

IBM DB2 Universal Database



Release-Informationen

Version 7.2/Version 7.1 FixPak 3

IBM DB2 Universal Database



Release-Informationen

Version 7.2/Version 7.1 FixPak 3

Inhaltsverzeichnis

Willkommen bei DB2 Universal Database Version 7! xiii

Teil 1. Besondere Hinweise 1

Besondere Hinweise 3

Bessere Zugriffsmöglichkeiten von DB2 UDB Version 7	3
Tastatureingabe und Navigation	3
Bessere Zugriffsmöglichkeiten für Anzeige	3
Alternative Alert-Hinweise	4
Kompatibilität mit Unterstützungstechnologien	4
Dokumentationsformate mit besseren Zugriffsmöglichkeiten	4
Zusätzliche Solaris-Programmkorrekturstufe erforderlich	4
Von DB2 Version 7 für Solaris unterstützte CPUs	4
Probleme beim Hinzufügen von Knoten zu einer partitionierten Datenbank	4
Fehler bei der Migration	6
Chinese Locale Fix on Red Flag Linux	6
DB2-Installation kann blockieren, wenn ein Wechselwerk nicht angeschlossen ist	7
Additional Locale Setting for DB2 for Linux in a Japanese and Simplified Chinese Linux Environment	7
Problem der Steuerzentrale bei Microsoft Internet Explorer	7
Inkompatibilität zwischen Information Catalog Manager und Sybase in der Windows-Umgebung	8
Verlust der Steuerzentralenfunktion.	8
Netscape-CD wird nicht mit DB2 UDB ausgeliefert	8
Fehler in den XML-Readme-Dateien	9
Möglicher Datenverlust unter Linux für S/390	9
DB2 UDB unter Windows 2000	9

Onlinedokumentation (HTML, PDF und Suche) 11

Unterstützte Web-Browser unter Windows 2000	11
Suchen nach DB2-Onlineinformationen unter Solaris	11
Einrichten von NetQuestion für OS/2 zur Verwendung von TCP/IP	12
Fehlernachrichten beim Versuch, Netscape zu starten	12
Konfigurationsanforderung für Adobe Acrobat Reader auf UNIX-basierten Systemen	13
SQL-Referenz in einer PDF-Datei verfügbar	13

Teil 2. Installation und Konfiguration 15

Allgemeine Installationsinformationen 17

Herunterladen von Installationspaketen für alle unterstützten DB2-Clients	17
---	----

Installieren von DB2 unter Windows 2000	17
Migrationsproblem bei mit Sonderregistern definierten Sichten	17
IPX/SPX-Protokollunterstützung unter Windows 2000	17
Stoppen von DB2-Prozessen vor dem Upgrade älterer Versionen von DB2.	18
Ausführen von db2iupdt nach der Installation von DB2, wenn ein anderes DB2-Produkt bereits installiert ist	18
Definieren der Linux-Umgebung zur Ausführung der DB2-Steuerzentrale	18
DB2 Universal Database Enterprise Edition und DB2 Connect Enterprise Edition für Linux unter S/390	19
DB2 Universal Database Enterprise - Extended Edition für UNIX Einstieg.	20
Kernel-Parameter "shmseg" für HP-UX	20
Migrieren von IBM Visual Warehouse-Steuerungsdatenbanken	20
Zugreifen auf Warehouse-Steuerungsdatenbanken	20

Data Links Manager Einstieg 21

Fehlschlagen von "dlfm start" mit der Nachricht, dass beim Abruf von afsfid für Präfix ein Fehler aufgetreten ist	21
Einrichten einer Tivoli Storage Manager-Verwaltungsklasse für Archivierungsdateien	21
Erforderlicher Plattenspeicherplatz für DFS Client Enabler.	21
Überwachen der Back-End-Prozesse von Data Links File Manager unter AIX	21
Installation und Konfiguration von DB2 Data Links Manager für AIX: Zusätzliche Installationsvoraussetzungen in DCE-DFS-Umgebungen.	22
Fehlgeschlagener Befehl "dlfm add_prefix"	22
Installation und Konfiguration von DB2 Data Links Manager für AIX: Installieren von DB2 Data Links Manager unter AIX mit dem Dienstprogramm db2setup	23
Installation und Konfiguration von DB2 Data Links Manager für AIX: Tasks nach Abschluss der Installation in DCE-DFS.	23
Installation und Konfiguration von DB2 Data Links Manager für AIX: Manuelles Installieren von DB2 Data Links Manager mit SMIT	24
Installation und Konfiguration von DB2 Data Links DFS Client Enabler	24
Installation und Konfiguration von DB2 Data Links Manager für Solaris	24
Auswählen einer Sicherungsmethode für DB2 Data Links Manager unter AIX.	24
Auswählen einer Sicherungsmethode für DB2 Data Links Manager in der Solaris-Betriebsumgebung	26
Auswählen einer Sicherungsmethode für DB2 Data Links Manager unter Windows NT	27

Sichern von JFS (Journaled File System) unter AIX	29
Administratorgruppenzugriffsrechte in Data Links unter Windows NT	30
Minimieren der Protokollierung für DLFF-Installation (Data Links File System Filter)	30
Protokollieren von Nachrichten nach der Installation	31
Deinstallation von DLFM-Komponenten mit SMIT entfernt unter Umständen zusätzliche Dateigruppen	32
Vorbereitung/Feststellen des Host-Namens	32
Arbeiten mit Data Links File Manager: Bereinigung nach dem Löschen von DB2 Data Links Manager aus einer DB2-Datenbank	33
DLFM1001E (Neue Fehlermeldung)	34
DLFM-Konfigurationsdateioption	34
Fehler beim Ausführen von des Data Links/DFS-Scripts "dmapp_prestart" unter AIX	34
Integration von Tivoli Space Manager in Data Links	35
Einschränkungen und Begrenzungen	36
Kapitel 4. Installieren und Konfigurieren von DB2 Data Links Manager für AIX	37
Allgemeine Installationsvoraussetzungen	37

Installation und Konfiguration: Ergänzung. 39

Kapitel 5. Installieren von DB2-Clients auf UNIX-Betriebssystemen	39
Kernel-Konfigurationsparameter für HP-UX	39
Kapitel 12. Ausführen Ihrer eigenen Anwendungen	39
Binden von Datenbankdienstprogrammen mit Run-Time Client	39
Client-Zugriff auf DB2 mit ODBC unter UNIX	39
Kapitel 24. Einrichten eines Systems zusammengesetzter Datenbanken für den Zugriff auf Einstellen eines Systems auf mehrere Datenquellen	39
Systeme zusammengesetzter Datenbanken	39
Installieren von DB2 Relational Connect	40
Kapitel 26. Zugriff auf Oracle-Datenquellen	42
Fehler in der Dokumentation	43
Zugriff auf Sybase-Datenquellen (neues Kapitel)	44
Hinzufügen von Sybase-Datenquellen zu einem Server mit zusammengesetzten Datenbanken	44
Angaben von Sybase-Codepages	50
Zugriff auf Microsoft SQL Server-Datenquellen mit ODBC (neues Kapitel)	51
Hinzufügen von Microsoft SQL Server-Datenquellen zu einem Server mit zusammengesetzten Datenbanken	51
Übersicht über Microsoft SQL Server-Codepages	56

Teil 3. Verwaltung 59

Systemverwaltung: Konzept 63

Kapitel 8. Entwerfen der physischen Datenbank	63
Partitionierungsschlüssel	63
Entwerfen von Knotengruppen	63
Kapitel 9. Entwerfen verteilter Datenbanken	63
Aktualisieren mehrerer Datenbanken	63
Kapitel 13. Hohe Verfügbarkeit in der Windows NT-Umgebung	64

Vor Ausführung von Dienstprogramm DB2MCS	
Warmstart der Maschine erforderlich	64
Kapitel 14. DB2 und hohe Verfügbarkeit unter Sun Cluster 2.2	64
Veritas-Unterstützung unter Solaris	64
Anhang B. Namenskonventionen	66
Hinweise zu Benutzer-IDs und Schemennamen mit mehr als 8 Zeichen	66
Benutzer-IDs und Kennwörter	66
Anhang D. Inkompatibilitäten zwischen Releases	66
Windows NT DLFS inkompatibel mit Norton Utilities	66
SET CONSTRAINTS ersetzt durch SET INTEGRITY	67
Anhang E. Unterstützung von Landessprachen (NLS)	67
Landessprachenversionen von DB2 Version 7	67
Einstellung der Länderangaben für den DB2-Verwaltungs-Server	70
DB2 UDB Supports the Baltic Rim Code Page (MS-1257) on Windows Platforms	71
Ableiten der Werte von Codepages	71
Unterstützung von Landescodes und Codepages	71
Zeichensätze	71

Systemverwaltung: Implementierung 73

Hinzufügen oder Erweitern von DMS-Behältern (Neuer Prozess)	73
Kapitel 1. Verwalten von DB2 mit den Tools für die grafische Benutzerschnittstelle	73
Kapitel 3. Erstellen einer Datenbank	74
Erstellen eines Tabellenbereichs	74
Erstellen einer Sequenz	75
Vergleichen von IDENTITY-Spalten und Sequenzen	77
Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation	77
Kapitel 4. Ändern einer Datenbank	78
Hinzufügen eines Behälters zu einem SMS-Tabellenbereich in einer Partition	78
Ändern einer IDENTITY-Spalte	78
Ändern einer Sequenz	79
Löschen einer Sequenz	79
Wechseln des Status eines Tabellenbereichs	80
Modifizieren von Behältern in einem DMS-Tabellenbereich	80
Kapitel 5. Steuern des Datenbankzugriffs	80
Sequenzzugriffsrechte	80
Datenverschlüsselung	80
Kapitel 8. Wiederherstellen einer Datenbank	82
Verwendung der zurückgestellten Ein-/Ausgabe	82
Teilsicherung und -wiederherstellung	84
Parallele Wiederherstellung	86
Sicherung auf benannte Pipes	87
Sicherung von einem geteilten Image	87
Protokollarchivierung nach Bedarf	89
Protokollspiegelung	90
Unterstützung von plattformübergreifendem Sichern und Wiederherstellen unter Sun Solaris und HP	91

Überlegungen zum DB2 Data Links Manager/Überlegungen zum Dienstprogramm BACKUP	91
Überlegungen zum DB2 Data Links Manager/Überlegungen zu den Dienstprogrammen RESTORE und ROLLFORWARD.	92
Wiederherstellen von Datenbanken von einer Offlinesicherung ohne aktualisierendes Wiederherstellen	94
Wiederherstellen von Datenbanken und Tabellenbereichen und aktualisierendes Wiederherstellen bis zum Ende der Protokolle.	95
DB2 Data Links Manager und Interaktionen bei Wiederherstellung	95
Erkennen von Situationen, die Abstimmung erfordern	100
Anhang C. Benutzer-Exit zur Datenbankwiederherstellung	101
Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionsserver	102
Anhang I. Knotenübergreifende Hochgeschwindigkeitskommunikation	102
Aktivieren von DB2 zur Ausführung mit VI	102

Systemverwaltung: Optimierung 105

Kapitel 3. Überlegungen zu Anwendungen	105
Angaben der Isolationsstufe	105
Anpassen der Optimierungsklasse	105
Dynamische Compound-Anweisungen	106
Kapitel 4. Überlegungen zur Umgebung	107
Größere Indexschlüssel	107
Kapitel 5. Systemkatalogstatistiken	107
Erfassen und Verwenden von Verteilungsstatistiken	107
Regeln zur Aktualisierung von Katalogstatistiken	108
Subelementstatistik	108
Kapitel 6. Der SQL-Compiler	111
Replizierte Übersichtstabellen	111
Datenzugriffskonzepte und Optimierung	113
Kapitel 8. Leistung bei der Ausführung.	114
Verwalten des Datenbankpufferpools	114
Verwalten mehrerer Datenbankpufferpools	115
Kapitel 9. Verwenden von Governor.	116
Kapitel 13. Konfigurieren von DB2	116
Zwischenspeicher für Sortierlisten (sortheap)	116
Schwellenwert für Sortierspeicher (sheapthres)	116
Maximale Anzahl Sperren pro Anwendung (maxlocks)	117
Konfigurieren von DB2/DB2 Data Links Manager/Data Links-Token-Verfallsintervall (dl_expint)	117
Datenbankkonfigurationsparameter MIN_DEC_DIV_3	118
Maximaler Zwischenspeicher für Anwendungssteuerung (app_ctl_heap_sz)	119
Zwischenspeicher für den Datenbanksystemmonitor (mon_heap_sz)	119
Maximale Anzahl aktiver Anwendungen (max-appls)	119

Vor dem bedingten Prüfpunkt zu schreibende Protokollsätze (softmax)	119
Geänderte Seiten protokollieren (trackmod)	120
Datenbankprotokollpfad ändern (newlogpath)	120
Pfad zu Protokolldateien (logpath)	121
Maximaler Speicher für Sperrenliste (locklist)	121
Anhang A. DB2-Registrierungsvariablen und DB2-Umgebungsvariablen	122
Tabelle der neuen und geänderten Registrierungsvariablen	122
Anhang C. EXPLAIN-Programme (SQL)	127

Administering Satellites Guide and Reference 129

Setting up Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition as Satellites.	129
Prerequisites.	129
Configuring the Version 7.2 System for Synchronization	131
Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System	132
Upgrading a Version 6 Steuerzentrale and Satellitenverwaltungszentrale	139

Command Reference 141

db2batch - Benchmark Tool.	141
db2cap (new command)	141
db2cap - CLI/ODBC Static Package Binding Tool	141
db2ckrst (new command)	143
db2ckrst - Check Incremental Restore Image Sequence	143
db2gncol (new command)	145
db2gncol - Update Generated Column Values	145
db2inidb - Initialize a Mirrored Database	146
db2look - DB2 Statistics Extraction Tool	147
db2updv7 - Update Database to Version 7 Current Fix Level	147
New Command Line Processor Option (-x, Suppress printing of column headings)	148
True Type Font Requirement for DB2 CLP.	148
ADD DATALINKS MANAGER	149
ARCHIVE LOG (new command)	149
Archive Log	149
BACKUP DATABASE	151
Syntax Diagram	151
DB2 Data Links Manager Considerations	152
BIND	152
CALL	153
DROP DATALINKS MANAGER (new command)	154
DROP DATALINKS MANAGER	154
EXPORT	157
GET DATABASE CONFIGURATION	158
GET ROUTINE (new command)	159
GET ROUTINE.	159
GET SNAPSHOT	159
IMPORT	160
LIST HISTORY	160
LOAD.	162
PING (new command)	163

PING	163
PUT ROUTINE (new command)	165
PUT ROUTINE	165
RECONCILE	166
REORGANIZE TABLE	166
RESTORE DATABASE	167
Syntax	167
DB2 Data Links Manager Considerations	168
ROLLFORWARD DATABASE	168
Documentation Error in CLP Return Codes	168

Versetzen von Daten Dienstprogramme und Referenz 169

Kapitel 2. Import	169
Verwenden von IMPORT mit gepufferten INSERT-Operationen	169
Kapitel 3. Load	169
Statusangaben für anstehende Aktionen nach einer Ladeoperation	169
LOAD-Rahmenbedingungen und -Einschränkungen	169
Dateitypänderungswert "totalreespace"	169
Kapitel 4. AutoLoader	170
"rexecd" erforderlich für Ausführung von Auto-loader bei aktivierter Authentifizierung.	170

