen. Für die Verwendung der Option FORCEIN in einer SBCS- oder DBCS-Umgebung gilt die folgende allgemeine Datentypsemantik (sofern nicht anders angegeben):

- In SBCS-Umgebungen kann mit der Option FORCEIN folgendes importiert werden:
  - Ein PC/IXF-Datentyp BIT (IXFCSBCP = 0 = IXFCDBCP für eine PC/IXF-Zeichenspalte) in eine Datenbankzeichenspalte (SBCS-CPGID ungleich Null und DBCS-CPGID gleich Null); nur vorhandene Tabellen
  - Ein PC/IXF-Datentyp MIXED (IXFCSBCP und IXFCDBCP ungleich Null) in eine Datenbankzeichenspalte; neue und vorhandene Tabellen
  - Ein PC/IXF-Datentyp GRAPHIC in eine Datenbankspalte mit dem Attribut FOR BIT DATA (SBCS-CPGID = 0 = DBCS-CPGID); nur neue Tabellen (dies ist für vorhandene Tabellen immer zulässig)
- Mit der Option FORCEIN wird der Bereich gültiger PC/IXF-Datentypen nicht erweitert.

PC/IXF-Spalten mit Datentypen, die in „PC/IXF-Datentypen“ auf Seite 277 nicht als gültig definiert sind, sind mit oder ohne die Option FORCEIN für den Import ungültig.

## PC-Version des IXF-Dateiformats

- In DBCS-Umgebungen kann mit der Option FORCEIN folgendes importiert werden:
  - Ein PC/IXF-Datentyp BIT in eine Datenbankzeichenspalte
  - Ein PC/IXF-Datentyp BIT in eine Grafikspalte der Datenbank. Wenn die PC/IXF-Spalte des Typs BIT eine feste Länge besitzt, muß die Länge ein gerader Wert sein. Eine PC/IXF-Spalte des Typs BIT mit fester Länge und einem ungeraden Längenwert ist mit einer Grafikspalte der Datenbank nicht kompatibel. Eine PC/IXF-Spalte des Typs BIT mit variabler Länge ist kompatibel, unabhängig davon, ob der Längenwert gerade oder ungerade ist, obwohl ein ungerader Längenwert aus einer Spalte mit variabler Länge ein ungültiger Wert für den Import in eine Grafikspalte der Datenbank ist.
  - Ein PC/IXF-Datentyp MIXED in eine Datenbankzeichenspalte

Tabelle 17 bietet eine Übersicht des Imports von PC/IXF-Dateien in neue oder vorhandene Datenbanktabellen unter Verwendung der Option FORCEIN.

Tabelle 17. Übersicht des Imports von PC/IXF-Dateien mit Verwendung der Option FORCEIN

PC/IXF- SPALTEN- DATENTYP	DATENBANKSPALTENDATENTYP											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>e</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
NUMERIC (Numerisch)												
-SMALLINT	N											
	V	V	V	V <sup>a</sup>	V							
-INTEGER		N										
	V <sup>a</sup>	V	V	V <sup>a</sup>	V							
-BIGINT			N									
	V <sup>a</sup>	V <sup>a</sup>	V	V <sup>a</sup>	V							
-DECIMAL				N								
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
-FLOAT					N							
	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V <sup>a</sup>	V							
CHARACTER (Zeichen)												
-(0,0)						N						
						V	V mit F	V mit F	V mit F	V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
-(SBCS,0)							N	N				
						V	V	V		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>

Tabelle 17. Übersicht des Imports von PC/IXF-Dateien mit Verwendung der Option FORCEIN (Forts.)

PC/IXF-SPALTEN-DATENTYP	DATENBANKSPALTENDATENTYP											
	NUMERIC					CHARACTER			GRAPH	DATETIME		
	SMALL INT	INT	BIGINT	DEC	FLT	(0,0)	(SBCS, 0) <sup>e</sup>	(SBCS, DBCS) <sup>b</sup>	<sup>b</sup>	DATE	TIME	TIME STAMP
-(SBCS, DBCS)							N mit F <sup>d</sup>	N		V <sup>c</sup>	V <sup>c</sup>	V <sup>c</sup>
						V	V mit F	V				
GRAPH (Grafik)												
						N mit F <sup>d</sup>			N			
						V			V			
DATETIME (Datum und Uhrzeit)												
-DATE										N		
										V		
-TIME											N	
											V	
-TIME STAMP												N
												V
<p><b>Anmerkung:</b> Wenn eine a PC/IXF-Spalte nur mit der Option FORCEIN in eine Datenbankspalte importiert werden kann, wird die Zeichenfolge 'mit F' zusammen mit 'N' oder 'V' gezeigt. Ein 'N' gibt an, daß das Dienstprogramm eine neue Datenbanktabelle erstellt. Ein 'V' gibt an, daß das Dienstprogramm Daten in eine vorhandene Datenbanktabelle importiert. Die Option FORCEIN wirkt sich nur auf die Kompatibilität von Zeichen- und Grafikdatentypen aus.</p> <p><sup>a</sup> Einzelne Werte werden zurückgewiesen, wenn sie nicht im Bereich für den numerischen Zieldatentyp liegen.</p> <p><sup>b</sup> Datentyp ist nur in DBCS-Umgebungen verfügbar.</p> <p><sup>c</sup> Einzelne Werte werden zurückgewiesen, wenn sie keine gültigen Datums- oder Uhrzeitwerte darstellen.</p> <p><sup>d</sup> Gilt nur, wenn der PC/IXF-Quellendatentyp nicht von der Zieldatenbank unterstützt wird.</p> <p><sup>e</sup> Datentyp ist in DBCS-Umgebungen nicht verfügbar.</p>												

### Unterschiede zwischen PC/IXF und Version 0 System /370 IXF

Im folgenden werden die Unterschiede zwischen PC/IXF, das vom Datenbankmanager verwendet wird, und Version 0 System /370 IXF, das von mehreren Host-Datenbankprodukten verwendet wird, beschrieben:

- PC/IXF-Dateien orientieren sich an ASCII und nicht an EBCDIC. PC/IXF-Dateien haben eine erheblich erweiterte Codepage-Kennzeichnung, einschließlich neuer Codepage-Kennungen im H-Datensatz und der Verwendung von Codepage-Werten in den Spaltendeskriptorsätzen. Es gibt auch einen Mechanismus zur Markierung von Spalten mit Zeichendaten als FOR BIT DATA. FOR BIT DATA-Spalten sind von besonderer Bedeutung, weil Umsetzungen, die ein PC/IXF-Dateiformat in ein anderes IXF- oder Datenbankdateiformat oder umgekehrt umsetzen, keine Codepage-Umsetzung der Werte durchführen können, die in FOR BIT DATA-Spalten enthalten sind.
- Es ist nur das Maschinendatenformat zulässig, d. h. das Feld IXFTFORM muß immer den Wert M enthalten. Weiterhin müssen die Maschinendaten in PC-Formaten vorliegen. Das bedeutet, daß das Feld IXFTMFRM den Wert PC enthalten muß. Das bedeutet, daß ganze Zahlen, Gleitkommazahlen und Dezimalzahlen in Datenteilen von PC/IXF-Datensätzen PC-Formate aufweisen müssen.
- Anwendungsdatensätze (A) sind nach dem H-Datensatz in einer PC/IXF-Datei zulässig. Sie werden nicht gezählt, wenn der Wert des Feldes IXFHH-CNT berechnet wird.
- Jeder PC/IXF-Datensatz beginnt mit einem Satzlängenanzeiger. Es handelt sich dabei um eine 6 Byte große Zeichendarstellung eines ganzzahligen Werts, die die Länge des PC/IXF-Datensatzes in Byte enthält (ohne Einschluß des Satzlängenanzeigers), d. h. die gesamte Satzlänge minus 6 Byte. Der Zweck des Satzlängenfeldes besteht darin, PC-Programmen die Erkennung von Satzbegrenzungen zu ermöglichen.
- PC/IXF unterstützt keine Version 0 IXF X-Sätze (jedoch D-Satzkennungen), um die kompakte Speicherung von Daten mit variabler Länge zu erleichtern und eine komplexe Verarbeitung zu vermeiden, wenn ein Feld in mehrere Sätze aufgeteilt ist. Wenn ein Feld mit variabler Länge oder ein Feld, das Nullwerte enthalten darf, das letzte Feld in einem D-Datensatz ist, ist es nicht nötig, die gesamte maximale Länge des Feldes in die PC/IXF-Datei zu schreiben.



## WSF (Worksheet File Format)

Die Produkte Lotus 1-2-3 und Symphony verwenden dasselbe Basisformat, wobei in jedem neuen Release zusätzliche Funktionen hinzugefügt werden. Der Datenbankmanager unterstützt eine Untergruppe der Arbeitsblatt Datensätze, die für alle Lotus-Produkte identisch sind. Dies bedeutet, daß für die vom Datenbankmanager unterstützten Releases von Lotus 1-2-3 und Symphony alle Dateinamen mit einer beliebigen Erweiterung aus drei Zeichen akzeptiert werden. Zum Beispiel: WKS, WK1, WRK, WR1, WJ2.

Jede WSF-Datei repräsentiert ein Arbeitsblatt. Der Datenbankmanager verwendet folgende Konventionen, um die Arbeitsblätter zu interpretieren und um die Konsistenz der im Rahmen der Exportoperationen generierten Arbeitsblätter zu gewährleisten:

- Zellen in der ersten Zeile (ROW-Wert 0) werden für die beschreibenden Informationen zum gesamten Arbeitsblatt reserviert. Alle Daten in dieser Zeile sind wahlfrei. Sie wird beim Import ignoriert.
- Zellen in der zweiten Zeile (ROW-Wert 1) werden für die Spaltenkennsätze verwendet.
- Die verbleibenden Zeilen sind Datenzeilen (Datensätze oder Zeilen mit Daten der Tabelle).
- Zellenwerte unter Spaltenüberschriften sind Werte für diese Spalte oder für dieses Feld.
- Ein NULL-Wert wird durch das Fehlen eines realen Datensatzes für den Zelleninhalt (z. B. ganze Zahl, Zahl, Kennsatz oder Formeldatensatz) in einer bestimmten Spalte innerhalb einer Zeile mit Datensätzen für den Zelleninhalt dargestellt.

**Anmerkung:** Eine Zeile, die nur NULL-Werte enthält, wird nicht exportiert oder importiert.

Beim Erstellen einer Datei im WSF-Format während einer Exportoperation können einige Daten verlorengehen.