Replikation Referenzhandbuch 171

Replikations-Server und Nicht-IBM Server.	171
Replikation unter Windows 2000	171
Bekannter Fehler beim Speichern von SQL-Dateien	171
DB2-Verwaltung	171
Dienstprogramm "Data Difference" im Internet	171
Kapitel 3. Beispielszenario für die Datenreplikation	171
Replikationsszenarios.	171
Kapitel 5. Planung der Replikationsumgebung	173
Tabellen- und Spaltennamen	173
DATALINK-Replikation	173
LOB-Einschränkungen	173
Planen der Replikation	173
Kapitel 6. Einrichten der Replikationsumgebung	174
Vorbedingung für beliebige Tabellenreplikation	174
Einrichten Ihrer Replikationsumgebung	174
Kapitel 8. Fehlerbestimmung	174
Kapitel 9. Capture und Apply für AS/400	176
Kapitel 10. Capture und Apply für OS/390	176
Vorbedingungen für DB2 DataPropagator für OS/390	176
UNICODE- und ASCII-Schemen für Codeumsetzung unter OS/390	177
Kapitel 11. Capture und Apply für UNIX-Plattformen	179
Einstellen der Umgebungsvariablen für Capture und Apply unter UNIX und Windows	179
Kapitel 14. Tabellenstrukturen	179
Kapitel 15. Nachrichten der Programme Capture und Apply	179
Anhang A. Starten der Programme Capture und Apply aus einer Anwendung heraus	180

System Monitor Guide and Reference 181

db2ConvMonStream	181
----------------------------	-----

Troubleshooting Guide 183

Starting DB2 on Windows 95, Windows 98, and Windows ME When the User Is Not Logged On.	183
Chapter 2. Troubleshooting the DB2 Universal Database Server	186

Verwenden von DB2 Universal Database auf 64-Bit-Plattformen 187

Kapitel 5. Konfiguration	187
LOCKLIST	187
shmsys:shminfo_shmmax	187
Kapitel 6. Einschränkungen.	187

XML Extender Verwaltung und Programmierung 189

MQSeries 191

Installation und Konfiguration für die DB2	
MQSeries-Funktionen	191
Installieren von MQSeries	191
Installieren von MQSeries AMI	192
Aktivieren der DB2 MQSeries-Funktionen.	192
MQSeries - Nachrichtendarstellung	193
Nachrichtenstruktur	193
MQSeries - Funktionsübersicht	194
Einschränkungen	195
Fehlercodes	195
Einsatzszenarios	195
Basis-Nachrichtenübertragung.	195
Senden von Nachrichten.	197
Nachrichten abrufen	197
Anwendungsübergreifende Konnektivität	199
enable_MQFunctions	204
enable_MQFunctions	204
disable_MQFunctions.	206
disable_MQFunctions.	206

Teil 4. Verwaltungs-Tools 207

Steuerzentrale 209

Fähigkeit zur Verwaltung von DB2-Servern für VSE und VM	209
Java 1.2-Unterstützung für die Steuerzentrale.	209
Verknüpfungsfehler bei der Verwendung der Onlinehilfefunktion im Windows-Betriebssystem	210
Java-Steuerzentrale unter OS/2	210
Fehler "Dateizugriff verweigert" beim Versuch, einen beendeten Job im Journal im Windows-Betriebssystem anzuzeigen	210
Testverbindung für Aktualisierung auf mehreren Systemen.	210
Steuerzentrale für DB2 für OS/390	210
Erforderliche Korrektur für die Steuerzentrale für OS/390	212
Änderung am Dialogfenster "Räumlichen Verweis erstellen"	212

Informationen zur Fehlerbehebung für die DB2-Steuerzentrale	213
Fehlerbehebung der Steuerzentrale auf UNIX-basierten Systemen	213
Mögliches Problem der Kurzhilfe bei OS/2	213
Hilfe für den Konfigurationsparameter "jdk11_path".	213
Solaris-Systemfehler (SQL10012N) bei der Verwendung der Prozedurzentrale oder des Journals.	214
Hilfe für die Datei DPREPL.DFT	214
Starten mehrerer Steuerzentralen-Applets	214
Onlinehilfefunktion für die Steuerzentrale bei der Ausführung als Applet	214
Ausführen der Steuerzentrale im Applet-Modus (Windows 95)	215
Arbeiten mit großen Abfrageergebnissen	215

Information - Unterstützung	217
Verknüpfungsfehler im Windows-Betriebssystem	217
Öffnen externer Web-Links in Netscape Navigator, wenn Netscape bereits geöffnet ist (UNIX-basierte Systeme)	217
Probleme beim Starten von "Information - Unterstützung".	217

Assistenten	219
Einstellen der Speicherbereichsgröße in "Assistent: Datenbank erstellen"	219
Assistent "MQSeries Assist"	219
Assistent "OLE DB Assist"	219

Teil 5. Informationsmanagement 221

Lernprogramm für das Informationsmanagement.	225
Überarbeitetes Lernprogramm für das Informationsmanagement	225

Data Warehouse-Zentrale Verwaltung 227	227
Fehlerbehebung	227
Definieren von Excel als Warehouse-Quelle	227
Definieren und Ausführen von Prozessen	227
Dialog "Metadaten exportieren"	227
Definieren von Werten für das Programm "OS/390-JCL-Jobstrom übergeben (VWPMVS)"	227
Änderungen des Anhangs mit der Data Warehouse-Beispieldatenbank	228
Nachrichten der Data Warehouse-Zentrale.	228
Erstellen einer Modellstruktur und Laden von Daten in den DB2 OLAP Integration Server	228
Verwenden von Classic Connect mit der Data Warehouse-Zentrale	228
Umgebungsdatenstruktur der Data Warehouse-Zentrale	229
Verwenden des Umsetzungsprogramms "Daten vertauschen"	229
Zugreifen auf Daten von DB2 Version 5 mit dem Warehouse-Agenten von DB2 Version 7.	229
Migrieren von Servern von DB2 Version 5.	229

Ändern der Agentenkonfiguration	230
Das Programm "ERwin MetaData Extract".	231
Inhalt	231
Softwarevoraussetzungen	232
Programmdateien	232
Erstellen von Befehlssprachendatei	233
Importieren einer Befehlssprachendatei in die Data Warehouse-Zentrale	234
Importieren einer Befehlssprachendatei in Information Catalog Manager	235
Fehlerbehebung	236
Zuordnung von ERwin zur DB2 Data Warehouse-Zentrale	238
Namens- und Adressbereinigung in der Data Warehouse-Zentrale	241
Schreiben einer Trillium Batch System-JCL-Datei	246
Schreiben einer Trillium Batch System-Script-Datei unter UNIX und Windows	247
Definieren eines Trillium Batch System-Schritts	247
Verwenden des benutzerdefinierten Programms von Trillium Batch System	247
Fehlerbehandlung	249
Integration von MQSeries in die Data Warehouse-Zentrale	251
Erstellen von Sichten für MQSeries-Nachrichten	251
Importieren von MQSeries-Nachrichten und XML-Metadaten	252
Unterstützung von Microsoft OLE DB und Data Transaction Services	255
Erstellen von Sichten für OLE DB-Tabellenfunktionen	256
Erstellen von Sichten für DTS-Pakete	256
Verwenden von inkrementellen COMMIT-Operationen mit Ersetzen	257
Komponenten-Trace-Datendateinamen	257
Open Client für Sybase-Quellen unter AIX und der Solaris-Betriebsumgebung erforderlich	257
Korrigierte Beispieleinträge.	257
Kapitel 3. Definieren von Warehouse-Quellen.	258
Zuordnen des Memofields in Microsoft Access zu einer Warehouse-Quelle	258
Kapitel 10. Verwalten der Warehouse-Datenbank	258
Verknüpfen von Tabellen mit einem Schrittsubtyp für das Programm DB2 UDB RUNSTATS.	258
Standard-Warehouse-Steuerungsdatenbank	258
Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung"	258
Ändern der aktiven Warehouse-Steuerungsdatenbank.	259
Erstellen und Initialisieren einer Warehouse-Steuerungsdatenbank.	259
Erstellen von Momentaufnahmen für SQL-Schritte	260
Ändern von Quellen und Zielen im Fenster "Prozessmodell"	260
Hinzufügen von Beschreibungen zu Objekten der Data Warehouse-Zentrale	260
Ausführen von Beispieinhalt	260
Bearbeiten einer SQL-Anweisung zur DDL-Erstellung	261
Migrieren von Geschäftssichten von Visual Warehouse	261

Generieren von Zieltabellen und Primärschlüsseln	261
Verwenden von Merant ODBC-Treibern	262
Neuer ODBC-Treiber	262
Definieren einer Warehouse-Quelle oder eines Warehouse-Ziels in einer OS/2-Datenbank	262
Überwachen des Status der Warehouse-Steuerungsdatenbank	262
Verwenden von SQL Assist mit der Beispieldatenbank TBC_MD	262
Verwenden der Funktion FormatDate	262
Ändern der Spracheinstellung	263
Verwenden des Umsetzungsprogramms "Schlüssel-tabelle generieren"	263
Aufrechterhalten der Verbindungen zu Datenbanken	263
Definieren eines fernen Clients der Data Warehouse-Zentrale	263
Definieren einer Warehouse-Quelle von DB2 für VM	264
Definieren einer Zieltabelle von DB2 für VM oder DB2 für VSE	264
Aktivieren der Unterstützung für begrenzte Bezeichner	264
DataJoiner-Fehler weist auf ein Bindeproblem hin	264
Einrichten und Ausführen von Replikation mit der Data Warehouse-Zentrale	265
Tipps zur Fehlerbehebung	268
Zugreifen auf Quellen und Ziele	268
Ergänzungen zu unterstützten Datenbankquellen von Fremdherstellern	270
Manuelles Erstellen einer Datenquelle in der Data Warehouse-Zentrale	272
Importieren und Exportieren von Metadaten mit CWMI (Common Warehouse Metadata Interchange)	272
Einführung	272
Importieren von Metadaten	273
Aktualisieren der Metadaten nach dem Ausführen des Importdienstprogramms	275
Exportieren von Metadaten	275
Schritt des OS/390-Dienstprogramms Runstats	277
Schritt des OS/390-Dienstprogramms Load	277
XML-Unterstützung für Common Warehouse Metamodel (CWM)	278
Prozesserstellungsprogramm	278
Schemamodellierungsprogramm	279
Musseingabefelder	279
Erweiterungen des Assistenten für die Data Warehouse-Zentrale	279
Ausgeben von Schrittinformationen in eine Datei	279

Data Warehouse Center Application Integration Guide. 281

Additional metadata templates	281
Commit.tag	281
ForeignKey.tag	282
ForeignKeyAdditional.tag	284
PrimaryKey.tag	286
PrimaryKeyAdditional.tag	287

Data Warehouse-Zentrale - Onlinehilfefunktion 291

Definieren von Tabellen oder Sichten zur Replikation	291
Ausführen von Essbase VWP's mit dem AS/400-Agenten	291
Verwenden der Fenster "Veröffentlichen von Metadaten der Data Warehouse-Zentrale" und dem zugehörigen Merkmalfenster	291
Fremdschlüssel	291
Replikationsnotizbücher	291
Importieren einer Befehlssprachendatei	291
Links zum Hinzufügen von Daten	291
Importieren von Tabellen	292
Korrektur der Onlinehilfefunktion zu RUNSTATS und REORGANIZE TABLE	292
Seite "Hinweis" (Anzeige der Warehouse-Merkmale und Notizbuch "Zeitplan")	292
Das Agentenmodulfeld im Notizbuch "Agenten-Site"	292

DB2 OLAP Starter Kit 293

OLAP Server-Website	293
Unterstützte Betriebssystemservicestufen	293
Abschließen der Installation von DB2 OLAP Starter Kit unter UNIX	293
Konfigurieren von ODBC für OLAP Starter Kit	294
Konfigurieren von Datenquellen auf UNIX-Systemen	294
Konfigurieren des OLAP-Metadatenkatalogs auf UNIX-Systemen	296
Konfigurieren von Datenquellen auf Windows-Systemen	296
Konfigurieren des OLAP-Metadatenkatalog auf Windows-Systemen	297
Vorgehensweise nach dem Konfigurieren einer Datenquelle	299
Anmelden über OLAP Starter Kit Desktop	299
Beispiel für Starter Kit-Anmeldung	300
Manuelles Erstellen und Konfigurieren der Beispieldatenbanken für OLAP Starter Kit	300
Migrieren von Anwendungen nach OLAP Starter Kit Version 7.2	301
Bekannte Probleme und Einschränkungen	302
Fehlende EQD-Dateien des OLAP-Tabellenkalkulations-Add-In	304

Information Catalog Manager Systemverwaltung 305

Initialisierungsprogramm von Information Catalog Manager	305
Lizenzfragen	305
Installationsfragen	305
Zugreifen auf Informationskataloge von DB2 Version 5 mit Information Catalog Manager von DB2 Version 7	306
Einrichten eines Informationskatalogs	307
Austauschen von Metadaten mit anderen Produkten	307

Austauschen von Metadaten mit dem Befehl "flgnxoln"	308
Austauschen von Metadata mit dem Befehl "MDIS-DGC"	308
Aufrufen von Programmen.	308

Information Catalog Manager Programming Guide and Reference	309
Information Catalog Manager Reason Codes	309

Information Catalog Manager Benutzerhandbuch	311
---	------------

Information Catalog Manager: Online-nachrichten	313
Nachricht FLG0260E	313
Nachricht FLG0051E	313
Nachricht FLG0003E	313
Nachricht FLG0372E	313
Nachricht FLG0615E	313

Information Catalog Manager: Online-hilfefunktion	315
Information Catalog Manager für das Web	315

DB2 Warehouse Manager Installation	317
Softwarevoraussetzungen für Warehouse-Umsetzungsprogramme	317
Connector für SAP R/3	317
Vorbedingungen zur Installation	317
Connector für das Web	317
Vorbedingungen zur Installation	317

Query Patroller Administration Guide	319
DB2 Query Patroller Client is a Separate Component	319
Migrating from Version 6 of DB2 Query Patroller Using dqpmigrate	319
Enabling Query Management	319
Location of Table Space for Control Tables.	319
New Parameters for dqpstart Command	320
New Parameter for iwmm_cmd Command	320
New Registry Variable: DQP_RECOVERY_INTERVAL	321
Starting Query Administrator	321
User Administration	321
Creating a Job Queue.	321
Using the Command Line Interface	321
Query Enabler Notes	322
DB2 Query Patroller Tracker may Return a Blank Column Page	322
Query Patroller and Replication Tools	322
Appendix B. Troubleshooting DB2 Query Patroller Clients	322

Teil 6. Anwendungsentwicklung 323

Administrative API Reference	327
---	------------

db2ArchiveLog (new API)	327
db2ArchiveLog	327
db2ConvMonStream	329
db2DatabasePing (new API)	330
db2DatabasePing - Ping Database	330
db2HistData.	331
db2HistoryOpenScan.	333
db2XaGetInfo (new API)	333
db2XaGetInfo - Get Information for Resource Manager	333
db2XaListIndTrans (new API that supercedes sqlxphqr)	334
db2XaListIndTrans - List Indoubt Transactions	334
db2GetSnapshot – Get Snapshot	337
Forget Log Record.	337
sqlaintp – Get Error Message	338
sqlbctcq – Close Tablespace Container Query.	338
sqlubkp – Backup Database	338
sqlureot – Reorganize Table	338
sqlurestore – Restore Database.	338
Documentation Error Regarding AIX Extended Shared Memory Support (EXTSHM).	339
SQLFUPD	339
locklist	339
SQLEDBDESC	340
SQLFUPD Documentation Error	340