WSF-Dateien verwenden eine Lotus-Codepunktzuordnung, die nicht notwendigerweise mit den Codepages übereinstimmt, die von DB2 unterstützt werden. Infolgedessen werden beim Importieren einer WSF-Datei Daten aus den Lotus-Codepunkten in die Codepunkte umgesetzt, die von der Codepage der Anwendung unterstützt werden. Beim Exportieren erfolgt die Umsetzung in umgekehrter Richtung. DB2 unterstützt die Umsetzung zwischen Lotus-Codepunkten und Codepunkten, die durch die Codepages 437, 819, 850, 860, 863 und 865 definiert sind.

**Anmerkung:** Für Benutzer von Mehrbytezeichensätzen wird keine Umsetzung ausgeführt.

## Worksheet File Format (WSF)

---

## Anhang D. Warnungen, Fehler- und Beendigungsnachrichten

Die von den verschiedenen Dienstprogrammen zum Versetzen von Daten generierten Nachrichten sind in den SQL-Nachrichten enthalten. Diese Nachrichten werden vom Datenbankmanager generiert, wenn eine Warnungs- oder Fehlerbedingung erkannt wurde. Jede Nachricht besitzt eine Nachrichten-ID, die aus einem Präfix (SQL) und einer vier- oder fünfstelligen Nachrichtennummer zusammengesetzt ist. Es gibt Informationsnachrichten, Warnungen und Nachrichten über kritische Systemfehler. Nachrichten-IDs, die mit **N** enden, sind Fehlernachrichten. IDs, die mit **W** enden, sind Warnungen oder Informationsnachrichten. Nachrichten-IDs, die mit **C** enden, weisen auf kritische Systemfehler hin.

Die Nachrichtennummer wird auch als *SQLCODE*-Wert bezeichnet. Der *SQLCODE*-Wert wird als positive oder negative Zahl an die Anwendung weitergegeben, je nach Nachrichtentyp (N, W oder C). N und C ergeben negative Werte, W ergibt positive Werte. DB2 gibt den *SQLCODE*-Wert an die Anwendung zurück, und die Anwendung kann die dem *SQLCODE*-Wert zugeordnete Nachricht abrufen. DB2 gibt auch einen *SQLSTATE*-Wert bei Bedingungen zurück, die das Ergebnis einer SQL-Anweisung sein könnten. Einigen *SQLCODE*-Werten sind *SQLSTATE*-Werte zugeordnet.

Weitere Informationen zu allen DB2-Nachrichten finden Sie im Handbuch *Fehlernachrichten*. Sie können die Informationen in diesen Buch verwenden, um einen Fehler oder ein Problem einzugrenzen und das Problem zu beheben, indem Sie die entsprechende Wiederherstellungsmaßnahme ausführen. Diese Informationen können auch verwendet werden, um sich darüber zu informieren, wo Nachrichten generiert und protokolliert werden.

SQL-Nachrichten und der Nachrichtentext, die den *SQLSTATE*-Werten zugeordnet sind, können auch über die Befehlszeile des Betriebssystems aufgerufen werden. Geben Sie folgendes an der Eingabeaufforderung des Betriebssystems ein, um die Hilfefunktion für diese Fehlernachrichten aufzurufen:

```
db2 ? SQLnnnnn
```

Dabei steht *nnnnn* für die Nachrichtennummer.

## Nachrichten

Die Nachrichten-ID, die als Parameter des Befehls **db2** akzeptiert wird, kann in beliebiger Schreibweise angegeben werden, und das letzte Zeichen ist nicht erforderlich. Demzufolge führen die folgenden Befehle alle zum gleichen Ergebnis:

```
db2 ? SQL0000N
db2 ? sq10000
db2 ? SQL0000n
```

Wenn der Nachrichtentext zu lang für den Bildschirm ist, können Sie den folgenden Befehl verwenden (auf UNIX-basierten Basisbetriebssystemen und anderen, die den Pipe-Befehl „more“ unterstützen):

```
db2 ? SQLnnnnn | more
```

Sie können die Ausgabe auch in eine Datei umleiten, die dann angezeigt werden kann.

Die Hilfe kann auch im interaktiven Eingabemodus aufgerufen werden. Geben Sie folgendes an der Eingabeaufforderung des Betriebssystems ein, um diesen Modus zu aktivieren:

```
db2
```

Geben Sie folgendes an der Eingabeaufforderung `db2 => ein`, um die Hilfe zu DB2-Nachrichten im interaktiven Eingabemodus aufzurufen:

```
? SQLnnnnn
```

Der Nachrichtentext, der SQLSTATE-Werten zugeordnet ist, kann durch Absetzen folgender Befehle abgerufen werden:

```
db2 ? nnnnn
oder
db2 ? nn
```

Dabei ist *nnnnn* ein fünfstelliger SQLSTATE-Wert (alphanumerisch) und *nn* ein zweistelliger SQLSTATE-Klassencode (die ersten beiden Stellen des SQLSTATE-Werts).

---

## Anhang E. Verwenden der DB2-Bibliothek

Die Bibliothek für DB2 Universal Database besteht aus Online-Hilfe, Handbüchern (PDF und HTML) und Beispielprogrammen in HTML-Format. Im folgenden wird beschrieben, welche Informationen bereitgestellt werden und wie Sie darauf zugreifen können.

Über **Information - Unterstützung** können Sie online auf die Produktinformationen zugreifen. Weitere Informationen finden Sie in „Zugreifen auf Informationen mit "Information - Unterstützung"“ auf Seite 332. Sie können sich im Web Informationen zu Tasks und zur Fehlerbehebung sowie DB2-Bücher, Beispielprogramme und DB2-Informationen anzeigen lassen.

---

### PDF-Dateien und gedruckte Bücher für DB2

#### Informationen zu DB2

In der folgenden Tabelle sind die DB2-Handbücher in vier Kategorien unterteilt:

#### **DB2-Benutzerhandbücher und -Referenzinformationen**

Diese Bücher enthalten die allgemeinen DB2-Informationen für alle Plattformen.

#### **DB2-Installations- und -Konfigurationsinformationen**

Diese Bücher gelten für DB2 auf einer bestimmten Plattform. So steht beispielsweise jeweils ein separates Handbuch *Einstieg* (Quick Beginnings) für DB2 für OS/2-, Windows- und UNIX-Plattformen zur Verfügung.

#### **Plattformübergreifende Beispielprogramme in HTML**

Bei diesen Beispielen handelt es sich um die HTML-Versionen der mit Application Development Client installierten Beispielprogramme. Sie dienen zur Information und können die Programme selbst nicht ersetzen.

#### **Release-Informationen**

Diese Dateien enthalten die neuesten Informationen, die in die DB2-Handbücher nicht mehr aufgenommen werden konnten.

Die Installationshandbücher, Release-Informationen und Lernprogramme können im HTML-Format direkt von der Produkt-CD-ROM angezeigt werden. Die meisten Handbücher stehen auf der Produkt-CD-ROM im HTML-Format zur Verfügung und können angezeigt werden. Auf der CD-ROM mit DB2-Veröffentlichungen stehen die Handbücher im PDF-Format zur Verfügung und können mit Adobe Acrobat angezeigt und gedruckt werden. Darüber hinaus können Sie gedruckte Veröffentlichungen bei IBM bestellen. Siehe hierzu „Bestellen der gedruckten Handbücher“ auf Seite 327. Die folgende Tabelle enthält eine Liste der Bücher, die bestellt werden können.

Auf OS/2- und Windows-Plattformen können Sie die HTML-Dateien im Verzeichnis `sql11ib\doc\html` installieren. Die DB2-Informationen werden in verschiedene Sprachen übersetzt, jedoch nicht alle Informationen in alle Sprachen. Sind bestimmte Informationen in einer Sprache nicht verfügbar, wird statt dessen die englische Version dieser Informationen zur Verfügung gestellt.

Auf UNIX-Plattformen können Sie die HTML-Dateien in mehreren Sprachen installieren, und zwar in den Unterverzeichnissen `doc/%L/html`, wobei `%L` für den Code der jeweiligen Landessprache steht. Weitere Informationen finden Sie im entsprechenden Handbuch *Einstieg*.

Es gibt verschiedene Möglichkeiten, auf DB2-Bücher und -Informationen zuzugreifen:

- „Anzeigen von Online-Informationen“ auf Seite 331
- „Suchen nach Online-Informationen“ auf Seite 336
- „Bestellen der gedruckten Handbücher“ auf Seite 327
- „Drucken der PDF-Handbücher“ auf Seite 326

Tabelle 18. Informationen zu DB2

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<b>DB2-Benutzerhandbücher und -Referenzinformationen</b>			
<i>Systemverwaltung</i>	<i>Systemverwaltung Konzept.</i> Dieses Handbuch enthält eine Übersicht über Datenbankkonzepte, Informationen zu Aspekten des Datenbankentwurfs (wie z. B. zum logischen und physischen Datenbankentwurf) sowie eine Erläuterung zu hohen Verfügbarkeit.	SC12-2879 db2d1g70  SC12-2877 db2d2g70	db2d0
	<i>Systemverwaltung Implementierung.</i> Dieses Handbuch enthält Informationen zu Implementierungsaspekten, wie beispielsweise zur Implementierung des Datenbankentwurfs, zum Zugriff auf Datenbanken sowie zu Prüfungs-, Sicherungs- und Wiederherstellungsverfahren.	SC12-2878 db2d3g70	
	<i>Systemverwaltung Optimierung.</i> Dieses Handbuch enthält Informationen zur Datenbankumgebung sowie zur Auswertung und Optimierung der Anwendungsleistung.		
	Sie können die drei Bände des Handbuchs <i>Systemverwaltung</i> in englischer Sprache in den USA und Kanada über die Formnummer SBOF-8934 bestellen.		