Application Building Guide 341

Chapter 1. Introduction	341
Supported Software	341
Sample Programs	343
Chapter 3. General Information for Building DB2 Applications.	343
Build Files, Makefiles, and Error-checking Utilities	343
Chapter 4. Building Java Applets and Applications	344
Setting the Environment.	344
Chapter 5. Building SQL Procedures.	346
Setting the SQL Procedures Environment	346
Setting the Compiler Environment Variables	346
Customizing the Compilation Command	347
Retaining Intermediate Files	351
Backup and Restore	351
Creating SQL Procedures	351
Calling Stored Procedures	352
Distributing Compiled SQL Procedures.	352
Chapter 7. Building HP-UX Applications.	353
HP-UX C.	353
HP-UX C++.	353
Chapter 9. Building OS/2 Applications.	353
VisualAge C++ for OS/2 Version 4.0	353
Chapter 10. Building PTX Applications	353
ptx/C++.	353
Chapter 12. Building Solaris Applications	354
SPARCCompiler C++.	354
Chapter 13. Building Applications for Windows 32-bit Operating Systems	354
VisualAge C++ Version 4.0	354

Application Development Guide 355

Chapter 2. Coding a DB2 Application	355
Activating the IBM DB2 Universal Database Project and Tool Add-ins for Microsoft Visual C++	355
Chapter 6. Common DB2 Application Techniques	355
Generating Sequential Values	355
Chapter 7. Stored Procedures	358
DECIMAL Type Fails in Linux Java Routines	358
Using Cursors in Recursive Stored Procedures	358
Writing OLE Automation Stored Procedures	358
Chapter 12. Working with Complex Objects: User- Defined Structured Types	359
Inserting Structured Type Attributes Into Columns	359
Chapter 13. Using Large Objects (LOBs)	359
Large object (LOBs) support in federated data- base systems	359
Tuning the system.	361
Part 5. DB2 Programming Considerations	361
IBM DB2 OLE DB Provider.	361
Chapter 20. Programming in C and C++	361
C/C++ Types for Stored Procedures, Functions, and Methods	361
Chapter 21. Programming in Java	363
Java Method Signature in PARAMETER STYLE JAVA Procedures and Functions	363
Connecting to the JDBC Applet Server	363
Appendix B. Sample Programs	363

CLI Guide and Reference 365

Binding Database Utilities Using the Run-Time Cli- ent	365
Using Static SQL in CLI Applications	365
Limitations of JDBC/ODBC/CLI Static Profiling	365
ADT Transforms	366
Chapter 3. Using Advanced Features	366
Writing Multi-Threaded Applications	366
Scrollable Cursors	367
Using compound SQL	368
Using Stored Procedures	369
Chapter 4. Configuring CLI/ODBC and Running Sample Applications	369
Configuration Keywords	369
Chapter 5. DB2 CLI Functions	369
SQLBindFileToParam - Bind LOB File Reference to LOB Parameter	369
SQLNextResult - Associate Next Result Set with Another Statement Handle	370
Appendix D. Extended Scalar Functions	372
Date and Time Functions	372
Appendix K. Using the DB2 CLI/ODBC/JDBC Trace Facility	372

Fehlernachrichten 373

Abrufen der Hilfe zu Nachrichten und SQLSTATE- Werten	373
Änderung der Zuordnung von SQLCODE-Werten in DB2 Connect	373
Neue und geänderte Nachrichten.	373
CLI-Nachrichten (Call Level Interface)	373

DB2-Nachrichten	373
DBI-Nachrichten	374
Nachrichten der Data Warehouse-Zentrale.	374
SQL-Nachrichten	375
Korrigierte SQLSTATE-Werte	377

SQL Reference 379

SQL-Referenz in einer PDF-Datei verfügbar	379
Chapter 3. Language Elements	379
Naming conventions and Implicit Object Name Qualifications	379
DATALINK Assignments	379
Expressions	379
Chapter 4. Functions	385
Enabling the New Functions and Procedures	385
Scalar Functions	386
Table Functions.	410
Procedures	414
Chapter 5. Queries	416
select-statement/syntax diagram	416
select-statement/fetch-first-clause.	417
Chapter 6. SQL Statements	417
Update of the Partitioning Key Now Supported	417
Larger Index Keys for Unicode Databases	418
ALTER SEQUENCE	418
ALTER TABLE	422
Compound SQL (Embedded)	425
Compound Statement (Dynamic)	425
CREATE FUNCTION (Source or Template)	429
CREATE FUNCTION (SQL Scalar, Table or Row)	430
CREATE METHOD	432
CREATE SEQUENCE.	434
CREATE TRIGGER	438
CREATE WRAPPER	440
DECLARE CURSOR	440
DELETE	441
DROP	442
GRANT (Sequence Privileges)	443
INSERT	444
SELECT INTO	445
SET ENCRYPTION PASSWORD	446
SET transition-variable	447
UPDATE	449
Chapter 7. SQL Procedures now called Chapter 7. SQL Control Statements	450
SQL Procedure Statement	450
FOR	452
Compound Statement changes to Compound Statement (Procedure)	454
RETURN	456
SIGNAL	458
Appendix A. SQL Limits	461
Appendix D. Catalog Views	461
SYSCAT.SEQUENCES	461

DB2 Stored Procedure Builder 463

Java 1.2-Unterstützung für DB2 Stored Procedure Builder	463
Fernes Debug von gespeicherten DB2-Prozeduren	464

Erstellen von SQL-Prozeduren auf Windows-, OS/2- oder UNIX-Plattformen	464
Verwenden von DB2 Stored Procedure Builder auf der Solaris-Plattform	464
Bekannte Probleme und Einschränkungen	464
Using DB2 Stored Procedure Builder with Traditional Chinese Locale	465
UNIX-Installationen (AIX, Sun Solaris, Linux) und Stored Procedure Builder	465
Erstellen von gespeicherten SQL-Prozeduren unter OS/390	465
Debug von gespeicherten SQL-Prozeduren	465
Exportieren gespeicherter Java-Prozeduren	466
Einfügen gespeicherter Prozeduren unter OS/390	466
Einstellen von Erstellungsoptionen für gespeicherte SQL-Prozeduren auf einem Workstation-Server	466
Automatisches Aktualisieren des WLM-Adressraums für gespeicherte Prozeduren unter OS/390	467
Entwicklung von gespeicherten Java-Prozeduren unter OS/390	467
Aufbau einer benutzerdefinierten Funktion in einer DB2-Tabelle für MQ Series und OLE DB	467
Unicode-Aktualisierungen	469
Einführung	469
DB2-Unicode-Datenbanken und -Anwendungen	469
Aktualisierung der Dokumentation	471
SQL Reference	471
Chapter 3 Language Elements	471
Chapter 4 Functions	475
CLI Guide and Reference	476
Chapter 3. Using Advanced Features	476
Appendix C. DB2 CLI and ODBC	481
Versetzen von Daten Dienstprogramme und Referenz	482
Anhang C, Dateiformate der Dienstprogramme	
EXPORT/IMPORT/LOAD	482

Teil 7. Verbindung zu Host-Systemen 483

Konnektivität: Ergänzung	485
Konfigurieren des Anwendungs-Servers in einer VM-Umgebung	485
Einstellungen von PATCH1 und PATCH2 bei der CLI/ODBC/JDBC-Konfiguration	485

Teil 8. Allgemeine Informationen 489

Allgemeine Informationen	491
DB2 Universal Database Informationsmanagement	
Kurzüberblick	491
DB2 Everywhere jetzt DB2 Everyplace	492
Maus erforderlich	492
Fehler "Bindedateien nicht gefunden" beim Versuch, von DB2 Run-Time Client aus zu binden	492
Discovery-Modus SEARCH.	492
Speicherfenster für HP-UX 11	492

Benutzeraktion für "dlfm client_conf" schlägt fehl	493
Wenn der Kopierdämon bei "dlfm stop" nicht stoppen sollte.	494
DB2 DFS Client Enabler deinstallieren	494
Client-Authentifizierung unter Windows NT	494
AutoLoader kann während einer Aufspaltung blockieren	494
DATALINK-Wiederherstellung	495
Definieren von Benutzer-ID und Kennwort in IBM Communications Server für Windows NT (CS/NT).	495
Knotendefinition	495
Einschränkungen bei Systemen zusammengeschlossener Datenbanken	496
Einschränkung bei DataJoiner	496
Hebrew Information Catalog Manager for Windows NT	496
Unterstützung für Microsoft SNA Server und SNA-Aktualisierung an mehreren Standorten (zweiphasige Festschreibung)	497
DB2 SNA SPM kann nach dem Booten von Windows nicht gestartet werden	497
Festlegen der Ländereinstellung für den DB2-Verwaltungsserver.	498
Nicht funktionierende Tastenkombinationen	498
Anforderungen an Servicebenutzereinträge für DB2 unter Windows NT und Windows 2000	498
Verlorenes Zugriffsrecht EXECUTE für in Version 6 erstellte Benutzer von Query Patroller	499
Query Patroller - Einschränkungen	499
Notwendigkeit zum Festschreiben aller benutzerdefinierten Programme, die in der Data Warehouse-Zentrale verwendet werden.	499
Neue Option für Export über die Befehlszeile der Data Warehouse-Zentrale	500
Sicherungsservice-APIs (XBSA)	500
OS/390-Agent	501
Installationsübersicht	501
Installationsdetails.	502
Konfigurieren zusätzlicher Agentenfunktionen	505
Planen von Warehouse-Schritten mit dem Auslöserprogramm (XTClient).	507
Umsetzungsprogramme	508
Zugreifen auf Datenbanken außerhalb der DB2-Systemfamilie	511
Ausführen von Dienstprogrammen von DB2 für OS/390	513
Replikation	515
Agentenprotokollierung	516
Client-seitiges Caching in Windows NT	517
Testprodukte auf den CD-ROMs von Enterprise Edition für UNIX	517
Testprodukte auf den CD-ROMs von DB2 Connect Enterprise Edition für UNIX	518
Löschen von Data Links Manager	519
Deinstallation von DLFM-Komponenten mit SMIT entfernt unter Umständen zusätzliche Dateigruppen	519
Fehler SQL1035N bei der Verwendung des Befehlszeilenprozessors unter Windows 2000	519
Verbesserung bei SQL Assist	519

DB2 für Linux - Integration der Gnome- und KDE- Desktops	520
Ausführen von DB2 unter einem Windows 2000- Terminalserver (im Verwaltungsmodus)	521
Onlinehilfefunktion für Befehle zum Sichern und Wiederherstellen	521
"Warehouse Manager" muss "DB2 Warehouse Manager" heißen	521

Teil 9. Zusätzliche Informationen 523

Zusätzliche Informationen	525
--	------------

Online-Unterstützung für DB2 Universal Database und DB2 Connect	525
DB2-Magazin	525

Teil 10. Anhänge und Schlussteil 527

Anhang. Bemerkungen	529
Marken	531

Index	533
------------------------	------------

Willkommen bei DB2 Universal Database Version 7!

Anmerkung: Geben Sie eine Monospace-Schrift an, damit Sie diese Release-Informationen bequemer lesen können.

Die Unterstützungs-Site von DB2 Universal Database und DB2 Connect wird regelmäßig aktualisiert. Überprüfen Sie <http://www.ibm.com/software/data/db2/udb/winos2unix/support> auf neueste Informationen.

Diese Datei enthält Informationen zu den folgenden Produkten, die bei der Drucklegung der DB2-Handbücher noch nicht verfügbar waren:

- IBM DB2 Universal Database Personal Edition, Version 7.2
- IBM DB2 Universal Database Workgroup Edition, Version 7.2
- IBM DB2 Universal Database Enterprise Edition, Version 7.2
- IBM DB2 Data Links Manager, Version 7.2
- IBM DB2 Universal Database Enterprise - Extended Edition, Version 7.2
- IBM DB2 Query Patroller, Version 7.2
- IBM DB2 Personal Developer's Edition, Version 7.2
- IBM DB2 Universal Developer's Edition, Version 7.2
- IBM DB2 Data Warehouse Manager, Version 7.2
- IBM DB2 Relational Connect, Version 7.2

Für die folgenden Produkte ist eine separate Datei namens READCON.TXT mit Release-Informationen verfügbar:

- IBM DB2 Connect Personal Edition, Version 7.2
- IBM DB2 Connect Enterprise Edition, Version 7.2

Das Handbuch *Neue Funktionen* enthält eine Übersicht über einige der wichtigsten Erweiterungen in DB2 Version 7.2. Wenn Sie das Handbuch *Neue Funktionen* für Version 7.2 nicht haben, können Sie es sich unter <http://www.ibm.com/software/data/db2/udb/winos2unix/support> anschauen und herunterladen.

Anmerkung: Eine Änderungsmarkierung (1) am linken Rand einer Seite zeigt an, dass die daneben stehende Zeile seit der ersten Veröffentlichung der Release-Informationen hinzugefügt oder geändert worden ist.

Teil 1. Besondere Hinweise

Besondere Hinweise

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

Bessere Zugriffsmöglichkeiten von DB2 UDB Version 7

Die DB2 UDB-Produktfamilie umfasst eine Reihe von Funktionen, durch die die Produkte für Benutzer mit Behinderungen leichter bedienbar werden. Dazu gehören die folgenden Funktionen und Merkmale:

- Funktionen, die die Tastatureingabe und die Navigation erleichtern
- Funktionen, die die Anzeigemerkmale verbessern
- Optionen für Warntöne und Warnanzeigen
- Kompatibilität mit Unterstützungstechnologien
- Kompatibilität mit Zugriffsmöglichkeiten des Betriebssystems
- Dokumentationsformate mit besseren Zugriffsmöglichkeiten

Tastatureingabe und Navigation

Tastatureingabe

Die DB2-Steuerzentrale kann z. T. über die Tastatur bedient werden. Menüpunkte und Steuerungen bieten Zugriffstasten, mit denen Benutzer direkt über die Tastatur ein Steuerelement aktivieren oder einen Menüpunkt auswählen können. Welche Tasten Zugriffstasten sind, wird durch die Unterstreichung des entsprechenden Buchstabens in den jeweiligen Steuerelementen oder Menüs erkennbar.

Tastatureingabebereich

Auf UNIX-basierten Systemen ist der Tastatureingabebereich hervorgehoben, so dass zu sehen ist, welcher Bereich des Fensters aktiv ist und wo Tastenanschläge des Benutzers eine Auswirkung haben werden.

Bessere Zugriffsmöglichkeiten für Anzeige

Die DB2-Steuerzentrale bietet eine Reihe von Funktionen, die die Benutzerschnittstelle erweitern und Zugriffsmöglichkeiten für sehbehinderte Benutzer verbessern. Diese Verbesserungen umfassen Unterstützung für starken Kontrast und anpassbare Schriftartmerkmale.

Modus für starken Kontrast

Die Schnittstelle der Steuerzentrale unterstützt den Modus für starken Kontrast, der vom Betriebssystem zur Verfügung gestellt wird. Diese Funktion hilft Benutzern, die einen stärkeren Kontrast zwischen Vordergrund- und Hintergrundfarben benötigen.