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>Administrative API Reference</i>	Dieses Handbuch enthält eine Beschreibung zu den DB2-Anwendungsprogrammierschnittstellen (APIs) und -Datenstrukturen, die Sie zum Verwalten Ihrer Datenbank verwenden können. Darüber hinaus wird in diesem Handbuch erläutert, wie Sie APIs von Ihren Anwendungen aus aufrufen können.	SC09-2947 db2b0e70	db2b0
<i>Application Building Guide</i>	Dieses Handbuch umfaßt Informationen zur Umgebungskonfiguration sowie Anweisungsschritte zum Kompilieren, Verbinden und Ausführen von DB2-Anwendungen auf Windows-, OS/2- und UNIX-Plattformen.	SC09-2948 db2axe70	db2ax
<i>APPC, CPI-C, and SNA Sense Codes</i>	Dieses Handbuch enthält Basisinformationen zu APPC-, CPI-DFV- und SNA-Prüfcodes, die bei der Arbeit mit DB2 Universal Database-Produkten ausgegeben werden können.	Keine Formnummer db2ape70	db2ap
	Nur im HTML-Format verfügbar.		
<i>Application Development Guide</i>	Dieses Handbuch enthält eine Erläuterung zur Entwicklung von Anwendungen, die mit Hilfe von eingebettetem SQL bzw. JAVA (JDBC und SQLJ) auf DB2-Datenbanken zugreifen. Unter anderem wird das Schreiben von gespeicherten Prozeduren, das Schreiben von benutzerdefinierten Funktionen, das Erstellen von benutzerdefinierten Typen, das Verwenden von Auslösern und das Entwickeln von Anwendungen in partitionierten Umgebungen oder mit Systemen zusammenschlossener Datenbanken beschrieben.	SC09-2949 db2a0e70	db2a0



Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>CLI Guide and Reference</i>	Dieses Handbuch erklärt die Entwicklung von Anwendungen, die für den Zugriff auf DB2-Datenbanken DB2 Call Level Interface verwenden, eine aufrufbare SQL-Schnittstelle, die mit der Microsoft-ODBC-Spezifikation kompatibel ist.	SC09-2950 db2l0e70	db2l0
<i>Command Reference</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung des Befehlszeilenprozessors und eine Beschreibung der DB2-Befehle für die Datenbankverwaltung.	SC09-2951 db2n0e70	db2n0
<i>Konnektivität Ergänzung</i>	Dieses Handbuch enthält Konfigurations- und Referenzinformationen zur Verwendung von DB2 für AS/400, DB2 für OS/390, DB2 für MVS oder DB2 für VM als DRDA-Anwendungs-Requester mit DB2 Universal Database-Servern. Darüber hinaus enthält dieses Handbuch Informationen zur Verwendung von DRDA-Anwendungs-Servern mit DB2 Connect-Anwendungs-Requestern.  Dieses Buch ist lediglich im HTML- und PDF-Format verfügbar.	Keine Form- nummer db2h1g70	db2h1
<i>Versetzen von Daten Dienstprogramme und Referenz</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung der DB2-Dienstprogramme, wie beispielsweise IMPORT, EXPORT, LOAD, AUTOLOADER und DPROF, die das Verschieben von Daten vereinfachen.	SC12-2881 db2dmg70	db2dm
<i>Data Warehouse-Zentrale Verwaltung</i>	Dieses Handbuch enthält Informationen zur Erstellung und Verwaltung eines Data Warehouse mit Hilfe der Data Warehouse-Zentrale.	SC12-2885 db2ddg70	db2dd
<i>Data Warehouse Center Application Integration Guide</i>	Dieses Handbuch enthält Informationen, die Programmierer bei der Integration von Anwendungen in die Data Warehouse-Zentrale sowie in den Information Catalog Manager unterstützen.	SC26-9994 db2ade70	db2ad

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Connect Benutzerhandbuch</i>	Dieses Handbuch enthält eine Beschreibung der Konzepte der DB2 Connect-Produkte, allgemeine Informationen zur Verwendung sowie Informationen zur Programmierung dieser Produkte.	SC12-2880 db2c0g70	db2c0
<i>DB2 Query Patroller Administration Guide</i>	Dieses Handbuch enthält eine Übersicht über den Betrieb des DB2 Query Patroller-Systems, spezifische Informationen zum Systembetrieb und zur Verwaltung sowie Task-Informationen zu den GUI-Verwaltungsdienstprogrammen.	SC09-2958 db2dwe70	db2dw
<i>DB2 Query Patroller User's Guide</i>	In diesem Handbuch wird die Verwendung der Tools und Funktionen von DB2 Query Patroller beschrieben.	SC09-2960 db2wwe70	db2ww
<i>Glossar</i>	Dieses Handbuch enthält Definitionen zu den in DB2 und den zugehörigen Komponenten verwendeten Begriffen.  Es ist im Handbuch <i>SQL Reference</i> enthalten und steht außerdem separat im HTML-Format zur Verfügung.	Keine Formnummer db2t0g70	db2t0
<i>DB2 UDB Image, Audio und Video Extender Verwaltung und Programmierung</i>	Dieses Handbuch enthält Basisinformationen zu DB2 Extender, Informationen zur Verwaltung und Konfiguration von IAV Extender sowie Informationen zur Programmierung mit Hilfe von IAV Extender. Es enthält Referenzinformationen, Diagnoseinformationen (mit Nachrichten) und Beispiele.	SC12-2892 dmbu7g70	dmbu7
<i>Information Catalog Manager Systemverwaltung</i>	Dieses Handbuch enthält eine Anleitung zur Verwaltung von Informationskatalogen.	SC12-2886 db2dig70	db2di
<i>Information Catalog Manager Programming Guide and Reference</i>	Dieses Handbuch enthält Definitionen für die Architekturschnittstellen für Information Catalog Manager.	SC26-9997 db2bie70	db2bi
<i>Information Catalog Manager Benutzerhandbuch</i>	Dieses Handbuch enthält Informationen zur Verwendung der Information Catalog Manager-Benutzerschnittstelle.	SC12-2887 db2aig70	db2ai

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Installation und Konfiguration Ergänzung</i>	Dieses Handbuch enthält Anweisungen zur Planung, Installation und Konfiguration von plattformspezifischen DB2-Clients. Darüber hinaus enthält es Informationen zu Bindevorgängen, zum Einrichten der Client/Server-Kommunikation, zu DB2-GUI-Tools, zu DRDR-AS, zur verteilten Installation, zur Konfiguration von verteilten Anforderungen sowie zum Zugriff auf heterogene Datenquellen.	GC12-2864 db2iyg70	db2iy
<i>Fehlernachrichten</i>	Dieses Handbuch enthält eine Liste der Nachrichten und Codes, die von DB2, vom Information Catalog Manager und von der Data Warehouse-Zentrale ausgegeben werden, sowie eine Beschreibung der jeweils erforderlichen Benutzeraktionen.  Sie können beide Bände des Handbuchs <i>Fehlernachrichten</i> in englischer Sprache in den USA und Kanada unter der Formnummer SBOF-8932 bestellen.	Band 1 GC12-2875 db2m1g70 Band 2 GC12-2888 db2m2g70	db2m0
<i>OLAP Integration Server Administration Guide</i>	Dieses Handbuch enthält eine Erläuterung zur Verwendung der Komponente Administration Manager von OLAP Integration Server.	SC27-0787 db2dpe70	n/v
<i>OLAP Integration Server Metaoutline User's Guide</i>	Dieses Handbuch enthält eine Erläuterung zum Erstellen und Ausfüllen von OLAP-Metastrukturen mit Hilfe der OLAP Metaoutline-Standard-schnittstelle (nicht mit Hilfe des OLAP Metaoutline Assistant).	SC27-0784 db2upe70	n/v
<i>OLAP Integration Server Model User's Guide</i>	Dieses Handbuch enthält eine Erläuterung zum Erstellen von OLAP-Modellen mit Hilfe der OLAP Model-Standard-schnittstelle (nicht mit Hilfe des OLAP Model Assistant).	SC27-0783 db2lpe70	n/v
<i>OLAP Konfiguration und Benutzerhandbuch</i>	Dieses Handbuch enthält Informationen zur Konfiguration und Einrichtung von OLAP Starter Kit.	SC12-2889 db2ipg70	db2ip

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Excel</i>	Dieses Handbuch enthält eine Beschreibung zur Verwendung des Tabellenkalkulationsprogramms Excel zum Analysieren von OLAP-Daten.	SC12-2890 db2epg70	db2ep
<i>OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Lotus 1-2-3</i>	Dieses Handbuch enthält eine Beschreibung zur Verwendung des Tabellenkalkulationsprogramms Lotus 1-2-3 zum Analysieren von OLAP-Daten.	SC12-2891 db2tpg70	db2tp
<i>Replikation Benutzer- und Referenzhandbuch</i>	Dieses Handbuch enthält Informationen zur Planung, Konfiguration, Verwaltung und Verwendung der mit DB2 gelieferten Replikations-Tools.	SC12-2884 db2e0g70	db2e0
<i>Spatial Extender Benutzer- und Referenzhandbuch</i>	Dieses Handbuch enthält Informationen zur Installation, Konfiguration, Verwaltung, Programmierung und Fehlerbehebung für den Spatial Extender. Darüber hinaus enthält es zentrale Beschreibungen räumlicher Datenkonzepte sowie spezifische Referenzinformationen (Nachrichten und SQL) für den Spatial Extender.	SC12-2894 db2sbg70	db2sb
<i>SQL Erste Schritte</i>	Dieses Handbuch enthält eine Einführung in die SQL-Konzepte sowie Beispiele für eine Reihe von Konstrukten und Tasks.	SC12-2882 db2y0g70	db2y0
<i>SQL Reference, Band 1 und Band 2</i>	Dieses Handbuch beschreibt die Syntax, die Semantik und die Regeln von SQL. Darüber hinaus enthält das Handbuch Informationen zu Inkompatibilitäten zwischen Release-Ständen, Produkt-einschränkungen und Katalogsichten.  Sie können beide Bände des Handbuchs <i>SQL Reference</i> in englischer Sprache in den USA und Kanada unter der Formnummer SBOF-8933 bestellen.	Band 1 SC09-2974 db2s1e70  Band 2 SC09-2975 db2s2e70	db2s0

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>System Monitor Guide and Reference</i>	Dieses Handbuch enthält eine Beschreibung zum Sammeln unterschiedlicher Informationen zu Datenbanken und dem Datenbankmanager. In diesem Buch wird erläutert, wie Sie mit Hilfe dieser Informationen einen Einblick in Datenbankaktivitäten erhalten, die Leistung verbessern und Fehlerursachen feststellen können.	SC09-2956 db2f0e70	db2f0
<i>Text Extender Verwaltung und Programmierung</i>	Dieses Handbuch enthält Basisinformationen zu DB2 Extender, Informationen zur Verwaltung und Konfiguration von Text Extender sowie zur Programmierung mit Hilfe von Text Extender. Es bietet Referenzinformationen, Diagnoseinformationen (mit Nachrichten) und Beispiele.	SC12-2893 desu9g70	desu9
<i>Troubleshooting Guide</i>	Dieses Handbuch hilft Ihnen bei der Bestimmung von Fehlerquellen, bei der Fehlerbehebung sowie bei der Verwendung von Diagnose-Tools, wenn Sie den DB2-Kundendienst in Anspruch nehmen.	GC09-2850 db2p0e70	db2p0
<i>Neue Funktionen</i>	Dieses Handbuch enthält eine Beschreibung der neuen Einrichtungen, Funktionen und Erweiterungen in DB2 Universal Database Version 7.	SC12-2883 db2q0g70	db2q0
<b>DB2-Installations- und -Konfigurationsinformationen</b>			
<i>DB2 Connect Enterprise Edition für OS/2 und Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration für DB2 Connect Enterprise Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2863 db2c6g70	db2c6