Schriftarteinstellungen

Die Schnittstelle der Steuerzentrale ermöglicht die Auswahl der Schriftfarbe, Schriftgröße und Schriftart für den Text in Menüs und Dialogfenstern.

Farberkennung nicht erforderlich

Die Benutzer müssen zur Verwendung der Funktionen des Produkts die Farben nicht unterscheiden können.

Alternative Alert-Hinweise

Der Benutzer kann auswählen, ob hörbar oder sichtbar auf Alerts hingewiesen werden soll.

Kompatibilität mit Unterstützungstechnologien

Die Schnittstelle der DB2-Steuerzentrale ist kompatibel mit Bildschirmsprachausgabenanwendungen wie z. B. Via Voice. Im Anwendungsmodus hat die Schnittstelle der Steuerzentrale die erforderlichen Merkmale, mit denen diese Unterstützungsanwendungen blinden Benutzern die auf dem Bildschirm angezeigten Informationen verfügbar machen können.

Dokumentationsformate mit besseren Zugriffsmöglichkeiten

Die Dokumentation für die DB2-Produktfamilie ist im HTML-Format verfügbar. Dadurch können Benutzer die Dokumentation mit den Vorgaben in ihrem Browser anzeigen. Außerdem wird dadurch die Verwendung von Bildschirmsprachausgabeprogrammen und anderen Unterstützungstechnologien möglich.

Zusätzliche Solaris-Programmkorrekturstufe erforderlich

DB2 Universal Database Version 7 für Solaris Version 2.6 setzt die Programmkorrektur 106285-02 oder höher zusätzlich zu den im Handbuch *DB2 für UNIX Einstieg* aufgelisteten Programmkorrekturen voraus.

Von DB2 Version 7 für Solaris unterstützte CPUs

CPU-Versionen vor UltraSparc werden nicht unterstützt.

Probleme beim Hinzufügen von Knoten zu einer partitionierten Datenbank

Wenn einer partitionierten Datenbank, die mindestens einen temporären Systemtabellenbereich mit einer anderen Seitengröße als der Standardseitengröße (4 KB) hat, Knoten hinzugefügt werden, kann folgende Fehlermeldung und ein SQLCODE-Wert angezeigt werden: „SQL6073N Das Hinzufügen von Knoten ist fehlgeschlagen“. Dies passiert, weil beim Erstellen des Knotens nur der Pufferpool IBMDEFAULTBP mit einer Seitengröße von 4 KB vorhanden ist. Sie können z. B. den Befehl **db2start** verwenden, um der aktuellen partitionierten Datenbank einen Knoten hinzuzufügen:

```
DB2START NODENUM 2 ADDNODE HOSTNAME neuerhost PORT 2
```

Wenn die partitionierte Datenbank temporäre Systemtabellenbereiche mit der Standardseitengröße hat, wird die folgende Nachricht zurückgegeben:

```
SQL6075W Die Operation START DATABASE MANAGER wurde erfolgreich zum Knoten hinzugefügt. Der Knoten ist erst aktiv, nachdem alle Knoten gestoppt und erneut gestartet wurden.
```

Wenn jedoch die partitionierte Datenbank temporäre Systemtabellenbereiche mit einer anderen Größe als der Standardseitengröße hat, wird die folgende Nachricht zurückgegeben:

```
SQL6073N Das Hinzufügen von Knoten ist fehlgeschlagen. SQLCODE-Wert = "<-902>"
```

In einem ähnlichen Beispiel können Sie den Befehl **ADD NODE** verwenden, nachdem die Datei `db2nodes.cfg` manuell mit der neuen Knotenbeschreibung aktuali-

siert wurde. Nachdem die Datei editiert und der Befehl ADD NODE mit einer partitionierten Datenbank mit temporären Systemtabellenbereichen mit der Standardseitengröße ausgeführt wurde, wird folgende Nachricht zurückgegeben:

```
DB20000I Der Befehl ADD NODE wurde erfolgreich ausgeführt.
```

Wenn jedoch die partitionierte Datenbank temporäre Systemtabellenbereiche mit einer anderen Größe als der Standardseitengröße hat, wird die folgende Nachricht zurückgegeben:

```
SQL6073N Das Hinzufügen von Knoten ist fehlgeschlagen. SQLCODE-Wert = "<-902>"
```

Eine Möglichkeit, die oben dargestellten Probleme zu vermeiden, ist die Ausführung des folgenden Befehls:

```
DB2SET DB2_HIDDENBP=16
```

Dieser Befehl muss vor dem Befehl **db2start** oder ADD NODE abgesetzt werden. Durch diese Registrierungsvariable kann DB2 verdeckte Pufferpools von je 16 Seiten zuordnen, die jeweils eine von der Standardgröße abweichende Seitengröße verwenden. Dadurch kann die Operation ADD NODE erfolgreich durchgeführt werden.

Eine weitere Möglichkeit zur Umgehung dieses Problems ist die Angabe der Klausel WITHOUT TABLESPACES beim Befehl ADD NODE oder **db2start**. Danach müssen Sie die Pufferpools mit der Anweisung CREATE BUFFERPOOL erstellen und die temporären Systemtabellenbereiche mit der Anweisung ALTER TABLESPACE dem Pufferpool zuordnen.

Wenn einer vorhandenen Knotengruppe, die mindestens einen Tabellenbereich mit einer anderen Seitengröße als der Standardseitengröße (4 KB) hat, Knoten hinzugefügt werden, kann folgende Fehlermeldung angezeigt werden: „SQL0647N Pufferpool "" ist zur Zeit nicht aktiv.“ Dieser Fehler tritt auf, weil die auf dem neuen Knoten erstellten Pufferpools mit einer vom Standard abweichenden Seitengröße für die Tabellenbereiche nicht aktiviert wurden.

Sie können z. B. die Anweisung ALTER NODEGROUP verwenden, um einen Knoten zu einer Knotengruppe hinzuzufügen:

```
DB2START  
CONNECT TO mpp1  
ALTER NODEGROUP ng1 ADD NODE (2)
```

Wenn die Knotengruppe Tabellenbereiche mit der Standardseitengröße hat, wird die folgende Nachricht zurückgegeben:

```
SQL1759W Es ist erforderlich, die Knotengruppe umzuverteilen, um die  
Datenpartitionierung für Objekte in der Knotengruppe  
"<name-der-knotengruppe>" zu ändern, damit hinzugefügte  
Knoten aufgenommen oder gelöschte Knoten entfernt werden können.
```

Wenn jedoch die Knotengruppe Tabellenbereiche mit einer anderen Größe als der Standardseitengröße hat, wird die folgende Nachricht zurückgegeben:

```
SQL0647N Pufferpool "" ist zur Zeit nicht aktiv.
```

Eine Möglichkeit zum Umgehen dieses Problems besteht darin, Pufferpools für jede Seitengröße zu erstellen und sie dann wieder mit der Datenbank zu verbinden, bevor Sie die Anweisung ALTER NODEGROUP absetzen:

```
DB2START
CONNECT TO mpp1
CREATE BUFFERPOOL bp1 SIZE 1000 PAGESIZE 8192
CONNECT RESET
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2)
```

Eine zweite Umgehungsmöglichkeit für dieses Problem besteht in der Ausführung des folgenden Befehls:

```
DB2SET DB2_HIDDENBP=16
```

Dies muss erfolgen, bevor Sie den Befehl **db2start** und die Anweisungen **CONNECT** und **ALTER NODEGROUP** absetzen.

Ein weiteres Problem kann auftreten, wenn die Anweisung **ALTER TABLESPACE** verwendet wird, um einem Tabellenbereich einen Knoten hinzuzufügen. Beispiel:

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

Diese Reihe von Befehlen und Anweisungen generiert die Fehlermeldung **SQL0647N** (nicht die erwartete Nachricht **SQL1759W**).

Damit diese Änderung korrekt abgeschlossen wird, sollten Sie die Verbindung zur Datenbank wiederherstellen, nachdem die Anweisung **ALTER NODEGROUP... WITHOUT TABLESPACES** ausgeführt wurde.

```
DB2START
CONNECT TO mpp1
ALTER NODEGROUP ng1 ADD NODE (2) WITHOUT TABLESPACES
CONNECT RESET
CONNECT TO mpp1
ALTER TABLESPACE ts1 ADD ('ts1') ON NODE (2)
```

Eine weitere Umgehungsmöglichkeit für dieses Problem besteht in der Ausführung des folgenden Befehls:

```
DB2SET DB2_HIDDENBP=16
```

Dies muss erfolgen, bevor Sie den Befehl **db2start** und die Anweisungen **CONNECT**, **ALTER NODEGROUP** und **ALTER TABLESPACE** absetzen.

Fehler bei der Migration

Bei der Migration erscheinen Fehlereinträge in der Datei `db2diag.log` (Datenbank nicht migriert), selbst wenn die Migration erfolgreich war. Diese können daher ignoriert werden.

Chinese Locale Fix on Red Flag Linux

If you are using Simplified Chinese Red Flag Linux Server Version 1.1, contact Red Flag to receive the Simplified Chinese locale fix. Without the Simplified Chinese locale fix for Version 1.1, DB2 does not recognize that the code page of Simplified Chinese is 1386.

DB2-Installation kann blockieren, wenn ein Wechsellaufwerk nicht angeschlossen ist

Die DB2-Installation kann nach der Auswahl der Installationsart blockieren, wenn ein Computer mit einem Wechsellaufwerk verwendet wird, das nicht angeschlossen ist. Geben Sie die Option `-a` an, um dieses Problem zu beheben:

```
setup.exe -a
```

Additional Locale Setting for DB2 for Linux in a Japanese and Simplified Chinese Linux Environment

An additional locale setting is required when you want to use the Java GUI tools, such as the Control Center, on a Japanese or Simplified Chinese Linux system. Japanese or Chinese characters cannot be displayed correctly without this setting. Please include the following setting in your user profile, or run it from the command line before every invocation of the Control Center.

```
For a Japanese system:  
export LC_ALL=ja_JP
```

```
For a Simplified Chinese system:  
export LC_ALL=zh_CN
```

Problem der Steuerzentrale bei Microsoft Internet Explorer

Die Sicherheitseinstellungen des Internet Explorer (IE) verursachen ein Problem. Da die Steuerzentrale JAR-Dateien ohne Vorzeichen verwendet, wird der Zugriff auf Systeminformationen durch den Sicherheitsmanager inaktiviert.

Konfigurieren Sie die IE-Sicherheitsoptionen wie folgt, um dieses Problem zu beheben:

1. Wählen Sie **Internetoptionen** im Menü **Anzeige** (IE4) bzw. im Menü **Extras** (IE5) aus.
2. Wählen Sie **Vertrauenswürdige Sites** auf der Seite **Sicherheit** aus.
3. Klicken Sie **Sites hinzufügen...** an.
4. Fügen Sie den Web-Server der Steuerzentrale zur Liste der vertrauenswürdigen Sites hinzu. Wenn der Web-Server der Steuerzentrale sich in derselben Domäne befindet, kann es sinnvoll sein, nur den Web-Server-Namen (ohne den Domänennamen) hinzuzufügen. Beispiel:

```
http://ccWebServer.ccWebServerDomain  
http://ccWebServer
```
5. Klicken Sie **OK** an.
6. Klicken Sie **Stufe anpassen...** an.
7. Blättern Sie zu **Java** → **Java-Einstellungen**, und wählen Sie **Benutzerdefiniert** aus.
8. Klicken Sie **Java-Einstellungen...** an.
9. Wählen Sie die Seite **Zugriffsrechte bearbeiten** aus.
10. Blättern Sie zu **Nicht signierter Inhalt** → **Nicht signierten Inhalt ausführen** → **Zusätzliche nicht signierte Zugriffsrechte** → **Systeminformation**, und wählen Sie **Aktivieren** aus.
11. Klicken Sie in jedem geöffneten Fenster **OK** an.

Inkompatibilität zwischen Information Catalog Manager und Sybase in der Windows-Umgebung

Wird Information Catalog Manager (ICM) Version 7 auf derselben Windows NT- oder Windows 2000-Maschine installiert wie Sybase Open Client, tritt ein Fehler auf, und die Sybase-Dienstprogramme funktionieren nicht mehr. Eine der folgenden ähnliche Fehlernachricht wird angezeigt:

```
Fail to initialize LIBTCL.DLL. Please make sure the SYBASE environment variable is set correctly.
```

Vermeiden Sie dieses Szenario, indem Sie den Umgebungsparameter LC_ALL aus den Windows-Umgebungsparametern entfernen. LC_ALL ist ein Parameter für Ländereinstellungskategorien. Ländereinstellungskategorien sind Konstanten, mit denen die Lokalisierungsroutinen angeben, welcher Abschnitt der Ländereinstellungsinformationen für ein Programm zu verwenden ist. Die *Ländereinstellung* bezieht sich auf die Lokalität (oder das Land), für die bestimmte Aspekte Ihres Programms angepasst werden können. Zu den Ländereinstellungen gehört z. B. das Datumsformat oder das Anzeigeformat für Währungen. LC_ALL betrifft das gesamte ländereinstellungsspezifische Verhalten (alle Kategorien).

Wenn Sie den Umgebungsparameter LC_ALL entfernen, so dass ICM unter Windows NT mit Sybase koexistieren kann, funktionieren folgende Funktionen nicht mehr:

- Information Catalog-Benutzer
- Information Catalog-Administrator
- Information Catalog-Manager

Verlust der Steuerzentralenfunktion

Durch die Anwendung von FixPak 2 auf einem DB2-Server sollten keine Probleme für Steuerzentralen-Clients mit einer älteren Version eingeführt werden. Bei DB2 Version 7.2 verlieren solche Steuerzentralen-Clients jedoch fast ihre gesamte Funktionalität. Als „Clients mit einer älteren Version“ werden in diesem Fall alle Clients der Version 6 vor FixPak 6 und alle Clients der Version 7 vor FixPak 2 bezeichnet. Clients der Version 5 sind nicht betroffen.

Der Korrekturvorschlag besteht in einem Upgrade der betroffenen Clients. Für Clients der Version 6 ist ein Upgrade auf FixPak 6 oder später und für Clients der Version 7 ein Upgrade auf FixPak 2 oder später erforderlich.

Netscape-CD wird nicht mit DB2 UDB ausgeliefert

Die Netscape-CD wird nicht mehr mit DB2 UDB ausgeliefert. Netscape-Produkte sind unter <http://www.netscape.com> verfügbar.

Fehler in den XML-Readme-Dateien

In der Datei README.TXT zu DB2 XML Extender Version 7.1 steht im Abschnitt zu den zu berücksichtigenden Überlegungen Folgendes:

3. Die Standardversion von DB2 UDB ist DB2 UDB Version 7.1. Wenn Sie DB2 UDB Version 6.1 unter AIX und Solaris verwenden wollen, sollten Sie sicherstellen, dass Sie ein Exemplar von DB2 UDB Version 6.1 und die Bibliotheken von DB2 UDB Version 6.1 ausführen.

Dies ist nicht korrekt. **DB2 XML Extender wird nur mit DB2 Version 7.1 und 7.2 unterstützt.**

Die Dateien "readme.aix", "readme.nt" und "readme.sun" listen die folgenden Softwarevoraussetzungen auf:

- DB2 UDB 6.1 mit FP1_U465423 oder höher (AIX)
- DB2 Universal Database Version 6.1 oder höher mit installiertem FixPak 3 (NT)
- DB2 UDB Version 6.1 mit FixPak FP1_U465424 oder höher (Sun)

Dies ist nicht korrekt. **DB2 XML Extender erfordert DB2 Version 7.1 oder 7.2.**

Möglicher Datenverlust unter Linux für S/390

Wenn Sie DB2 unter Linux für S/390 mit einem Kernel der Serie 2.2 verwenden, sollte der verfügbare Arbeitsspeicher auf der Linux-Maschine auf weniger als 1 GB begrenzt werden. Die Begrenzung des Arbeitsspeichers auf 1 GB verhindert mögliche Datenverluste in DB2 aufgrund eines Linux-Kernel-Programmfehlers.

Dies gilt nur unter Linux für S/390 und nicht unter Linux für Intel-Umgebungen.

Eine Kernel-Programmkorrektur wird unter

http://www10.software.ibm.com/developerworks/opensource/linux390/alpha_src.html

zur Verfügung gestellt werden, die den Einsatz von mehr als 1 GB Arbeitsspeicher ermöglicht.

DB2 UDB unter Windows 2000

Wenn in diesen Release-Informationen auf Windows NT verwiesen wird, schließt dies Windows 2000 ein, wenn nichts Anderes angegeben ist.

Onlinedokumentation (HTML, PDF und Suche)

Unterstützte Web-Browser unter Windows 2000

Wir empfehlen unter Windows 2000 die Verwendung von Microsoft Internet Explorer.

Wenn Sie Netscape verwenden, beachten Sie bitte Folgendes:

- Das Suchen in der DB2-Onlineinformation kann bei Verwendung von Netscape unter Windows 2000 lang dauern. Netscape nutzt alle verfügbaren CPU-Ressourcen und scheint nicht fertig zu werden. Zwar werden die Suchergebnisse schließlich angezeigt. Es empfiehlt sich jedoch, das Fenster nach dem Starten der Suche zu wechseln. Die Suchergebnisse werden dann in einer vernünftigen Zeitspanne zurückgegeben.
- Es kann auch Folgendes auftreten: Wenn Sie Hilfe aufrufen, wird diese korrekt in einem Netscape-Browser-Fenster angezeigt. Wenn Sie das Browser-Fenster jedoch geöffnet lassen und später von einer anderen Stelle in der Steuerzentrale den Hilfetext aufrufen, wird im Browser keine neue Seite angezeigt. Wenn Sie das Browser-Fenster schließen und dann erneut die Hilfe aufrufen, wird die korrekte Hilfeseite angezeigt. Sie können dieses Problem anhand der Schritte in „Fehlernachrichten beim Versuch, Netscape zu starten“ auf Seite 12 beheben. Sie können dieses Problem auch umgehen, indem Sie das Browser-Fenster schließen, bevor Sie Hilfe für die Steuerzentrale aufrufen.
- Wenn Sie Hilfe zur Steuerzentrale oder ein Thema aus **Information - Unterstützung** aufrufen, kann eine Fehlernachricht angezeigt werden. Führen Sie die in „Fehlernachrichten beim Versuch, Netscape zu starten“ auf Seite 12 beschriebenen Schritte aus, um dieses Problem zu beheben.

Suchen nach DB2-Onlineinformationen unter Solaris

Wenn beim Durchsuchen der DB2-Onlineinformationen unter Solaris Probleme auftreten, prüfen Sie bitte die Kernel-Parameter Ihres Systems in `/etc/system`. Im Folgenden sind die für das DB2-Suchsystem, NetQuestion, erforderlichen Mindesteinstellungen der Kernel-Parameter aufgelistet:

```
semsys:seminfo_semmni 256
semsys:seminfo_semmap 258
semsys:seminfo_semmns 512
semsys:seminfo_semmnu 512
semsys:seminfo_semmnl 50
shmsys:shminfo_shmmax 6291456
shmsys:shminfo_shmseg 16
shmsys:shminfo_shmmni 300
```

Fügen Sie am Ende der Datei `/etc/system` folgende Zeile hinzu, um einen Kernel-Parameter zu setzen:

```
set <semaphorname> = wert
```

Sie müssen Ihr System neu starten, damit neue oder geänderte Werte wirksam werden.

Einrichten von NetQuestion für OS/2 zur Verwendung von TCP/IP

Die Anweisungen zum Einrichten von NetQuestion zur Verwendung von TCP/IP auf OS/2-Systemen sind unvollständig. Die Speicherposition der in diesen Anweisungen erwähnten *.cfg-Dateien ist das Datenunterverzeichnis des Installationsverzeichnisses von NetQuestion. Sie können das Installationsverzeichnis von NetQuestion durch die Eingabe eines der folgenden Befehle ermitteln:

```
echo %IMNINSTSRV% //für SBCS-Installationen
echo %IMQINSTSRV% //für DBCS-Installationen
```

Fehlernachrichten beim Versuch, Netscape zu starten

Es kann vorkommen, dass beim Starten von Netscape die folgenden Fehlernachrichten angezeigt werden:

```
Cannot find file <file path> (or one of its components).
Check to ensure the path and filename are correct and that all
required libraries are available.
```

```
Unable to open "D:\Program Files\SQLLIB\CC\..\doc\html\db2help\XXXXX.htm"
```

Führen Sie dann die folgenden Schritte aus, um das Problem unter Windows NT, 95 oder 98 zu beheben (Anweisungen für Windows 2000 finden Sie im Anschluss):

1. Wählen Sie im Menü **Start** die Option **Programme** —> **Windows-Explorer** aus. Der Windows-Explorer wird geöffnet.
2. Wählen Sie im Windows-Explorer **Ansicht** —> **Optionen** aus. Das Notizbuch **Optionen** wird geöffnet.
3. Klicken Sie die Indexzunge **Dateitypen** an. Die Seite **Dateitypen** wird geöffnet.
4. Heben Sie **Netscape Hypertext Document** im Feld **Registrierte Dateitypen** hervor, und klicken Sie **Bearbeiten** an. Das Fenster **Dateityp bearbeiten** wird geöffnet.
5. Heben Sie **Öffnen** im Feld **Vorgänge** hervor.
6. Klicken Sie den Druckknopf **Bearbeiten** an. Das Fenster **Vorgang bearbeiten für Typ** wird geöffnet.
7. Machen Sie die Auswahl des Markierungsfelds **DDE verwenden** rückgängig.
8. Im Feld **Anwendung für diesen Vorgang** muss am Ende der Zeichenfolge "%1" (eingeschlossen in Anführungszeichen und mit einer Leerstelle vor dem ersten Anführungszeichen) stehen.

Wenn die Nachrichten unter Windows 2000 angezeigt werden, sollten Sie die folgenden Schritte ausführen:

1. Wählen Sie **Windows-Explorer** im Menü **Start** aus. Der Windows-Explorer wird geöffnet.
2. Wählen Sie im Windows-Explorer **Extras** —> **Ordneroptionen** aus. Das Notizbuch **Ordneroptionen** wird geöffnet.
3. Klicken Sie die Indexzunge **Dateitypen** an.
4. Markieren Sie auf der Seite **Dateitypen** im Feld **Registrierte Dateitypen** die Option „HTM Netscape Hypertext Document“, und klicken Sie „Erweitert“ an. Das Fenster **Dateityp bearbeiten** wird geöffnet.
5. Heben Sie **open** im Feld **Vorgänge** hervor.

6. Klicken Sie den Druckknopf **Bearbeiten** an. Das Fenster **Vorgang bearbeiten für Typ** wird geöffnet.
7. Machen Sie die Auswahl des Markierungsfelds **DDE verwenden** rückgängig.
8. Im Feld **Anwendung für diesen Vorgang** muss am Ende der Zeichenfolge "%1" (eingeschlossen in Anführungszeichen und mit einer Leerstelle vor dem ersten Anführungszeichen) stehen.
9. Klicken Sie **OK** an.
10. Wiederholen Sie Schritt 4 bis 8 für die Dateitypen **HTML Netscape Hypertext Document** und **SHTML Netscape Hypertext Document**.

Konfigurationsanforderung für Adobe Acrobat Reader auf UNIX-basierten Systemen

Acrobat Reader wird auf UNIX-basierten Plattformen nur auf Englisch angeboten. Es kann daher zu Fehlernachrichten kommen, wenn versucht wird, PDF-Dateien mit einer anderen Ländereinstellung als Englisch zu öffnen. Diese Fehler weisen auf Probleme beim Zugriff auf Schriften oder bei der Extraktion der PDF-Datei hin, entstehen aber tatsächlich dadurch, dass der englische Acrobat Reader in UNIX mit anderen Ländereinstellungen als Englisch nicht korrekt funktioniert.

Wenn Sie solche PDF-Dateien anzeigen wollen, wechseln Sie mit einem der unten aufgeführten Schritte zur englischen Ländereinstellung, bevor Sie den englischen Acrobat Reader starten:

- Bearbeiten Sie das Acrobat Reader-Start-Script, indem Sie folgende Zeile nach der Anweisung `#!/bin/sh` in der Start-Script-Datei hinzufügen:

```
LANG=C;export LANG
```

Dadurch wird das korrekte Verhalten sichergestellt, wenn Acrobat Reader von anderen Anwendungen, z. B. von Netscape Navigator oder von einem Anwendungshilfemenü aus gestartet wird.

- Geben Sie `LANG=C` an der Eingabeaufforderung ein, um die Anwendungsumgebung von Acrobat Reader auf Englisch einzustellen.

Weitere Informationen erhalten Sie bei Adobe Systems (<http://www.Adobe.com>).

SQL-Referenz in einer PDF-Datei verfügbar

Im Anhang *Verwenden der DB2-Bibliothek* (Using the DB2 Library) in jedem Handbuch wird angegeben, dass die SQL-Referenz im PDF-Format in zwei Bänden verfügbar ist. Dies ist nicht korrekt.

Obwohl das gedruckte Buch aus zwei Bänden besteht und die beiden Formnummern korrekt sind, gibt es nur eine PDF-Datei, die beide Bände enthält. Der Name dieser PDF-Datei lautet: `db2s0x70`.

Teil 2. Installation und Konfiguration

Allgemeine Installationsinformationen	17
Herunterladen von Installationspaketen für alle unterstützten DB2-Clients	17
Installieren von DB2 unter Windows 2000	17
Migrationsproblem bei mit Sonderregistern definierten Sichten	17
IPX/SPX-Protokollunterstützung unter Windows 2000	17
Stoppen von DB2-Prozessen vor dem Upgrade älterer Versionen von DB2.	18
Ausführen von db2iupdt nach der Installation von DB2, wenn ein anderes DB2-Produkt bereits installiert ist	18
Definieren der Linux-Umgebung zur Ausführung der DB2-Steuerzentrale	18
DB2 Universal Database Enterprise Edition und DB2 Connect Enterprise Edition für Linux unter S/390	19
DB2 Universal Database Enterprise - Extended Edition für UNIX Einstieg.	20
Kernel-Parameter "shmseg" für HP-UX	20
Migrieren von IBM Visual Warehouse-Steuerungsdatenbanken	20
Zugreifen auf Warehouse-Steuerungsdatenbanken	20
Data Links Manager Einstieg	21
Fehlschlagen von "dlfm start" mit der Nachricht, dass beim Abruf von afsfid für Präfix ein Fehler aufgetreten ist	21
Einrichten einer Tivoli Storage Manager-Verwaltungsklasse für Archivierungsdateien	21
Erforderlicher Plattenspeicherplatz für DFS Client Enabler.	21
Überwachen der Back-End-Prozesse von Data Links File Manager unter AIX	21
Installation und Konfiguration von DB2 Data Links Manager für AIX: Zusätzliche Installationsvoraussetzungen in DCE-DFS-Umgebungen.	22
Fehlgeschlagener Befehl "dlfm add_prefix"	22
Installation und Konfiguration von DB2 Data Links Manager für AIX: Installieren von DB2 Data Links Manager unter AIX mit dem Dienstprogramm db2setup	23
Installation und Konfiguration von DB2 Data Links Manager für AIX: Tasks nach Abschluss der Installation in DCE-DFS.	23
Installation und Konfiguration von DB2 Data Links Manager für AIX: Manuelles Installieren von DB2 Data Links Manager mit SMIT	24
Installation und Konfiguration von DB2 Data Links DFS Client Enabler	24
Installation und Konfiguration von DB2 Data Links Manager für Solaris	24
Auswählen einer Sicherungsmethode für DB2 Data Links Manager unter AIX.	24
Auswählen einer Sicherungsmethode für DB2 Data Links Manager in der Solaris-Betriebsumgebung	26
Auswählen einer Sicherungsmethode für DB2 Data Links Manager unter Windows NT	27
Sichern von JFS (Journaled File System) unter AIX	29
Administratorgruppenzugriffsrechte in Data Links unter Windows NT.	30
Minimieren der Protokollierung für DLFF-Installation (Data Links File System Filter)	30
Protokollieren von Nachrichten nach der Installation	31
Deinstallation von DLFM-Komponenten mit SMIT entfernt unter Umständen zusätzliche Dateigruppen.	32
Vorbereitung/Feststellen des Host-Namens.	32
Arbeiten mit Data Links File Manager: Bereinigung nach dem Löschen von DB2 Data Links Manager aus einer DB2-Datenbank.	33
DLFM1001E (Neue Fehlernachricht)	34
DLFM-Konfigurationsdateioption	34
Fehler beim Ausführen von des Data Links/DFS-Scripts "dmapp_prestart" unter AIX	34
Integration von Tivoli Space Manager in Data Links	35
Einschränkungen und Begrenzungen	36
Kapitel 4. Installieren und Konfigurieren von DB2 Data Links Manager für AIX	37
Allgemeine Installationsvoraussetzungen	37
Migration von DB2 File Manager Version 5.2 auf DB2 Data Links Manager Version 7	37
Installation und Konfiguration: Ergänzung	39
Kapitel 5. Installieren von DB2-Clients auf UNIX-Betriebssystemen	39
Kernel-Konfigurationsparameter für HP-UX	39
Kapitel 12. Ausführen Ihrer eigenen Anwendungen	39
Binden von Datenbankdienstprogrammen mit Run-Time Client.	39
Client-Zugriff auf DB2 mit ODBC unter UNIX.	39
Kapitel 24. Einrichten eines Systems zusammenschlossener Datenbanken für den Zugriff auf Einstellen eines Systems auf mehrere Datenquellen	39
Systeme zusammenschlossener Datenbanken	39
Einschränkung	40
Installieren von DB2 Relational Connect	40
Installieren von DB2 Relational Connect auf Windows NT-Servern	40
Installieren von DB2 Relational Connect auf Servern mit AIX, Linux und der Solaris-Betriebsumgebung	41
Kapitel 26. Zugriff auf Oracle-Datenquellen.	42
Fehler in der Dokumentation	43
Zugriff auf Sybase-Datenquellen (neues Kapitel)	44
Hinzufügen von Sybase-Datenquellen zu einem Server mit zusammenschlossenen Datenbanken	44
Schritt 1: Festlegen der Umgebungsvariablen und Aktualisieren der Profilregistrierdatenbank.	44
Schritt 2: Verbinden von DB2 mit der Sybase-Client-Software (nur AIX und Solaris)	46

Schritt 3: DB2-Exemplar erneut starten	46
Schritt 4: Erstellen und Definieren einer Schnittstellendatei	46
Schritt 5: Erstellen der Oberfläche	46
Schritt 6: Optional: Festlegen der Umgebungsvariablen DB2_DJ_COMM	47
Schritt 7: Erstellen des Servers	47
Optional: Schritt 8: Festlegen der Serveroption CONNECTSTRING.	48
Schritt 9: Erstellen einer Benutzerzuordnung	49
Schritt 10: Erstellen von Kurznamen für Tabellen und Sichten	49
Angeben von Sybase-Codepages	50
Zugriff auf Microsoft SQL Server-Datenquellen mit ODBC (neues Kapitel).	51
Hinzufügen von Microsoft SQL Server-Datenquellen zu einem Server mit zusammenschlossenen Datenbanken.	51
Schritt 1: Festlegen der Umgebungsvariablen (nur AIX)	52
Schritt 2: Ausführen des Shell-Scripts (nur AIX).	52
Schritt 3: Optional: Festlegen der Umgebungsvariablen DB2_DJ_COMM	53
Schritt 4: DB2-Exemplar erneut starten (nur AIX).	53
Schritt 5: Erstellen der Oberfläche	53
Schritt 6: Erstellen des Servers	54
Schritt 7: Erstellen einer Benutzerzuordnung	55
Schritt 8: Erstellen von Kurznamen für Tabellen und Sichten	55
Schritt 9: Optional: Abrufen von ODBC-Traces	56
Übersicht über Microsoft SQL Server-Codepages	56

Allgemeine Installationsinformationen

Herunterladen von Installationspaketen für alle unterstützten DB2-Clients

Wenn Sie Installationspakete für alle unterstützten DB2-Clients, einschließlich der Versionen vor 7, herunterladen wollen, rufen Sie die Website für IBM DB2 Client Application Enabler Pack unter folgender Adresse auf:
<http://www.ibm.com/software/data/db2/db2tech/clientpak.html>

Installieren von DB2 unter Windows 2000

Wenn Sie unter Windows 2000 über eine ältere Version von DB2 installieren oder die aktuelle Version erneut installieren, stellen Sie sicher, dass die Wiederherstellungsoptionen für alle DB2-Services auf "Take No Action" eingestellt sind.