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Connect Enterprise Edition für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation, Konfiguration und Ausführung von Tasks für DB2 Connect Enterprise Edition auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2862 db2cyg70	db2cy
<i>DB2 Connect Personal Edition Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation, Konfiguration und Ausführung von Tasks für DB2 Connect Personal Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für alle unterstützten Clients.	GC12-2869 db2c1g70	db2c1
<i>DB2 Connect Personal Edition für Linux Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Connect Personal Edition für alle unterstützten Linux-Varianten.	GC12-2865 db2c4g70	db2c4
<i>DB2 Data Links Manager Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Konfiguration und Ausführung von Tasks für DB2 Data Links Manager unter AIX und 32-Bit-Windows-Betriebssystemen.	GC12-2868 db2z6g70	db2z6
<i>DB2 Enterprise - Extended Edition für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation und Konfiguration für DB2 Enterprise - Extended Edition auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2867 db2v3g70	db2v3

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Enterprise - Extended Edition für Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation und Konfiguration für DB2 Enterprise - Extended Edition unter 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2866 db2v6g70	db2v6
<i>DB2 für OS/2 Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration von DB2 Universal Database für das Betriebssystem OS/2. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2870 db2i2g70	db2i2
<i>DB2 für UNIX Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Migration, Installation und Konfiguration von DB2 Universal Database auf UNIX-Plattformen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2872 db2ixg70	db2ix
<i>DB2 für Windows Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database unter 32-Bit-Windows-Betriebssystemen. Darüber hinaus enthält dieses Handbuch Installations- und Konfigurationsinformationen für eine Reihe von unterstützten Clients.	GC12-2873 db2i6g70	db2i6
<i>DB2 Personal Edition Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database Personal Edition unter OS/2 und 32-Bit-Windows-Betriebssystemen.	GC12-2871 db2i1g70	db2i1
<i>DB2 Personal Edition für Linux Einstieg</i>	Dieses Handbuch enthält Informationen zur Planung, Installation, Migration und Konfiguration für DB2 Universal Database Personal Edition für alle unterstützten Linux-Varianten.	GC12-2874 db2i4g70	db2i4

Tabelle 18. Informationen zu DB2 (Forts.)

Name	Beschreibung	IBM Form PDF-Dateiname	HTML- Verzeichnis
<i>DB2 Query Patroller Installation Guide</i>	Dieses Handbuch enthält Installationsinformationen zu DB2 Query Patroller.	GC09-2959 db2iwe70	db2iw
<i>DB2 Warehouse Manager Installation</i>	Dieses Handbuch enthält Installationsinformationen für Warehouse-Agenten, Warehouse- Umsetzungsprogramme und den Information Catalog Manager.	GC12-2876 db2ide70	db2id
<b>Plattformübergreifende Beispielprogramme in HTML</b>			
Beispielprogramme in HTML	Dieses Handbuch enthält die Beispiel- programme für die Programmier- sprachen auf allen von DB2 unterstützten Plattformen im HTML- Format. Die Beispielprogramme werden lediglich zu Informationszwecken zur Verfügung gestellt. Nicht alle Beispiele sind für alle Programmiersprachen verfügbar. Die HTML-Beispiele stehen nur dann zur Verfügung, wenn der DB2 Application Development Client installiert ist.  Weitere Informationen zu den Programmen finden Sie im Handbuch <i>Application Building Guide</i> .	Keine Form- nummer	db2hs
<b>Release-Informationen</b>			
<i>DB2 Connect Release- Informationen</i>	Dieses Dokument enthält die neuesten Informationen, die in die DB2 Connect- Handbücher nicht mehr aufgenom- men werden konnten.	Siehe Anmerkung 2.	db2cr
<i>DB2 Installationsinformationen</i>	Dieses Dokument enthält die neuesten Informationen zur Installation, die in die DB2-Handbücher nicht mehr aufge- nommen werden konnten.	Nur auf der Produkt-CD- ROM verfügbar.	
<i>DB2-Release-Informationen</i>	Dieses Dokument enthält die neuesten Informationen zu allen DB2-Produkten und -Funktionen, die in die DB2- Handbücher nicht mehr aufgenommen werden konnten.	Siehe Anmerkung 2.	db2ir



### **Anmerkungen:**

1. Das Zeichen an der sechsten Stelle des Dateinamens gibt die Landessprache eines Buchs an. So kennzeichnet der Dateiname db2d0e70 die englische Version des Handbuchs *Systemverwaltung*, der Dateinamen db2d0f70 kennzeichnet die französische Version des Buchs. Folgende Buchstaben werden an der sechsten Stelle des Dateinamens verwendet, um die Landessprache für ein Handbuch anzugeben:

<b>Sprache</b>	<b>Kennung</b>
Brasilianisches Portugiesisch	b
Bulgarisch	u
Dänisch	d
Deutsch	g
Englisch	e
Finnisch	y
Französisch	f
Griechisch	a
Italienisch	i
Japanisch	j
Koreanisch	k
Niederländisch	q
Norwegisch	n
Polnisch	p
Portugiesisch	v
Russisch	r
Schwedisch	s
Slowenisch	l
Spanisch	z
Trad. Chinesisch	t
Tschechisch	x
Türkisch	m
Ungarisch	h
Vereinf. Chinesisch	c

2. Kurzfristig verfügbare Informationen, die in die DB2-Handbücher nicht mehr aufgenommen werden können, sind in den Release-Informationen enthalten, die im HTML-Format und als ASCII-Datei verfügbar sind. Die HTML-Version steht über 'Information - Unterstützung' und auf den Produkt-CD-ROMs zur Verfügung. Gehen Sie wie folgt vor, um die ASCII-Dateien anzuzeigen:
  - Rufen Sie auf UNIX-Plattformen die Datei `Release.Notes` auf. Diese Datei befindet sich im Verzeichnis `DB2DIR/Readme/%L`. Dabei ist `%L` die länderspezifische Angabe und `DB2DIR` eine der folgenden Angaben:
    - `/usr/lpp/db2_07_01` (unter AIX)
    - `/opt/IBMd2/V7.1` (unter HP-UX, PTX, Solaris und Silicon Graphics IRIX)
    - `/usr/IBMd2/V7.1` (unter Linux)
  - Rufen Sie auf anderen Plattformen die Datei `RELEASE.TXT` auf. Diese Datei befindet sich in dem Verzeichnis, in dem das Produkt installiert ist. Auf OS/2-Plattformen können Sie auch den Ordner **IBM DB2** und anschließend das Symbol **Release-Informationen** doppelt anklicken.

## Drucken der PDF-Handbücher

Wenn Sie eine gedruckte Version der Handbücher bevorzugen, können Sie die PDF-Dateien auf der CD-ROM mit DB2-Veröffentlichungen ausdrucken. Mit Adobe Acrobat Reader können Sie entweder das gesamte Handbuch oder bestimmte Teile des Handbuchs ausdrucken. Die Namen der einzelnen Handbücher in der Bibliothek finden Sie in Tabelle 18 auf Seite 315.

Die neueste Version von Adobe Acrobat Reader finden Sie auf der Adobe-Web-Site unter <http://www.adobe.com>.

Die PDF-Dateien befinden sich auf der CD-ROM mit DB2-Veröffentlichungen und haben die Dateierweiterung PDF. Führen Sie folgende Schritte aus, um auf die PDF-Dateien zuzugreifen:

1. Legen Sie die CD-ROM mit DB2-Veröffentlichungen in das CD-ROM-Laufwerk ein. Auf UNIX-Plattformen: Hängen Sie die CD-ROM mit den DB2-Veröffentlichungen an. Das Handbuch *Einstieg* enthält Anweisungen zu den Mount-Prozeduren.
2. Starten Sie Acrobat Reader.

3. Öffnen Sie die gewünschte PDF-Datei von einer der folgenden Positionen aus:

- Auf OS/2- und Windows-Plattformen:

Verzeichnis *x:\doc\sprache*. Dabei gibt *x* das CD-ROM-Laufwerk an, *sprache* den zweistelligen Landescode für die verwendete Sprache (z. B. EN für Englisch).

- Auf UNIX-Plattformen:

Verzeichnis */cdrom/doc/%L* auf der CD-ROM. Dabei gibt */cdrom* den Mount-Punkt der CD-ROM an, *%L* den Namen der gewünschten länderspezifischen Angaben.

Sie können die PDF-Dateien auch von der CD-ROM in ein lokales Laufwerk oder ein Netzlaufwerk kopieren und sie von dort aus lesen.

### **Bestellen der gedruckten Handbücher**

Sie können die gedruckten DB2-Handbücher einzeln bestellen. In den USA und Kanada ist es außerdem möglich, mehrere Bücher als Paket unter einer SBOF-Nummer zu bestellen. Setzen Sie sich mit Ihrem IBM Vertragshändler oder Vertriebsbeauftragten in Verbindung, oder bestellen Sie die Handbücher telefonisch bei IBM Direkt unter der Nummer 0180/55 090. Darüber hinaus können Sie die Handbücher über die Web-Seite mit Veröffentlichungen unter <http://www.elink.ibm.com/pbl/pbl> bestellen.

Es sind zwei Gruppen von Handbüchern verfügbar. Die Gruppe mit der Formnummer SBOF-8935 umfaßt Referenzinformationen und Informationen zur Verwendung für DB2 Warehouse Manager. Die Gruppe mit der Formnummer SBOF-8931 umfaßt Referenzinformationen und Informationen zur Verwendung für alle anderen DB2 Universal Database-Produkte und -Funktionen. Der Inhalt der SBOF-Gruppen ist in der folgenden Tabelle aufgeführt.