Migrationsproblem bei mit Sonderregistern definierten Sichten

Sichten werden nach einer Datenbankmigration unbrauchbar, wenn das Sonderregister USER oder CURRENT SCHEMA für die Definition einer Sichtspalte verwendet wird. Beispiel:

```
create view v1 (c1) as values user
```

In Version 5 hatten USER und CURRENT SCHEMA den Datentyp CHAR(8). Seit Version 6 werden sie jedoch als VARCHAR(128) definiert. In diesem Beispiel hat Spalte c1 beim Erstellen der Sicht in Version 5 den Datentyp CHAR. Dieser Datentyp wird auch nach der Datenbankmigration beibehalten. Wenn die Sicht nach der Migration verwendet wird, wird sie zur Laufzeit kompiliert, schlägt dann jedoch fehl, weil die Datentypen nicht übereinstimmen.

Diese Sicht muss gelöscht und dann neu erstellt werden. Bevor Sie die Sicht löschen, müssen Sie die Syntax, durch die sie erstellt wurde, durch Abfrage der Katalogsicht SYSCAT.VIEWS erfassen. Beispiel:

```
select text from syscat.views where viewname='<>'
```

IPX/SPX-Protokollunterstützung unter Windows 2000

Diese Informationen beziehen sich auf den Abschnitt zu möglichen Client-Server-Konnektivitätsszenarios im Kapitel zur Installationsplanung des Handbuchs „Einstieg“.

Die veröffentlichte Tabelle für die Protokollunterstützung ist nicht ganz korrekt. Ein Windows 2000-Client mit einer Verbindung zu einem OS/2- oder UNIX-basierten Server über IPX/SPX wird nicht unterstützt. Desgleichen werden OS/2- oder UNIX-basierte Clients mit einer Verbindung zu einem Windows 2000-Server über IPX/SPX nicht unterstützt.

Stoppen von DB2-Prozessen vor dem Upgrade älterer Versionen von DB2

Diese Informationen beziehen sich auf die Migrationsinformationen in Ihrem Handbuch „Einstieg“ von DB2 für Windows.

Wenn Sie einen Upgrade von einer älteren Version von DB2 ausführen, die auf Ihrer Windows-Maschine aktiv ist, gibt das Installationsprogramm eine Warnung mit einer Liste von Prozessen aus, die DB2-DLLs im Speicher haben. Sie können dann die Prozesse in der Liste manuell stoppen oder automatisch vom Installationsprogramm stoppen lassen. Es empfiehlt sich, alle DB2-Prozesse vor der Installation manuell zu stoppen, um Datenverluste zu vermeiden. Die beste Möglichkeit, um sicherzustellen, dass keine DB2-Prozesse mehr aktiv sind, ist die Anzeige Ihrer Systemprozesse im Fenster **Dienste** von Windows. Stellen Sie im Fenster **Dienste** von Windows sicher, dass keine DB2-, OLAP- oder Data Warehouse-Dienste mehr aktiv sind.

Anmerkung: Auf Windows-Plattformen kann jeweils nur eine Version von DB2 ausgeführt werden. Sie können beispielsweise nicht DB2 Version 7 und DB2 Version 6 auf derselben Windows-Maschine ausführen. Wenn Sie DB2 Version 7 auf einer Maschine installieren, auf der bereits DB2 Version 6 installiert ist, wird bei der Installation DB2 Version 6 gelöscht. Weitere Informationen zur Migration von älteren Versionen von DB2 finden Sie im entsprechenden Handbuch *Einstieg*.

Ausführen von db2iupdt nach der Installation von DB2, wenn ein anderes DB2-Produkt bereits installiert ist

Die folgenden Informationen sind auch im Einstiegshandbuch zur Installation enthalten.

Wenn Sie DB2 UDB Version 7 auf UNIX-basierten Systemen installieren und bereits ein DB2-Produkt installiert ist, müssen Sie den Befehl **db2iupdt** ausführen, um die Exemplare zu aktualisieren, mit denen Sie die neuen Funktionen dieses Produkts verwenden wollen. Einige Funktionen werden erst verfügbar, nachdem dieser Befehl ausgeführt wurde.

Definieren der Linux-Umgebung zur Ausführung der DB2-Steuerzentrale

Diese Informationen finden Sie im Kapitel zur Installation der DB2-Steuerzentrale im Handbuch „Einstieg“.

Nachdem das DB2-Installationsprogramm unter Linux beendet ist und Sie zum Terminalfenster zurückgekehrt sind, geben Sie die folgenden Befehle ein, um die korrekte Umgebung zur Ausführung der DB2-Steuerzentrale zu definieren:

```
su -l <exemplarname>
export JAVA_HOME=/usr/jdk118
export DISPLAY=<ihr maschinename>:0
```

Öffnen Sie dann ein weiteres Terminalfenster und geben Sie Folgendes ein:

```
su root
xhost +<ihr maschinename>
```

Schließen Sie diese Terminalfenster und kehren Sie zum Terminalfenster zurück, in dem Sie sich mit der Exemplareigner-ID angemeldet haben. Geben Sie dort den folgenden Befehl ein:

```
db2cc
```

Damit wird die Steuerzentrale gestartet.

DB2 Universal Database Enterprise Edition und DB2 Connect Enterprise Edition für Linux unter S/390

DB2 Universal Database Enterprise Edition und DB2 Connect Enterprise Edition sind jetzt für Linux unter S/390 verfügbar. Vor der Installation von Linux auf einer S/390-Maschine sollten Sie die Software- und Hardwarevoraussetzungen kennen:

Hardware

S/390 9672 Generation 5 oder höher, Multiprise 3000

Software

- SuSE Linux Version 7.0 für S/390 oder Turbolinux Server 6 für zSeries und S/390
- Kernel-Stufe 2.2.16 mit Programmkorrekturen für S/390 (siehe unten)
- glibc 2.1.3
- libstdc++ 6.1

Folgende Programmkorrekturen sind für Linux unter S/390 erforderlich:

- Momentan sind keine Programmkorrekturen erforderlich.

Aktuelle Aktualisierungen finden Sie auf der Website
<http://www.software.ibm.com/data/db2/linux>.

Anmerkungen:

1. Nur 32-Bit Intel-basiertes Linux und Linux auf S/390 werden unterstützt.
2. Folgendes ist unter Linux/390 in DB2 Version 7 *nicht* verfügbar:
 - DB2 UDB Enterprise - Extended Edition
 - DB2 Extender
 - Data Links Manager
 - DB2 Administration Client
 - Unterstützung für Kennwortänderung
 - LDAP-Unterstützung

DB2 Universal Database Enterprise - Extended Edition für UNIX Einstieg

In Kapitel 5 zum Installieren und Konfigurieren von DB2 Universal Database unter Linux wird angegeben, dass jeder physische Knoten im Linux-EEE-Cluster dieselbe Kernel-, glibc- und libstdc++-Stufe haben muss.

Eine Testversion von DB2 EEE für Linux kann von der folgenden Website heruntergeladen werden: <http://www6.software.ibm.com/dl/db2udbd1/db2udbd1-p>

Kernel-Parameter "shmseg" für HP-UX

Die Informationen zur Aktualisierung der Kernel-Konfigurationsparameter für HP-UX im Handbuch „Einstieg“ sind nicht korrekt. Der empfohlene Wert für den Kernel-Parameter shmseg für HP-UX sollte ignoriert werden.

Stattdessen sollte der HP-UX-Standardwert (120) verwendet werden.

Migrieren von IBM Visual Warehouse-Steuerungsdatenbanken

Das Handbuch *DB2 Universal Database für Windows Einstieg* enthält Informationen zur Migration der aktiven Warehouse-Steuerungsdatenbank während einer Standardinstallation von DB2 Universal Database Version 7 unter Windows NT und Windows 2000. Wenn Sie mehrere Warehouse-Steuerungsdatenbanken migrieren möchten, müssen Sie dazu das Fenster **Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung** verwenden. Es kann immer nur eine Warehouse-Steuerungsdatenbank aktiv sein. Wenn die zuletzt migrierte Datenbank nicht diejenige ist, die Sie bei der nächsten Anmeldung an der Data Warehouse-Zentrale verwenden wollen, müssen Sie die gewünschte Datenbank im Fenster **Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung** registrieren.

Zugreifen auf Warehouse-Steuerungsdatenbanken

Bei einer Standardinstallation von DB2 Version 7 unter Windows NT wird neben dem Warehouse-Server eine DB2 Version 7-Warehouse-Steuerungsdatenbank erstellt. Bei einer Warehouse-Steuerungsdatenbank von Visual Warehouse müssen Sie zunächst den DB2-Server mit der Warehouse-Steuerungsdatenbank auf DB2 Version 7 aktualisieren, bevor die Metadaten in der Warehouse-Steuerungsdatenbank migriert und von der Data Warehouse-Zentrale von DB2 Version 7 verwendet werden können. Sie müssen alle Warehouse-Steuerungsdatenbanken auf Version 7 migrieren, die Sie weiterhin verwenden möchten. Die Metadaten in Ihrer aktiven Warehouse-Steuerungsdatenbank werden während der Installation von DB2 Version 7 auf Version 7 migriert. Verwenden Sie zum Migrieren weiterer Warehouse-Steuerungsdatenbanken das entsprechende Migrationshilfsprogramm, das Sie unter Windows NT durch folgende Auswahl starten: **Start** —> **Programme** —> **IBM DB2** —> **Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung**. Informationen zur Migration Ihrer Warehouse-Steuerungsdatenbanken finden Sie in *DB2 Universal Database für Windows Einstieg*.

Data Links Manager Einstieg

Fehlschlagen von "dlfm start" mit der Nachricht, dass beim Abruf von afsfid für Präfix ein Fehler aufgetreten ist

Wenn bei einem Data Links Manager in der DCE-DFS-Umgebung der Befehl **dlfm start** mit der folgenden Fehlermeldung fehlschlägt, wenden Sie sich an den IBM Kundendienst:

"Fehler beim Abruf von afsfid für Präfix"

Dieser Fehler kann auftreten, wenn eine DFS-Dateigruppe, die für den Data Links Manager mit "dlfm add_prefix" registriert wurde, gelöscht wurde.

Einrichten einer Tivoli Storage Manager-Verwaltungsklasse für Archivierungsdateien

Setzen Sie den Eintrag `DLFM_TSM_MGMTCLASS` der DB2-Registrierdatenbank auf den entsprechenden Verwaltungsklassennamen, um anzugeben, welche Tivoli Storage Manager-Verwaltungsklasse für Archivierungsdateien verwendet werden soll.

Erforderlicher Plattenspeicherplatz für DFS Client Enabler

DFS Client Enabler ist eine optionale Komponente, die Sie bei der Installation des DB2 Universal Database-Clients oder -Servers auswählen können. Sie können DFS Client Enabler nicht ohne DB2 Universal Database-Client oder -Server installieren, obwohl DFS Client Enabler auch ohne einen DB2 UDB-Client oder -Server eigenständig funktioniert. Neben dem für den Code von DFS Client Enabler erforderlichen Plattenspeicherplatz von 2 MB sollten Sie weitere 40 MB freihalten, wenn Sie DFS Client Enabler im Rahmen der Installation von DB2 Run-Time Client installieren. Sie benötigen mehr Plattenspeicherplatz, wenn Sie DFS Client Enabler im Rahmen der Installation eines DB2 Administration-Clients oder eines DB2-Servers installieren. Weitere Informationen zum erforderlichen Plattenspeicherplatz für DB2 Universal Database-Produkte finden Sie im Handbuch *DB2 für UNIX Einstieg*.

Überwachen der Back-End-Prozesse von Data Links File Manager unter AIX

Die Ausgabe des Befehls **dlfm see** wurde geändert. Wenn Sie diesen Befehl zur Überwachung der Back-End-Prozesse von Data Links File Manager unter AIX absetzen, ähnelt die Ausgabe der nachstehenden Tabelle.

PID	PPID	PGID	RUNAME	UNAME	ETIME	DAEMON NAME
17500	60182	40838	dlfm	root	12:18	dlfm_copyd_(dlfm)
41228	60182	40838	dlfm	root	12:18	dlfm_chownd_(dlfm)
49006	60182	40838	dlfm	root	12:18	dlfm_upcall_(dlfm)
51972	60182	40838	dlfm	root	12:18	dlfm_gcd_(dlfm)
66850	60182	40838	dlfm	root	12:18	dlfm_retrieved_(dlfm)
67216	60182	40838	dlfm	dlfm	12:18	dlfm_deigrpd_(dlfm)
60182	1	40838	dlfm	dlfm	12:18	dlfmd_(dlfm)

Die DLFM SEE-Anforderung war erfolgreich.

Das Name in runden Klammern gibt das Data Links File Manager-Exemplar an, in diesem Fall "dlfm".

Installation und Konfiguration von DB2 Data Links Manager für AIX: Zusätzliche Installationsvoraussetzungen in DCE-DFS-Umgebungen

Im Abschnitt "Installationsvoraussetzungen" müssen folgende neue Informationen hinzugefügt werden:

Außerdem müssen Sie ein e-Fix für DFS 3.1 oder PTF-Set 1 installieren (wenn es verfügbar wird). Das e-Fix ist unter folgender URL verfügbar:
http://www.transarc.com/Support/dfs/datalinks/efix_dfs31_main_page.html

Weitere Informationen:

Der dfs-Client muss aktiv sein, bevor Sie Data Links Manager installieren. Verwenden Sie den Befehl "db2setup" oder "smitty".