Tabelle 19. Bestellen der gedruckten Handbücher

SBOF-Nummer	In dieser Gruppe enthaltene Handbücher	
SBOF-8931	<ul style="list-style-type: none"> <li>• Administration Guide: Planning</li> <li>• Administration Guide: Implementation</li> <li>• Administration Guide: Performance</li> <li>• Administrative API Reference</li> <li>• Application Building Guide</li> <li>• Application Development Guide</li> <li>• CLI Guide and Reference</li> <li>• Command Reference</li> <li>• Data Movement Utilities Guide and Reference</li> <li>• Data Warehouse Center Administration Guide</li> <li>• Data Warehouse Center Application Integration Guide</li> <li>• DB2 Connect User's Guide</li> <li>• Installation and Configuration Supplement</li> <li>• Image, Audio, and Video Extenders Administration and Programming</li> <li>• Message Reference, Volumes 1 and 2</li> </ul>	<ul style="list-style-type: none"> <li>• OLAP Integration Server Administration Guide</li> <li>• OLAP Integration Server Metaoutline User's Guide</li> <li>• OLAP Integration Server Model User's Guide</li> <li>• OLAP Integration Server User's Guide</li> <li>• OLAP Setup and User's Guide</li> <li>• OLAP Spreadsheet Add-in User's Guide for Excel</li> <li>• OLAP Spreadsheet Add-in User's Guide for Lotus 1-2-3</li> <li>• Replication Guide and Reference</li> <li>• Spatial Extender Administration and Programming Guide</li> <li>• SQL Getting Started</li> <li>• SQL Reference, Volumes 1 and 2</li> <li>• System Monitor Guide and Reference</li> <li>• Text Extender Administration and Programming</li> <li>• Troubleshooting Guide</li> <li>• What's New</li> </ul>
SBOF-8935	<ul style="list-style-type: none"> <li>• Information Catalog Manager Administration Guide</li> <li>• Information Catalog Manager User's Guide</li> <li>• Information Catalog Manager Programming Guide and Reference</li> </ul>	<ul style="list-style-type: none"> <li>• Query Patroller Administration Guide</li> <li>• Query Patroller User's Guide</li> </ul>

### Zugreifen auf die Online-Hilfefunktion

Die Online-Hilfefunktion ist für alle DB2-Komponenten verfügbar. In der folgenden Tabelle werden die verschiedenen Hilfearten beschrieben.

Hilfearten	Inhalt	Zugriff
<i>Hilfe für Befehl</i>	Erklärt die Syntax von Befehlen im Befehlszeilenprozessor.	Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:  ? <i>befehl</i>  Dabei stellt <i>befehl</i> ein Schlüsselwort bzw. den vollständigen Befehl dar.  So kann beispielsweise durch die Eingabe von ? catalog Hilfe für alle CATALOG-Befehle angezeigt werden, während mit ? catalog database lediglich Hilfe für den Befehl CATALOG DATABASE angezeigt wird.
<b>Hilfe für Client-Konfiguration - Unterstützung</b>	Erläutert die Tasks, die Sie in einem Fenster oder Notizbuch ausführen können. Die Hilfe umfaßt Übersichtsinformationen und unbedingt erforderliche Informationen sowie eine Beschreibung zur Verwendung der Steuerelemente im Fenster oder Notizbuch.	Klicken Sie in einem Fenster oder in einem Notizbuch den Druckknopf <b>Hilfe</b> an oder drücken Sie die Taste <b>F1</b> .
<i>Hilfe für die Befehlszentrale</i>		
<i>Hilfe für die Steuerzentrale</i>		
<i>Hilfe für die Data Warehouse-Zentrale</i>		
<b>Hilfe für Event Analyzer</b>		
<i>Hilfe für Information Catalog Manager</i>		
<i>Hilfe für die Satellitenverwaltungszentrale</i>		
<i>Hilfe für die Prozedurenzentrale</i>		

---

Hilfearten	Inhalt	Zugriff
<i>Nachrichtenhilfe</i>	Beschreibt die Ursache von Nachrichten sowie die auszuführenden Benutzeraktionen.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:</p> <pre>? XXXnnnnn</pre> <p>Dabei ist <i>XXXnnnnn</i> eine gültige Nachrichtenennung.</p> <p>Bei Eingabe von ? SQL30081 wird z. B. die Hilfe zur Nachricht SQL30081 angezeigt.</p> <p>Wenn Sie die Nachrichtenhilfe seitenweise anzeigen möchten, geben Sie den folgenden Befehl ein:</p> <pre>? XXXnnnnn   more</pre> <p>Geben Sie folgenden Befehl ein, um die Nachrichtenhilfe in einer Datei zu speichern:</p> <pre>? XXXnnnnn &gt; datei.erw</pre> <p>Dabei ist <i>datei.erw</i> die Datei, in der Sie die Nachrichtenhilfe speichern möchten.</p>
<i>Hilfe für SQL</i>	Erklärt die Syntax von SQL-Anweisungen.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:</p> <pre>help anweisung</pre> <p>Dabei gibt <i>anweisung</i> eine SQL-Anweisung an.</p> <p>So kann beispielsweise durch die Eingabe von <code>help SELECT</code> die Hilfe zur Anweisung <code>SELECT</code> angezeigt werden.</p> <p><b>Anmerkung:</b> Die Hilfe für SQL ist auf UNIX-Plattformen nicht verfügbar.</p>
<i>SQLSTATE-Hilfe</i>	Erklärt SQLSTATE-Werte und SQL-Klassencodes.	<p>Geben Sie im interaktiven Modus des Befehlszeilenprozessors folgendes ein:</p> <pre>? sqlstate oder ? klassencode</pre> <p>Datei ist <i>sqlstate</i> ein gültiger, fünfstelliger SQL-Status, und <i>klassencode</i> stellt die ersten zwei Ziffern des SQL-Statuswerts dar.</p> <p>So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, während mit ? 08 Hilfe für den Klassencode 08 angezeigt wird.</p>

## Anzeigen von Online-Informationen

Die zum Lieferumfang dieses Produkts gehörenden Handbücher werden als Softcopy im HTML-Format (HTML - Hypertext Markup Language) bereitgestellt. In einer Softcopy können Sie die Informationen auf einfache Art suchen und anzeigen und über Hypertextverbindungen auf zugehörige Informationen zugreifen. Außerdem wird die gemeinsame Nutzung der Bibliothek in Ihrem gesamten Unternehmen erleichtert.

Sie können die Online-Bücher und Beispielprogramme mit jedem Browser anzeigen, der den Spezifikationen von HTML Version 3.2 entspricht.

Führen Sie die nachfolgend beschriebenen Schritte aus, um Online-Bücher oder Beispielprogramme anzuzeigen:

- Wenn Sie DB2-Verwaltungs-Tools ausführen, verwenden Sie **Information - Unterstützung**.
- Klicken Sie in einem Browser **Datei**—>**Seite öffnen** an. Die geöffnete Seite enthält eine Übersicht über die DB2-Informationen und Verbindungen (Links) zu diesen Informationen:

- Öffnen Sie auf UNIX-Plattformen die folgende Seite:

```
INSTHOME/sql11ib/doc/%L/html/index.htm
```

Dabei ist %L die länderspezifische Angabe.

- Öffnen Sie auf anderen Plattformen die folgende Seite:

```
sql11ib\doc\html\index.htm
```

Der Pfad befindet sich auf dem Laufwerk, auf dem DB2 installiert ist.

Wenn Sie **Information - Unterstützung** nicht installiert haben, können Sie die Seite öffnen, indem Sie das Symbol **DB2-Informationen** doppelt anklicken. Je nach verwendetem Betriebssystem befindet sich das Symbol im Hauptproduktordner bzw. unter Windows im Menü **Start**.

### Installieren des Netscape-Browsers

Wenn Sie nicht bereits einen Web-Browser installiert haben, können Sie Netscape von der im Lieferumfang des Produkts enthaltenen Netscape-CD-ROM aus installieren. Führen Sie folgende Schritte aus, um ausführliche Informationen zur Installation zu erhalten:

1. Legen Sie die Netscape-CD-ROM ein.
2. Nur auf UNIX-Plattformen: Hängen Sie die CD-ROM an. Das Handbuch *Einstieg* enthält Anweisungen zu den Mount-Prozeduren.
3. Installationsanweisungen finden Sie in der Datei `CDNAVnn.txt`. Dabei ist *nn* die zweistellige Landeskennung. Die Datei befindet sich im Stammverzeichnis der CD-ROM.

**Zugreifen auf Informationen mit "Information - Unterstützung"**  
**Information - Unterstützung** ermöglicht Ihnen den schnellen Zugriff auf DB2-Produktinformationen. **Information - Unterstützung** ist auf allen Plattformen mit DB2-Verwaltungs-Tools verfügbar.

Sie können 'Information - Unterstützung' öffnen, indem Sie das entsprechende Symbol doppelt anklicken. Abhängig vom verwendeten System befindet sich das Symbol im Hauptproduktordner im Ordner 'Information' bzw. unter Windows im Menü **Start**.

Sie können auf 'Information - Unterstützung' auch zugreifen, indem Sie die Funktionsleiste und das Menü **Hilfe** auf der DB2-Windows-Plattform verwenden.

Unter 'Information - Unterstützung' finden Sie sechs verschiedene Arten von Informationen. Klicken Sie die entsprechende Indexzunge an, um die für diese Informationsart verfügbaren Themen aufzurufen.

**Funktionen** Die Hauptfunktionen, die Sie mit DB2 ausführen können.

**Referenz** DB2-Referenzinformationen, wie beispielsweise Schlüsselwörter, Befehle und APIs.

**Handbücher** DB2-Handbücher.

**Fehlerbehebung**

Kategorien von Fehlermeldungen sowie die entsprechenden Benutzeraktionen.

**Beispielprogramme**

Beispielprogramme, die in DB2 Application Development Client enthalten sind. Wenn Sie DB2 Application Development Client nicht installiert haben, wird diese Indexzunge nicht angezeigt.

**Web** DB2-Informationen im World Wide Web. Sie müssen über Ihr System eine Verbindung zum Web herstellen können, um auf diese Informationen zugreifen zu können.

Wenn Sie einen Eintrag aus einer der Listen auswählen, startet **Information - Unterstützung** eine Funktion zum Anzeigen der Informationen. Bei der Anzeigefunktion kann es sich abhängig von der ausgewählten Informationsart um die Hilfanzeige des Systems, einen Editor oder einen Web-Browser handeln.

In 'Information - Unterstützung' steht eine Suchfunktion zur Verfügung, mit der Sie nach einem bestimmten Thema suchen können, ohne in den Listen blättern zu müssen.



Rufen Sie über die Hypertextverbindung in 'Information - Unterstützung' das Suchformular **In DB2-Online-Informationen suchen** auf.

Der HTML-Such-Server wird normalerweise automatisch gestartet. Wenn eine Suche in HTML-Informationen fehlschlägt, müssen Sie möglicherweise mit einer der nachfolgend aufgeführten Methoden den Such-Server starten:

#### Unter Windows

Klicken Sie **Start** an und wählen Sie **Programme** —> **IBM DB2** —> **Informationen** —> **HTML-Such-Server starten** aus.

#### Unter OS/2

Klicken Sie den Ordner **DB2 für OS/2** und anschließend das Symbol für **HTML-Such-Server starten** doppelt an.

Falls andere Probleme bei der Suche in HTML-Informationen auftreten, finden Sie möglicherweise entsprechende Hinweise in den Release-Informationen.

**Anmerkung:** Die Suchfunktion steht in Linux-, PTX- und Silicon Graphics IRIX-Umgebungen nicht zur Verfügung.

### Verwenden der DB2-Assistenten

Assistenten unterstützen Sie bei der Ausführung bestimmter Verwaltungsaufgaben, indem sie Sie Schritt für Schritt durch jede Aufgabe führen. Assistenten stehen über die Steuerzentrale und 'Client-Konfiguration - Unterstützung' zur Verfügung. In der folgenden Tabelle sind die einzelnen Assistenten und deren Verwendungszweck aufgeführt.

**Anmerkung:** In Umgebungen mit partitionierten Datenbanken sind die Assistenten **Datenbank erstellen**, **Index erstellen**, **Aktualisierung auf mehreren Systemen konfigurieren** und **Leistungskonfiguration** verfügbar.

Assistent	Verwendung	Zugriff
<i>Datenbank hinzufügen</i>	Katalogisieren einer Datenbank auf einer Client-Workstation.	Klicken Sie in <b>Client-Konfiguration - Unterstützung</b> die Option <b>Hinzufügen</b> an.
<i>Datenbank sichern</i>	Festlegen, Erstellen und Terminieren eines Sicherungsplans.	Klicken Sie in der Steuerzentrale die zu sichernde Datenbank mit der rechten Maustaste an und wählen Sie <b>Sichern</b> —> <b>Datenbank mit Assistent</b> aus.
<i>Aktualisierung auf mehreren Systemen konfigurieren</i>	Konfigurieren einer Aktualisierung auf mehreren Systemen, einer verteilten Transaktion oder einer zweiphasigen Festschreibung.	Klicken Sie in der Steuerzentrale den Ordner <b>Datenbanken</b> mit der rechten Maustaste an und wählen Sie <b>Aktualisierung auf mehreren Systemen</b> aus.

<b>Assistent</b>	<b>Verwendung</b>	<b>Zugriff</b>
<i>Datenbank erstellen</i>	Erstellen einer Datenbank und Ausführen einiger grundlegender Konfigurationsfunktionen.	Klicken Sie in der Steuerzentrale den Ordner <b>Datenbanken</b> mit der rechten Maustaste an und wählen Sie <b>Erstellen</b> —> <b>Datenbank mit Assistent</b> aus.
<i>Tabelle erstellen</i>	Auswählen eines Basisdatentyps und Erstellen eines Primärschlüssels für die Tabelle.	Klicken Sie in der Steuerzentrale das Symbol <b>Tabellen</b> mit der rechten Maustaste an und wählen Sie <b>Erstellen</b> —> <b>Tabelle mit Assistent</b> aus.
<i>Tabellenbereich erstellen</i>	Erstellen eines neuen Tabellenbereichs.	Klicken Sie in der Steuerzentrale das Symbol <b>Tabellenbereiche</b> mit der rechten Maustaste an und wählen Sie <b>Erstellen</b> —> <b>Tabellenbereich mit Assistent</b> aus.
<i>Index erstellen</i>	Hinweise zum Erstellen und Löschen von Indizes für Ihre Abfragen.	Klicken Sie in der Steuerzentrale das Symbol <b>Index</b> mit der rechten Maustaste an und wählen Sie <b>Erstellen</b> —> <b>Index mit Assistent</b> aus.
<i>Leistungskonfiguration</i>	Optimieren der Leistung einer Datenbank durch Aktualisieren der Konfigurationsparameter, so daß sie den Anforderungen Ihres Unternehmens entsprechen.	Klicken Sie in der Steuerzentrale die Datenbank, die optimiert werden soll, mit der rechten Maustaste an und wählen Sie <b>Leistung mit Assistent konfigurieren</b> aus.  Klicken Sie in einer Umgebung mit partitionierten Datenbanken in der Sicht für Datenbankpartitionen die erste Datenbankpartition, die optimiert werden soll, mit der rechten Maustaste an und wählen Sie <b>Leistung mit Assistent konfigurieren</b> aus.
<i>Datenbank wiederherstellen</i>	Wiederherstellen einer Datenbank nach einem Fehler. Dieser Assistent hilft Ihnen, zu entscheiden, welche Sicherungskopie Sie verwenden und welche Protokolle Sie erneut abarbeiten.	Klicken Sie in der Steuerzentrale die Datenbank, die wiederhergestellt werden soll, mit der rechten Maustaste an und wählen Sie <b>Wiederherstellen</b> —> <b>Datenbank mit Assistent</b> aus.

## Einrichten eines Dokument-Servers

Die DB2-Informationen werden standardmäßig auf Ihrem lokalen System installiert. Das bedeutet, daß alle Benutzer, die Zugriff auf DB2-Informationen benötigen, dieselben Dateien installieren müssen. Führen Sie folgende Schritte aus, um die DB2-Informationen an einer einzigen Position zu speichern:

1. Kopieren Sie alle Dateien und Unterverzeichnisse aus dem Verzeichnis `\sql11ib\doc\html` Ihres lokalen Systems auf einen Web-Server. Jedem Handbuch ist ein Unterverzeichnis zugeordnet, das alle erforderlichen HTML- und GIF-Dateien enthält, aus denen das Handbuch besteht. Stellen Sie sicher, daß die Verzeichnisstruktur erhalten bleibt.
2. Konfigurieren Sie den Web-Server so, daß er die Dateien an der neuen Speicherposition sucht. Informationen hierzu finden Sie im Anhang zu NetQuestion im Handbuch *DB2 Installation und Konfiguration Ergänzung*.
3. Wenn Sie die Java-Version von **Information - Unterstützung** verwenden, können Sie eine Basis-URL-Adresse für alle HTML-Dateien angeben. Sie sollten die URL-Adresse für das Bücherverzeichnis verwenden.
4. Wenn Sie die Buchdateien anzeigen können, ist es möglich, bei häufig aufgerufenen Themen Lesezeichen zu setzen. Es empfiehlt sich, folgende Seiten mit einem Lesezeichen zu versehen:
  - Bücherverzeichnis
  - Inhaltsverzeichnis häufig verwendeter Handbücher
  - Themen, auf die häufig verwiesen wird, wie beispielsweise zum Ändern von Tabellen
  - Suchformular

Informationen dazu, wie Sie die DB2 Universal Database-Online-Dokumentationsdateien auf einer zentralen Maschine zur Verfügung stellen können, finden Sie im Anhang zu NetQuestion im Handbuch *DB2 Installation und Konfiguration Ergänzung*.

## Suchen nach Online-Informationen

Verwenden Sie eine der folgenden Methoden, um nach Informationen in den HTML-Dateien zu suchen:

- Klicken Sie im obersten Rahmen auf **Suchen**. Verwenden Sie das Suchformular, um nach einem bestimmten Thema zu suchen. Diese Funktion steht in Linux-, PIX- oder Silicon Graphics IRIX-Umgebungen nicht zur Verfügung.
- Klicken Sie im obersten Rahmen auf **Index**. Mit Hilfe des Indexes können Sie nach einem bestimmten Thema im Buch suchen.
- Rufen Sie das Inhaltsverzeichnis oder den Index der Hilfe oder des HTML-Buchs auf und verwenden Sie die Suchfunktion des Web-Browsers, um nach einem bestimmten Thema im Buch zu suchen.
- Mit Hilfe der Lesezeichenfunktion des Web-Browsers können Sie schnell zu einem bestimmten Thema zurückkehren.
- Mit Hilfe der Suchfunktion von **Information - Unterstützung** können Sie bestimmte Themen suchen. Weitere Informationen finden Sie in „Zugreifen auf Informationen mit "Information - Unterstützung"“ auf Seite 332.

---

## Anhang F. Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, daß nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. Anstelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France, zu richten. Anfragen an obige Adresse müssen auf englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Web-Sites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Web-Sites dar. Das über diese Web-Sites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Web-Sites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne daß eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited  
Office of the Lab Director  
1150 Eglinton Ave. East  
North York, Ontario  
M3C 1H7  
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, daß diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

#### COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. \_Jahr/Jahre angeben\_. Alle Rechte vorbehalten.

---

## **Neue deutsche Rechtschreibung**

Durch die Einführung der neuen deutschen Rechtschreibung bei IBM zum 1. September 1999 kann es vorkommen, dass in dem vorliegenden Handbuch bestimmte Wörter sowohl nach der alten als auch nach der neuen Schreibweise verwendet werden, und zwar immer dann, wenn auf existierende Handbuchkapitel und/oder Programmteile zurückgegriffen wird.

---

## **Änderungen in der IBM Terminologie**

Die ständige Weiterentwicklung der deutschen Sprache nimmt auch Einfluss auf die IBM Terminologie. Durch die daraus resultierende Umstellung der IBM Terminologie kann es u. U. vorkommen, dass in diesem Handbuch sowohl alte als auch neue Termini gleichbedeutend verwendet werden. Dies ist der Fall, wenn auf ältere existierende Handbuchkapitel und/oder Programmteile zurückgegriffen wird.

Aufgrund kurzfristiger Änderungen der Software, die in die Dokumentation nicht mehr aufgenommen werden konnten, entsprechen die in den Handbüchern aufgeführten Programmelemente möglicherweise nicht den im eigentlichen Programm angezeigten Elementen.



---

## Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
IBM System AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RS/6000
DataPropagator	IBM System /370
DataRefresher	SP
DB2	SQL/DS
DB2 Connect	SQL/400
DB2 Extenders	System/370
DB2 OLAP Server	IBM System /390
DB2 Universal Database	SystemView
Distributed Relational Database Architecture	VisualAge
DRDA	VM/ESA
eNetwork	VSE/ESA
Extended Services	VTAM
FFST	WebExplorer
First Failure Support Technology	WIN-OS/2

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen:

Microsoft, Windows und Windows NT sind Marken oder eingetragene Marken von Microsoft Corporation.

Java und alle auf Java basierenden Marken und Logos sowie Solaris sind in gewissen Ländern Marken von Sun Microsystems, Inc.

Tivoli und NetView sind in gewissen Ländern Marken von Tivoli Systems Inc.