Im Abschnitt "Chiffrierschlüsseldatei" ist ein Fehler, der wie folgt korrigiert werden muss:

Diese Chiffrierschlüsseldatei, die Informationen zum Principal und zum Kennwort enthält, sollte den Namen "datalinks.ktb" erhalten und ...

Der korrekte Name `datalink.ktb` wird in dem Beispiel unten verwendet. Der Abschnitt "Chiffrierschlüsseldatei" sollte auf den Abschnitt "Tasks nach Abschluss der Installation in DCE-DFS" folgen, da diese Datei erst erstellt werden kann, nachdem das Exemplar DLMADMIN erstellt wurde.

Im Abschnitt zu den Data Links File Manager-Servern und -Clients müssen Sie darauf achten, den Data Links Manager-Server vor allen Data Links Manager-Clients zu installieren.

fügen Sie einen neuen Abschnitt mit dem Titel "Sicherungsverzeichnis" hinzu:

Wenn die Sicherung auf einem lokalen Dateisystem erfolgt, muss sich das Verzeichnis auf einem DFS-Dateisystem befinden. Stellen Sie sicher, dass diese DFS-Dateigruppe von einem DFS-Administrator erstellt wurde. Dies sollte keine DMLFS-Gruppe sein.

Fehlgeschlagener Befehl "dlfm add_prefix"

Bei einem in der DCE/DFS-Umgebung ausgeführten Data Links Manager führt der Befehl `dlfm add_prefix` möglicherweise zu einem Fehler mit Rückkehrcode -2061 (Sicherung ist fehlgeschlagen). Führen Sie in diesem Fall folgende Schritte aus:

1. Stoppen Sie durch Absetzen des Befehls `dlfm stop` die Data Links Manager-Dämonprozesse.
2. Stoppen Sie durch Absetzen des Befehls `dlfm stopdbm` die DB2-Prozesse.
3. Rufen Sie durch Absetzen des Befehls `dce_login root` die DCE-Root-Berechtigungen ab.
4. Starten Sie durch Absetzen des Befehls `dlfm startdbm` die DB2-Prozesse.
5. Registrieren Sie durch Absetzen des Befehls `dlfm add_prefix` die Dateigruppe für Data Links Manager.
6. Starten Sie durch Absetzen des Befehls `dlfm start` die Data Links Manager-Dämonprozesse.

Installation und Konfiguration von DB2 Data Links Manager für AIX: Installieren von DB2 Data Links Manager unter AIX mit dem Dienstprogramm db2setup

Im Abschnitt "Erstellen der DB2-Datenbank DLFM_DB" wird DLFM_DB nicht in der DCE_DFS-Umgebung erstellt. Dies muss als Schritt nach Abschluss der Installation geschehen.

Im Abschnitt „DMAPP-Registrierung vor dem Start in DCE-DFS“ muss Schritt 2 folgendermaßen geändert werden:

2. Der Datei /opt/dcelocal/tcl/user_cmd.tcl werden Befehle hinzugefügt, um sicherzustellen, dass DMAPP beim Start von DFS gestartet wird.

Installation und Konfiguration von DB2 Data Links Manager für AIX: Tasks nach Abschluss der Installation in DCE-DFS

Fügen Sie den folgenden Abschnitt "Abschließen der Installation von Data Links Manager" neu hinzu:

Auf dem Data Links Manager-Server müssen Sie folgende Schritte auszuführen, um die Installation abzuschließen:

1. Erstellen Sie die Chiffrierschlüsseldatei entsprechend der Beschreibung unter "Chiffrierschlüsseldatei" im Abschnitt "Zusätzliche Installationsvoraussetzungen in DCE-DFS-Umgebungen" im Kapitel "Installation und Konfiguration von DB2 Data Links Manager für AIX".
2. Geben Sie als Root folgende Befehle zum Starten von DMAPP ein:

```
stop.dfs all  
start.dfs all
```

3. Führen Sie "dlfm setup" unter Verwendung der DCE-Root-Berechtigungen wie folgt aus:
 - a. Melden Sie sich als Data Links Manager-Administrator (DLMADMIN) an.
 - b. Setzen Sie als Root **dce_login** ab.
 - c. Geben Sie den Befehl **dlfm setup** ein.

Auf dem Data Links Manager-Client müssen Sie folgende Schritte auszuführen, um die Installation abzuschließen:

1. Erstellen Sie die Chiffrierschlüsseldatei entsprechend der Beschreibung unter "Chiffrierschlüsseldatei" im Abschnitt "Zusätzliche Installationsvoraussetzungen in DCE-DFS-Umgebungen" im Kapitel "Installation und Konfiguration von DB2 Data Links Manager für AIX".
2. Geben Sie als Root folgende Befehle zum Starten von DMAPP ein:

```
stop.dfs all  
start.dfs all
```

Installation und Konfiguration von DB2 Data Links Manager für AIX: Manuelles Installieren von DB2 Data Links Manager mit SMIT

Modifizieren Sie im Abschnitt „Tasks nach Abschluss der Installation mit SMIT“ Schritt 7 und geben Sie an, dass der Befehl `dce_login root` vor `dlfm setup` abgesetzt werden muss. Schritt 11 ist nicht erforderlich. Dieser Schritt wird automatisch in Schritt 6 (`dlfm server_conf`) oder in Schritt 8 (`dlfm client_conf`) ausgeführt. Entfernen Sie außerdem Schritt 12 (`dlfm start`). Führen Sie zum Abschluss der Installation folgende Schritte aus:

1. Erstellen Sie die Chiffrierschlüsseldatei entsprechend der Beschreibung unter "Chiffrierschlüsseldatei" im Abschnitt "Zusätzliche Installationsvoraussetzungen in DCE-DFS-Umgebungen" im Kapitel "Installation und Konfiguration von DB2 Data Links Manager für AIX".
2. Geben Sie als Root folgende Befehle zum Starten von DMAPP ein:

```
stop.dfs all
start.dfs all
```

Installation und Konfiguration von DB2 Data Links DFS Client Enabler

Fügen Sie dem Schritt 2 im Abschnitt "Konfigurieren von DFS Client Enabler" folgende Informationen hinzu:

Mit der Ausführung der "secval"-Befehle wird normalerweise die Installation abgeschlossen. Möglicherweise muss jedoch außerdem ein Warmstart der Maschine durchgeführt werden. Wenn Probleme beim Zugriff auf READ PERMISSION DB-Dateien auftreten, führen Sie einen Warmstart der Maschine aus, auf welcher DB2 DFS Client Enabler gerade installiert worden ist.

Installation und Konfiguration von DB2 Data Links Manager für Solaris

Folgende Aktionen müssen nach der Installation von DB2 Data Links Manager für Solaris ausgeführt werden:

1. Fügen Sie folgende drei Zeilen der Datei `/etc/system` hinzu:

```
set dlfsdrv:glob_mod_pri=0x100800
set dlfsdrv:glob_mesg_pri=0xff
set dlfsdrv:ConfigDlfsUid=UID
```

UID ist die Benutzer-ID der ID `dlfm`.

2. Führen Sie einen Warmstart der Maschine durch, um die Änderungen zu aktivieren.

Auswählen einer Sicherungsmethode für DB2 Data Links Manager unter AIX

Neben Disk Copy und XBSA können Sie Tivoli Storage Manager (TSM) zum Sichern von Dateien verwenden, die sich auf einem Data Links-Server befinden.

Gehen Sie wie folgt vor, um Tivoli Storage Manager als Archivierungs-Server zu verwenden:

1. Installieren Sie Tivoli Storage Manager auf einem Data Links-Server. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu Tivoli Storage Manager.

2. Registrieren Sie die Client-Anwendung des Data Links-Servers auf dem Tivoli Storage Manager-Server. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu Tivoli Storage Manager.
3. Fügen Sie den Script-Dateien `db2profile` oder `db2cshrc` von Data Links Manager Administrator folgende Umgebungsvariablen hinzu:


```
(für Bash-, Bourne- oder Korn-Shell)
export DSMI_DIR=/usr/tivoli/tsm/client/api/bin
export DSMI_CONFIG=$HOME/tsm/dsm.opt
export DSMI_LOG=$HOME/dldump
export PATH=$PATH:$DSMI_DIR

(für C-Shell)
setenv DSMI_DIR /usr/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:$DSMI_DIR
```
4. Stellen Sie sicher, dass sich die TSM-Standardvorgabendatei `dsm.sys` im Verzeichnis `$DSMI_DIR` befindet.
5. Stellen Sie sicher, dass sich die TSM-Benutzeroptionsdatei `dsm.opt` im Verzeichnis `INSTHOME/tsm` befindet. Dabei ist `INSTHOME` das Ausgangsverzeichnis von Data Links Manager Administrator.
6. Setzen Sie die Option `PASSWORDACCESS` in der TSM-Standardvorgabendatei `/usr/tivoli/tsm/client/api/bin/dsm.sys` auf `generate`.
7. Registrieren Sie das TSM-Kennwort mit der Option `generate`, *bevor* Sie Data Links File Manager zum ersten Mal starten. Auf diese Weise müssen Sie kein Kennwort eingeben, wenn Data Links File Manager eine Verbindung zum TSM-Server initiiert. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu TSM.
8. Setzen Sie die Registrierungsvariable `DLFM_BACKUP_TARGET` auf TSM. Der Wert der Registrierungsvariablen `DLFM_BACKUP_DIR_NAME` wird in diesem Fall ignoriert. Dadurch wird die Sicherungsoption von Tivoli Storage Manager aktiviert.

Anmerkungen:

- a. Wenn Sie die Einstellung der Registrierungsvariablen `DLFM_BACKUP_TARGET` zwischen TSM und dem Datenträger während der Laufzeit ändern, beachten Sie, dass die Archivierungsdateien nicht an die neu angegebene Archivierungsposition versetzt werden. Wenn beim Start von Data Links File Manager beispielsweise der Wert der Registrierungsvariablen `DLFM_BACKUP_TARGET` auf TSM gesetzt ist und Sie den Wert in eine Position auf dem Datenträger ändern, werden die Archivierungsdateien an der neuen Position auf dem Datenträger gespeichert. Die zuvor auf TSM archivierten Dateien werden nicht an die neue Position auf dem Datenträger versetzt.
 - b. Mithilfe der neuen Registrierungsvariablen `DLFM_TSM_MGMTCLASS` können Sie die TSM-Standardverwaltungsklasse überschreiben. Wenn diese Registrierungsvariable nicht gesetzt wird, wird die TSM-Standardverwaltungsklasse verwendet.
9. Stoppen Sie Data Links File Manager durch die Eingabe des Befehls **dlfm stop**.
 10. Starten Sie Data Links File Manager durch die Eingabe des Befehls **dlfm start**.

Auswählen einer Sicherungsmethode für DB2 Data Links Manager in der Solaris-Betriebsumgebung

Neben Disk Copy und XBSA können Sie Tivoli Storage Manager (TSM) zum Sichern von Dateien verwenden, die sich auf einem Data Links-Server befinden.

Gehen Sie wie folgt vor, um Tivoli Storage Manager als Archivierungsserver zu verwenden:

1. Installieren Sie Tivoli Storage Manager auf einem Data Links-Server. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu Tivoli Storage Manager.
2. Registrieren Sie die Client-Anwendung des Data Links-Servers auf dem Tivoli Storage Manager-Server. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu Tivoli Storage Manager.
3. Fügen Sie den Script-Dateien `db2profile` oder `db2cshrc` von Data Links Manager Administrator folgende Umgebungsvariablen hinzu:

```
(für Bash-, Bourne- oder Korn-Shell)
export DSMI_DIR=/opt/tivoli/tsm/client/api/bin
export DSMI_CONFIG=${HOME}/tsm/dsm.opt
export DSMI_LOG=${HOME}/dldump
export PATH=$PATH:/opt/tivoli/tsm/client/api/bin
```

```
(für C-Shell)
setenv DSMI_DIR /opt/tivoli/tsm/client/api/bin
setenv DSMI_CONFIG ${HOME}/tsm/dsm.opt
setenv DSMI_LOG ${HOME}/dldump
setenv PATH=${PATH}:/opt/tivoli/tsm/client/api/bin
```

4. Stellen Sie sicher, dass sich die TSM-Standardvorgabendatei `dsm.sys` im Verzeichnis `/opt/tivoli/tsm/client/api/bin` befindet.
5. Stellen Sie sicher, dass sich die TSM-Benutzeroptionsdatei `dsm.opt` im Verzeichnis `INSTHOME/tsm` befindet. Dabei ist `INSTHOME` das Ausgangsverzeichnis von Data Links Manager Administrator.
6. Setzen Sie die Option `PASSWORDACCESS` in der TSM-Standardvorgabendatei `/opt/tivoli/tsm/client/api/bin/dsm.sys` auf `generate`.
7. Registrieren Sie das TSM-Kennwort mit der Option `generate`, *bevor* Sie Data Links File Manager zum ersten Mal starten. Auf diese Weise müssen Sie kein Kennwort eingeben, wenn Data Links File Manager eine Verbindung zum TSM-Server einleitet. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu TSM.
8. Setzen Sie die Registrierungsvariable `DLFM_BACKUP_TARGET` auf TSM. Der Wert der Registrierungsvariablen `DLFM_BACKUP_DIR_NAME` wird in diesem Fall ignoriert. Dadurch wird die Sicherungsoption von Tivoli Storage Manager aktiviert.

Anmerkungen:

- a. Wenn Sie die Einstellung der Registrierungsvariablen `DLFM_BACKUP_TARGET` zwischen TSM und dem Datenträger während der Laufzeit ändern, beachten Sie, dass die Archivierungsdateien nicht an die neu angegebene Archivierungsposition versetzt werden. Wenn beim Start von Data Links File Manager beispielsweise der Wert der Registrierungsvariablen `DLFM_BACKUP_TARGET` auf TSM gesetzt ist und Sie den Wert in eine Position auf dem Datenträger ändern, werden die Archivierungsdateien an die neue Position auf dem Datenträger versetzt. Die zuvor auf TSM archivierten Dateien werden nicht an die neue Position auf dem Datenträger versetzt.

- b. Mithilfe der neuen Registrierungsvariablen DLFM_TSM_MGMTCLASS können Sie die TSM-Standardverwaltungsklasse überschreiben. Wenn diese Registrierungsvariable nicht gesetzt wird, wird die TSM-Standardverwaltungsklasse verwendet.
9. Stoppen Sie Data Links File Manager durch die Eingabe des Befehls **dlfm stop**.
10. Starten Sie Data Links File Manager durch die Eingabe des Befehls **dlfm start**.

Auswählen einer Sicherungsmethode für DB2 Data Links Manager unter Windows NT

Immer wenn ein DATALINK-Wert in eine Tabelle mit einer DATALINK-Spalte eingefügt wird, die für die Wiederherstellung definiert ist, werden die entsprechenden DATALINK-Dateien zur Sicherung auf einem Archivierungs-Server terminiert. Momentan werden die zwei Optionen Disk Copy (Standardmethode) und Tivoli Storage Manager für die Dateisicherung auf einem Archivierungs-Server unterstützt. In künftigen Releases von DB2 Data Links Manager für Windows NT werden auch Sicherungsdatenträger und -Software anderen Unternehmen unterstützt.

Disk Copy (Standardmethode)

Wenn der Befehl **backup** auf dem DB2-Server eingegeben wird, stellt er sicher, dass die verbundenen Dateien in der Datenbank gesichert werden. Die Sicherung erfolgt auf dem Data Links-Server in dem Verzeichnis, das in der Umgebungsvariablen DLFM_BACKUP_DIR_NAME angegeben wurde. Der Standardwert für diese Variable ist c:\dlfmbackup. Dabei steht c:\ für das Sicherungsinstallationslaufwerk von Data Links Manager.

Geben Sie folgenden Befehl ein, um diese Variable auf c:\dlfmbackup zu setzen:

```
db2set -g DLFM_BACKUP_DIR_NAME=c:\dlfmbackup
```

Die durch die Umgebungsvariable DLFM_BACKUP_DIR_NAME angegebene Speicherposition darf sich *nicht* auf einem Dateisystem befinden, auf dem ein Data Links Filesystem Filter verwendet wird. Stellen Sie sicher, dass der erforderliche Speicherplatz in dem für die Sicherungsdateien angegebenen Verzeichnis vorhanden ist.

Stellen Sie außerdem sicher, dass die Variable DLFM_BACKUP_TARGET durch die Eingabe des folgenden Befehls auf LOCAL gesetzt wird:

```
db2set -g DLFM_BACKUP_TARGET=LOCAL
```

Nachdem Sie diese Variablen gesetzt oder geändert haben, stoppen Sie Data Links File Manager mit dem Befehl **dlfm stop**, und führen Sie mit dem Befehl **dlfm start** einen Neustart durch.

Tivoli Storage Manager

Gehen Sie wie folgt vor, um Tivoli Storage Manager als Archivierungs-Server zu verwenden:

1. Installieren Sie Tivoli Storage Manager auf einem Data Links-Server. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu Tivoli Storage Manager.
2. Registrieren Sie die Client-Anwendung des Data Links-Servers auf dem Tivoli Storage Manager-Server. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu Tivoli Storage Manager.

- Klicken Sie **Start** an und wählen Sie **Einstellungen** —> **Systemsteuerung** —> **System** aus. Das Fenster **Systemeigenschaften** wird geöffnet. Wählen Sie die Indexzunge **Umgebung** aus und geben Sie die folgenden Umgebungsvariablen und entsprechenden Werte ein:

Variable	Wert
DSMI_DIR	c:\tsm\baclient
DSMI_CONFIG	c:\tsm\baclient\dsm.opt
DSMI_LOG	c:\tsm\dldump

- Stellen Sie sicher, dass sich die TSM-Standardvorgabendatei `dsm.sys` im Verzeichnis `c:\tsm\baclient` befindet.
- Stellen Sie sicher, dass sich die TSM-Benutzeroptionsdatei `dsm.opt` im Verzeichnis `c:\tsm\baclient` befindet.
- Setzen Sie die Option `PASSWORDACCESS` in der TSM-Standardvorgabendatei `c:\tsm\baclient\dsm.sys` auf `generate`.
- Registrieren Sie das TSM-Kennwort mit der Option `generate`, *bevor* Sie Data Links File Manager zum ersten Mal starten. Auf diese Weise müssen Sie kein Kennwort eingeben, wenn Data Links File Manager eine Verbindung zum TSM-Server initiiert. Weitere Informationen finden Sie in Ihrer Produktdokumentation zu TSM.
- Setzen Sie die Umgebungsvariable `DLFM_BACKUP_TARGET` mit folgendem Befehl auf TSM:

```
db2set -g DLFM_BACKUP_TARGET=TSM
```

Der Wert der Umgebungsvariablen `DLFM_BACKUP_DIR_NAME` wird in diesem Fall ignoriert. Dadurch wird die Sicherungsoption von Tivoli Storage Manager aktiviert.

Anmerkungen:

- Wenn Sie die Einstellung der Umgebungsvariablen `DLFM_BACKUP_TARGET` zwischen TSM und LOCAL während der Laufzeit ändern, beachten Sie, dass die Archivierungsdateien nicht an die neu angegebene Archivierungsposition versetzt werden. Wenn beim Start von Data Links File Manager beispielsweise der Wert der Umgebungsvariablen `DLFM_BACKUP_TARGET` auf TSM gesetzt ist und Sie den Wert in LOCAL ändern, werden die Archivierungsdateien an die neue Position auf dem Datenträger versetzt. Die zuvor auf TSM archivierten Dateien werden nicht an die neue Position auf dem Datenträger versetzt.
 - Mithilfe der neuen Umgebungsvariablen `DLFM_TSM_MGMT_CLASS` können Sie die TSM-Standardverwaltungsklasse überschreiben. Wenn diese Variable nicht gesetzt wird, wird die TSM-Standardverwaltungsklasse verwendet.
- Stoppen Sie Data Links File Manager durch die Eingabe des Befehls **dlfm stop**.
 - Starten Sie Data Links File Manager durch die Eingabe des Befehls **dlfm start**.

Sichern von JFS (Journaled File System) unter AIX

Im Handbuch wird gesagt, dass Data Links Manager gestoppt werden muss und auf dem Dateisystem eine Offlinesicherung durchgeführt werden soll. Der folgende Ansatz, bei dem Data Links Manager nicht gestoppt werden muss, richtet sich an Benutzer, die höhere Verfügbarkeit benötigen.

1. Greifen Sie auf die CLI-Quellendatei `quiesce.c` und das Shell-Script `online.sh` zu. Diese Dateien befinden sich im Verzeichnis `/samples/dl_fm`.
2. Kompilieren Sie `quiesce.c`:

```
xlc -o quiesce -L$HOME/sqllib/lib -I$HOME/sqllib/include -c quiesce.c
```
3. Führen Sie das Script als Root auf dem Knoten mit dem DLFS-Dateisystem aus.

Das Shell-Script `online.sh` geht davon aus, dass für jede bei Data Link Manager registrierte Datenbank ein Katalogeintrag auf dem Data Link Manager-Knoten vorhanden ist. Außerdem geht es davon aus, dass `/etc/filesystems` den vollständigen Eintrag für das DLFS-Dateisystem enthält. Das Shell-Script führt folgende Aufgaben aus:

- Versetzt alle Tabellen in Datenbanken in den Wartemodus, die bei Data Links Manager registriert sind. Dadurch werden alle neuen Aktivitäten gestoppt.
- Hängt das Dateisystem ab und hängt es als schreibgeschütztes Dateisystem wieder an.
- Führt eine Sicherung des Dateisystems aus.
- Hängt das Dateisystem ab und hängt es als Dateisystem mit Schreib-/Lesezugriff wieder an.
- Setzt die DB2-Tabellen wieder aus dem internen Wartestatus zurück.

Sie müssen das Script folgendermaßen an Ihre Umgebung anpassen:

1. Wählen Sie den Sicherungsbefehl aus und geben Sie die Funktion **`do_backup`** des Scripts an.
2. Setzen Sie im Script folgende Umgebungsvariablen:
 - Setzen Sie `DLFM_INST` auf den DLFM-Exemplarnamen.
 - Setzen Sie `PATH_OF_EXEC` auf den Pfad, in dem sich die ausführbare Datei zum Versetzen in den Wartemodus befindet.

Rufen Sie das Script wie folgt auf:

```
online.sh <name-des-dateisystems>
```

Administratorgruppenzugriffsrechte in Data Links unter Windows NT

Unter Windows NT hat ein Benutzer, der zur Administratorgruppe gehört, für die meisten Funktionen dieselben Zugriffsrechte in Bezug auf Dateien, die mit Hilfe von Data Links verbunden sind, wie ein Root unter UNIX. Die folgende Tabelle enthält einen Vergleich.

Operation	Unix (Root)	Windows NT (Administrator)
Umbenennen	Ja	Ja
Dateizugriff ohne Token	Ja	Ja
Löschen	Ja	Nein (siehe Anm. unten)
Aktualisieren	Ja	Nein (siehe Anm. unten)

Anmerkung: Das NTFS gestattet diese Operationen nicht für eine schreibgeschützte Datei. Der Administrator kann diese Operationen erfolgreich ausführen, indem er den Schreibzugriff für die Datei aktiviert.

Minimieren der Protokollierung für DLFF-Installation (Data Links File System Filter)

Sie können die Protokollierung für die DLFF-Installation (Data Links File System Filter) durch eine Änderung der Datei `dlfs_cfg` minimieren. Die Datei `dlfs_cfg` wird an die Routine `strload` übermittelt, um die Treiber- und Konfigurationsparameter zu laden. Die Datei befindet sich im Verzeichnis `/usr/lpp/db2_07_01/cfg/`. Durch eine symbolische Verbindung kann sich die Datei auch im Verzeichnis `/etc` befinden. Die Datei `dlfs_cfg` hat folgendes Format:

```
d 'treibername' 'vfs-nummer' 'dlfm-id' 'globale-nachrichtenpriorität'
  'globale-modulpriorität' - 0 1
```

Dabei gilt Folgendes:

d Der Parameter `d` gibt an, dass der Treiber geladen werden soll.

treibername

„treibername“ ist der vollständige Pfad des zu ladenden Treibers. Der vollständige Pfad für DB2 Version 7 ist beispielsweise `/usr/lpp/db2_07_01/bin/dlfsdrv`. Der Name des Treibers lautet `dlfsdrv`.

vfs-nummer

Dies ist der `vfs`-Eintrag für DLFS in `/etc/vfs`.

dlfm-id

Dies ist die Benutzer-ID des Data Links Manager-Administrators.

globale-nachrichtenpriorität

Die globale Nachrichtenpriorität.

globale-modulpriorität

Die globale Modulpriorität.

0 1 sind die untergeordneten Nummern für die Erstellung nicht geklonter Knoten für diesen Treiber. Die Erstellung der Knotennamen erfolgt durch Anfügen der untergeordneten Nummer an den Knotennamen des geklonen Treibers. Maximal fünf untergeordnete Nummern können angegeben werden (0-4).

Ein realitätsnahes Beispiel könnte wie folgt aussehen:

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,255,-1 - 0 1
```

Die protokollierten Nachrichten sind abhängig von den Einstellungen für die globale Nachrichtenpriorität und die globale Modulpriorität. Sie können den Wert der globalen Nachrichtenpriorität ändern, um das Protokollieren zu minimieren.

Sie können vier Nachrichtenprioritätswerte verwenden:

```
#define LOG_EMERGENCY    0x01
#define LOG_TRACING      0x02
#define LOG_ERROR        0x04
#define LOG_TROUBLESHOOT 0x08
```

Die meisten Nachrichten in DLFF haben LOG_TROUBLESHOOT als Nachrichtenpriorität. Hier ein paar alternative Konfigurationsbeispiele:

Wenn Sie Ausnahme- und Fehlermeldungen benötigen, legen Sie für die globale Nachrichtenpriorität 5 (1+4) in der Konfigurationsdatei dlfs_cfg fest.

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,5,-1 - 0 1
```

Wenn nur Fehlermeldungen erforderlich sind, legen Sie 4 für die globale Nachrichtenpriorität fest:

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,4,-1 - 0 1
```

Benötigen Sie keine Protokollierung für DLFS, legen Sie 0 für die globale Nachrichtenpriorität fest:

```
d /usr/lpp/db2_07_01/bin/dlfsdrv 14,208,0,-1 - 0 1
```

Protokollieren von Nachrichten nach der Installation

Wenn Sie nach der Installation Ausnahme-, Fehler- und Fehlerbehebungsnachrichten protokollieren müssen, müssen Sie die Datei dlfs_cfg ändern. Die Datei dlfs_cfg befindet sich im Verzeichnis /usr/lpp/db2_07_01/cfg. Für die globale Nachrichtenpriorität muss 255 (maximale Priorität) oder 13 (8+4+1) angegeben werden. Bei der Einstellung 13 (8+4+1) werden Ausnahme-, Fehler- und Fehlerbehebungsnachrichten protokolliert.

Nach der Definition der globalen Nachrichtenpriorität müssen Sie das DLFS-Filterdateisystem abhängen und den Treiber dlfsdrv erneut laden, damit die neuen Prioritätswerte beim Laden festgelegt werden. Nach dem erneuten Laden des Treibers dlfsdrv muss das DLFS-Filterdateisystem wieder angehängt werden.

Anmerkung: Die Einstellungen für dlfs_cfg bleiben für nachfolgende Ladevorgänge des Treibers dlfsdrv aktiv, bis die Datei dlfs_cfg wieder geändert wird.

Deinstallation von DLFM-Komponenten mit SMIT entfernt unter Umständen zusätzliche Dateigruppen

Vor der Deinstallation von DB2 (Version 5, 6 oder 7) von einer AIX-Maschine, auf der Data Links Manager installiert ist, führen Sie folgende Schritte aus:

1. Als Root erstellen Sie eine Kopie von /etc/vfs mit folgendem Befehl:

```
cp -p /etc/vfs /etc/vfs.bak
```
2. Deinstallieren Sie DB2.
3. Als Root ersetzen Sie /etc/vfs durch die in Schritt 1 erstellte Sicherungskopie:

```
cp -p /etc/vfs.bak /etc/vfs
```

Vorbereitung/Feststellen des Host-Namens

Sie müssen die Namen all Ihrer DB2-Server und Data Links-Server ermitteln. Diese Host-Namen müssen Sie zur Prüfung der Installation kennen. Wenn Sie eine Verbindung zu einem DB2 Data Links File Manager herstellen, sendet der DB2 UDB-Server intern die folgenden Informationen an den DLFM:

- Datenbankname
- Exemplarname
- Host-Name

Der DLFM vergleicht dann diese Informationen mit seinen internen Tabellen, um festzustellen, ob die Verbindung zulässig ist. Er lässt die Verbindung nur zu, wenn diese Kombination aus Datenbankname, Exemplarname und Host-Name für ihn mit dem Befehl **dlfm add_db** registriert wurde. Der Host-Name, der im Befehl **dlfm add_db** verwendet wird, muss exakt dem Host-Namen entsprechen, der intern vom DB2 UDB-Server gesendet wird.

Verwenden Sie den exakten Host-Namen, der wie folgt abgerufen werden kann:

1. Geben Sie den Befehl **hostname** auf Ihrem DB2-Server ein. Dieser Befehl könnte z. B. db2server zurückgeben.
2. Führen Sie je nach Plattform einen der folgenden Schritte aus:
 - Unter AIX: Geben Sie den Befehl **host db2server** ein, wobei "db2server" der im vorherigen Schritt abgerufene Name ist. Dieser Befehl sollte eine Ausgabe ähnlich der Folgenden erzeugen:

```
db2server.services.com is 9.11.302.341, Aliases: db2server
```
 - Unter Windows NT: Geben Sie den Befehl **nslookup db2server** ein, wobei "db2server" der im vorherigen Schritt abgerufene Name ist. Dieser Befehl sollte eine Ausgabe ähnlich der Folgenden erzeugen:

```
Server: dnsserv.services.com
Address: 9.21.14.135
Name: db2server.services.com
Address: 9.21.51.178
```
 - Unter Solaris: Geben Sie **cat /etc/hosts | grep 'hostname'** ein. Dieser Befehl sollte eine Ausgabe ähnlich der Folgenden erzeugen, wenn der Host-Name ohne einen Domännennamen in /etc/hosts angegeben wird:

```
9.112.98.167 db2server loghost
```

Wenn der Host-Name mit einem Domännennamen angegeben wird, erzeugt dieser Befehl eine Ausgabe ähnlich der Folgenden:

```
9.112.98.167 db2server.services.com loghost
```

Verwenden Sie "db2server.services.com" als Host-