UNIX ist eine eingetragene Marke und wird ausschließlich von der X/Open Company Limited lizenziert.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

---

# Index

## A

Aktualisierung auf mehreren Systemen konfigurieren, Assistent 333  
Anwendungsdatensatz,  
  PC/IXF 266  
anyorder 140  
Anzeigen  
  Online-Informationen 331  
ASC, als Importdateityp 44  
ASC, Datei  
  Beispiel 249  
  Format 248  
ASC, Datentypbeschreibungen 250  
ASC-Datei, Beispiel 249  
ASCII-Format mit universellen Zeilenbegrenzern, Dateiformat 248  
ASCII-Format ohne universelle Zeilenbegrenzer (DEL)  
  Versetzen von Daten zwischen Plattformen 210  
ASCII-Format ohne universelle Zeilenbegrenzer (DEL), Dateiformat 240  
Assistent  
  Datenbank wiederherstellen 334  
Assistenten  
  Aktualisierung auf mehreren Systemen konfigurieren 333  
  Assistenten 333  
  Datenbank erstellen 333  
  Datenbank hinzufügen 333, 334  
  Datenbank sichern 333  
  Index 334  
  Leistungskonfiguration 334  
  Tabelle erstellen 334  
  Tabellenbereich erstellen 334  
  Tasks ausführen 333  
Ausnahmetabelle  
  Dienstprogramm LOAD 157  
AutoLoader  
  Einschränkungen 199  
  Fehlerbehebung 199  
  Rahmenbedingungen 199  
  Übersicht 181  
zur Verwendung erforderliche Berechtigungen und Zugriffsrechte 182

## B

Beendigungsnachrichten 311  
Beendigungssatz, PC/IXF 274  
Befehlssyntax  
  interpretieren 231  
Begrenzer  
  Zeichenfolge 242  
Beispiel  
  FORCEIN 300  
Beispielprogramme  
  HTML 324  
  plattformübergreifend 324  
Benutzerdefinierte einzigartige Datentypen (UDTs)  
  Dienstprogramm IMPORT 40  
Berechtigungen  
  für Dienstprogramm AutoLoader 182  
  für Dienstprogramm EXPORT 3  
  für Dienstprogramm IMPORT 31  
  für Dienstprogramm LOAD 89  
Bezugswert  
  Satzlänge 254  
binarynumerics 149

## C

chardel 19, 69, 152  
codepage 145  
Codepage-Umsetzung  
  beim Importieren oder Laden von PC/IXF-Daten 294  
  Dateien 294  
coldel 19, 70, 152  
compound 63

## D

Data Warehouse-Zentrale  
  Übersicht 228  
dateformat 65, 145  
Dateiformate  
  ASC 248  
  DEL (ASCII-Format ohne universelle Zeilenbegrenzer) 240  
  PC/IXF 253  
  WSF 309  
zum Exportieren von Tabelle in Datei 10  
zum Importieren von Datei in Tabelle 47

Dateitypwerte  
  Dienstprogramm EXPORT 19  
  Dienstprogramm IMPORT 63  
  Dienstprogramm LOAD 140  
Datenbank  
  Warehouse 228  
  wiederherstellbar und nicht wiederherstellbar, Optionen des Dienstprogramms LOAD zur Verwendung mit 84  
Datenbank erstellen, Assistent 333  
Datenbank hinzufügen, Assistent 333, 334  
Datenbank sichern, Assistent 333  
Datensatz, PC/IXF 264  
Datentypbeschreibungen  
  ASC 250  
  DEL 244  
  PC/IXF 285  
Datentypen  
  PC/IXF 277  
Datenübertragung  
  AutoLoader 181  
  zwischen Host und Workstation 217  
  zwischen Plattformen 209  
datesiso 19, 70, 152  
DB2-Bibliothek  
  Assistenten 333  
  Dokument-Server einrichten 335  
  Drucken von PDF-Handbüchern 326  
  gedruckte Handbücher bestellen 327  
  Handbücher 313  
  Information - Unterstützung 332  
  neueste Informationen 326  
  Online-Hilfefunktion 329  
  Online-Informationen anzeigen 331  
  Online-Informationen suchen 336  
  Sprachenennung für Bücher 325  
  Struktur 313  
DB2 Data Links Manager  
  Dienstprogramm EXPORT 201  
  Dienstprogramm IMPORT 206  
  Dienstprogramm LOAD 207

- DB2 Data Links Manager (*Forts.*)
    - Exportieren zwischen Exemplaren 204
    - Fehlerbehebung für das Dienstprogramm LOAD 207
  - db2LoadQuery - Load Query 135
  - DB2LOADREC 98
  - db2move 212
  - decplusblank 19, 70, 152
  - decpt 19, 70, 152
  - DEL, Datei
    - Beispiel 242
    - Format 240
  - DEL-Datei, Beispiel 242
  - DEL-Datentypbeschreibungen 244
  - delprioritychar 71, 152
  - Dienstprogrammdateiformate 239
  - dldel 19, 71, 153
  - Dokument-Server einrichten 335
  - Drucken von PDF-Handbüchern 326
  - dumpfile 146
  - DWC (Data Warehouse-Zentrale) 228
- E**
- Erneutes Erstellen einer exportierten Tabelle
    - Dienstprogramm EXPORT 4
    - Dienstprogramm IMPORT 38
    - in einer IXF-Datei gespeicherte Tabellenattribute 38
    - Tabellenattribute, die nicht in einer IXF-Datei gespeichert werden 38
  - EXPORT, Dienstprogramm
    - Datenübertragung zwischen Host und Workstation 217
    - DB2 Data Links Manager 201
    - erneutes Erstellen einer exportierten Tabelle 4
    - große Objekte (LOBS) 5
    - Identitätsspalten 4
    - Paralleles Exportieren mit b2batch 5
    - Rahmenbedingungen 27
    - Übersicht 2
    - zur Verwendung erforderliche Berechtigungen und Zugriffsrechte 3
  - EXPORT, Dienstprogramm, Dateiformate 239
  - EXPORT, Nachrichtendateien 27, 78, 180
  - Exportieren
    - Angaben von Spaltennamen 13
- Exportieren (*Forts.*)
    - Dateitypwerte für 19
  - Exportoperationen
    - durch Data Warehouse-Zentrale unterstützte 228
- F**
- fastparse 140
  - Fehlernachrichten 311
  - forcein 72, 153
  - FORCEIN
    - allgemeine Semantik 298
    - Beispiel 300
    - Codepage-Semantik 299
    - Datentypsemantik 305
    - Option 298
    - Übersicht über Import von PC/IXF-Dateien mit 306
  - Fortsetzungssatz, PC/IXF 273
- G**
- Geänderte Syntax und Funktionsweise
    - Dienstprogramm LOAD 86
  - generatedignore 63, 141
  - generatedmissing 63, 141
  - generatedoverride 141
  - Generierte Spalte 36, 93
  - Gepufferte INSERT-Operationen
    - Dienstprogramm IMPORT 34
  - Gespeicherte Prozeduren
    - Umsetzungsprogramm 229
  - Große Objekte (LOBS)
    - Dienstprogramm EXPORT 5
    - Dienstprogramm IMPORT 39
  - Gültiger PC/IXF-Datentyp 284
- H**
- Handbücher 313, 327
  - Hash-Algorithmus 182
  - Hierarchiesatz, PC/IXF 269
  - Hinweise zu Codepages
    - Dienstprogramm IMPORT 74
    - Dienstprogramm LOAD 160
  - HTML
    - Beispielprogramme 324
- I**
- IBM Relational Data Replication Tools
    - Komponenten 227
    - Übersicht 225
  - Identitätssatz, PC/IXF 275
  - Identitätsspalte 4, 34, 91
  - identityignore 64, 142
  - identitymissing 64, 142
  - identityoverride 142
  - implieddecimal 66, 146
  - Import
    - von PC/IXF-Dateien, mit FORCEIN 306
  - IMPORT, Dienstprogramm
    - benutzerdefinierte einzigartige Datentypen (UDTs) 40
    - Client/Server 33
    - Datenübertragung zwischen Host und Workstation 217
    - DB2 Data Links Manager 206
    - Einschränkungen 78
    - erneutes Erstellen einer exportierten Tabelle 38
    - ferne Datenbank 33
    - generierte Spalten 36
    - gepufferte INSERT-Operationen 34
    - große Objekte (LOBS) 39
    - Hinweise zu Codepages 74
    - Identitätsspalten 34
    - im Vergleich zum Dienstprogramm LOAD 235
    - Leistung 77
    - Optimieren der Leistung 77
    - Rahmenbedingungen 78
    - Sperren von Tabellen 40
    - Übersicht 30
    - zur Verwendung erforderliche Berechtigungen und Zugriffsrechte 31
  - IMPORT, Dienstprogramm, Dateiformate 239
  - IMPORT, Nachrichtendateien 27, 78, 180
  - Importieren
    - Dateitypwerte für 63
    - PC/IXF-Datei in Tabelle 42
  - Importieren von PC/IXF-Dateien
    - allgemeine Regeln 292
    - datentypspezifische Regeln 294
  - Index, Assistent 334
  - indexfreespace 143
  - indexixf 72
  - Indexsatz, PC/IXF 267
  - indexschema 72
  - Information - Unterstützung 332
  - Inkompatible Spalten 293
  - Installation
    - Netscape-Browser 331
  - Integration Exchange Format (IXF) 253

## K

keepblanks 71, 153  
Kennsatz, PC/IXF 255

## L

Laden

Dateitypwerte für 140

Laden von Daten

AutoLoader, Dienstprogramm  
zum Laden von Daten in  
Datenbankpartitionen 181

Ladeoperationen

durch Data Warehouse-Zentrale  
unterstützte 228

Leistung

Dienstprogramm IMPORT 77  
Dienstprogramm LOAD 174

Leistungskonfiguration, Assi-  
stent 334

LOAD

temporäre Dateien 113

LOAD, Dienstprogramm

Ausnahmetabelle 157

BUILD-Phase 81

Datenbankwiederherstellung 84

DB2 Data Links Manager 207

DELETE-Phase 81

Einschränkungen 180

geänderte Syntax und Funktions-  
weise 86

generierte Spalten 93

Hinweise zu Codepages 160

Identitätsspalten 91

im Vergleich zum Dienst-  
programm IMPORT 235

LOAD-Phase 80

Optimieren der Leistung 174

Parallelität 88

Protokollsätze 159

Rahmenbedingungen 180

Speicherausgangsdatei 158

temporäre Dateien 158

Übersicht 80

Übersicht über Prozeß 80

Wiederherstellung nach Feh-  
ler 98

zur Verwendung erforderliche  
Berechtigungen und Zugriffs-  
rechte 89

LOAD, Dienstprogramm, Datei-  
formate 239

LOAD, Nachrichtendateien 27, 78,  
180

LOAD DELETE START COMPEN-  
SATION, Protokollsatz 159

LOAD PENDING LIST, Protokoll-  
satz 159

LOAD QUERY 116

LOAD QUERY

(db2LoadQuery) 135

LOAD START, Protokollsatz 159

lobsinfile 19, 64, 143

## N

Nachrichten 311

Nachrichtendateien: EXPORT,  
IMPORT und LOAD 27, 78, 180

Netscape-Browser

Installation 331

Neueste Informationen 326

Nicht als Identitätsspalte generierte  
Spalte 36, 93

Nicht wiederherstellbare Datenbank

Optionen des Dienstprogramms  
LOAD zur Verwendung

mit 84

no\_type\_id 64

nochecklengths 68, 72, 150, 154

nodefaults 64

nodoubledel 20, 71, 153

noeofchar 66, 149

noheader 143

norowwarnings 143

nullindchar 68, 150

## O

Online-Hilfefunktion 329

Online-Informationen

anzeigen 331

suchen 336

Option

FORCEIN 298

## P

packeddecimal 150

pagefreespace 144

Paralleles Exportieren

mit db2batch 5

Parallelität

Dienstprogramm LOAD 88

Parameter

Syntax für 231

Partitionierung von Daten

AutoLoader 181

Partitionierungsschlüssel 181

PC/IXF

Codepage-

Umsetzungsdateien 294

Datentypbeschreibungen 285

Datentypen 277

gültiger Datentyp 284

PC/IXF (*Forts.*)

im Gegensatz zu System /370

IXF 308

Satzarten 253, 255

ungültige Spaltenwerte 293

ungültiger Datentyp 284, 292

PC/IXF, Datei

Format 253

PC/IXF, Satzart

Anwendung 266

Beendigung 274

Daten 264

Fortsetzung 273

Hierarchie 269

Identität 275

Index 267

Kennsatz 255

Spaltendeskriptor 259

Tabelle 257

untergeordnete Tabelle 271

PC/IXF-Datei importieren

allgemeine Regeln 292

datentypspezifische Regeln 294

mit FORCEIN 306

PC/IXF-Dateiformat

Versetzen von Daten zwischen

Plattformen 209

PC-Version des IXF (PC/IXF)-

Dateiformats 253

PDF 326

Protokollsätze

Dienstprogramm LOAD 159

## R

reclen 68, 151

Regeln für den Import von PC/IXF-  
Dateien 292, 294

Release-Informationen 326

Replikation

durch Data Warehouse-Zentrale  
unterstützte Replikationsar-  
ten 228

## S

Satz für untergeordnete Tabelle,  
PC/IXF 271

Satzart, PC/IXF

Anwendung 266

Beendigung 274

Daten 264

Fortsetzung 273

Hierarchie 269

Identität 275

Index 267

Kennsatz 255

Satzart, PC/IXF (Forts.)  
 Spaltendeskriptor 259  
 Tabelle 257  
 untergeordnete Tabelle 271

Satzarten  
 PC/IXF 253, 255

Satzlängenanzeiger 254

Schlüsselwörter  
 Syntax für 231

SELECT, Anweisung  
 im EXPORT-Befehl 11

Semantik  
 FORCEIN, allgemein 298  
 FORCEIN, Codepage 299  
 FORCEIN, Datentyp 305

Spalte  
 Angeben zum Importieren 53

Spalten, inkompatible 293

Spaltendeskriptorsatz, PC/IXF 259

Spaltenwerte, ungültige 293

Speicherauszugsdatei  
 Dienstprogramm LOAD 158

Sperrn  
 Dienstprogramm IMPORT 40

Sprachenkennung  
 Handbücher 325

SQL-Nachrichten 311

SQL-UExPT-OUT, Struktur 18

SQLCODE 311

SQLSTATE 311

SQLUIMPT-IN, Struktur 60

SQLUIMPT-OUT, Struktur 61

SQLULOAD-IN, Struktur 128

SQLULOAD-OUT, Struktur 133

Statusangaben  
 Laden anstehend 173  
 Löschen anstehend 173  
 Sicherung anstehend 173  
 Überprüfung anstehend 173

Statusangaben für anstehende Aktien 173

striptblanks 69, 151

striptnulls 69, 151

Struktur  
 ASC-Datei 248  
 DEL-Datei 240

Suche  
 Online-Informationen 336

Suchen  
 Online-Informationen 333

Syntaxdiagramm 231

System /370IXF 308  
 im Gegensatz zu PC/IXF 308

## T

Tabelle  
 exportierte Tabelle erneut erstellen 38

Tabelle erstellen, Assistent 334

Tabellenbereich erstellen, Assistent 334

Tabellensatz, PC/IXF 257

TABLE LOAD DELETE START, Protokollsatz 159

Teilen von Daten 181

Temporäre Dateien  
 Dienstprogramm LOAD 113, 158

timeformat 66, 147

timestampformat 67, 148

Tool zum Versetzen von Daten 212

totalreespace 144

Traversierfolge 30  
 benutzerdefiniert 222  
 Standard 221  
 typisierte Tabellen 221

Typisierte Tabellen  
 Auswahl beim Versetzen von Daten 222  
 Dienstprogramm EXPORT 219  
 Dienstprogramm IMPORT 219  
 Traversierfolge 30, 221  
 Versetzen von Daten, Beispiele 223  
 Versetzen von Daten zwischen 219

## U

Überprüfung auf Integritätsbedingungen 96

Übersichtstabelle  
 Importbeschränkung 78

Umgebungsvariablen  
 DB2LOADREC 98

Umsetzungsprogramm für gespeicherte Prozeduren 229

Ungültige PC/IXF-Spaltenwerte 293

Ungültiger PC/IXF-Datentyp 284

Unterschiede zwischen PC/IXF und System /370 IXF 308

usedefaults 65, 144

## V

Variablen  
 Syntax für 231

Versetzen von Daten  
 zwischen Plattformen 209

## W

Warnungen 311

Werte, Dateityp  
 für Dienstprogramm EXPORT 19  
 für Dienstprogramm IMPORT 63  
 für Dienstprogramm LOAD 140

Wiederherstellbare Datenbank  
 Optionen des Dienstprogramms LOAD zur Verwendung mit 84

Wiederherstellen, Assistent 334

Worksheet File Format (WSF) 309

WSF, Datei  
 Format 309

WSF-Dateiformat  
 Versetzen von Daten zwischen Plattformen 211

## Z

Zeichenfolgebegrenzer 242

zonedecimal 151

Zugriffsrechte  
 für Dienstprogramm AutoLoader 182  
 für Dienstprogramm EXPORT 3  
 für Dienstprogramm IMPORT 31  
 für Dienstprogramm LOAD 89

---

## Kontaktaufnahme mit IBM

Bei technischen Problemen lesen Sie bitte die entsprechenden Korrekturmaßnahmen im Handbuch *Troubleshooting Guide* und führen Sie diese aus, bevor Sie sich mit der IBM Kundenunterstützung in Verbindung setzen. Mit Hilfe dieses Handbuchs können Sie Informationen sammeln, die die DB2-Kundenunterstützung zu Fehlerbehebung verwenden kann.

Wenn Sie weitere Informationen benötigen oder eines der DB2 Universal Database-Produkte bestellen möchten, setzen Sie sich mit einem IBM Ansprechpartner in einer lokalen Geschäftsstelle oder einem IBM Software-Vertriebspartner in Verbindung.

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Über 0190/772 243 erreichen Sie die DB2 Helpline, wo Sie Antworten zu DB2-spezifischen Problemen erhalten.

---

## Produktinformationen

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180/55 090 können Sie Handbücher telefonisch bestellen.

**<http://www.ibm.com/software/data/>**

Auf den DB2-World Wide Web-Seiten erhalten Sie aktuelle DB2-Informationen wie Neuigkeiten, Produktbeschreibungen, Schulungspläne und vieles mehr.

**<http://www.ibm.com/software/data/db2/library/>**

Mit **DB2 Product and Service Technical Library** können Sie auf häufig gestellte Fragen, Berichtigungen, Handbücher und aktuelle technische DB2-Informationen zugreifen.

**Anmerkung:** Diese Informationen stehen möglicherweise nur auf Englisch zur Verfügung.

**<http://www.elink.ibm.com/pbl/pbl/>**

Auf der Web-Site für die Bestellung internationaler Veröffentlichungen (International Publications) finden Sie Informationen zum Bestellverfahren.

**<http://www.ibm.com/education/certify/>**

Das 'Professional Certification Program' auf der IBM Web-Site stellt Zertifizierungstestinformationen für eine Reihe von IBM Produkten, u. a. auch DB2, zur Verfügung.

**<ftp://software.ibm.com>**

Melden Sie sich als *anonymous* an. Im Verzeichnis /ps/products/db2 finden Sie Demo-Versionen, Berichtigungen, Informationen und Tools zu DB2 und vielen zugehörigen Produkten.

**<comp.databases.ibm-db2>, <bit.listserv.db2-1>**

Über diese Internet-Newsgroups können DB2-Benutzer Ihre Erfahrungen mit den DB2-Produkten austauschen.

**Für CompuServe: GO IBMDB2**

Geben Sie diesen Befehl ein, um auf IBM DB2 Family Forums zuzugreifen. Alle DB2-Produkte werden über diese Foren unterstützt.

In Anhang A des Handbuchs *IBM Software Support Handbook* finden Sie Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können. Rufen Sie die folgende Web-Seite auf, um auf dieses Dokument zuzugreifen:

<http://www.ibm.com/support/>. Wählen Sie anschließend die Verbindung zum IBM Software Support Handbook am unteren Rand der Seite aus.

**Anmerkung:** In einigen Ländern sollten sich die IBM Vertragshändler an die innerhalb ihrer Händlerstruktur vorgesehene Unterstützung wenden, nicht an die IBM Unterstützungsfunktion.



---

# Antwort

**IBM DB2 Universal Database**  
**Versetzen von Daten Dienstprogramme und Referenz**  
**Version 7**

**IBM Form SC12-2881-00**

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

**Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 01803/31 32 33) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.**

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: [comment@tcvn.vnet.ibm.com](mailto:comment@tcvn.vnet.ibm.com)

---

Name

---

Adresse

---

Firma oder Organisation

---

Rufnummer

---

E-Mail-Adresse

**Antwort**  
SC12-2881-00



IBM Deutschland Informationssysteme GmbH  
SW NLS Center

70548 Stuttgart





Printed in Ireland

SC12-2881-00

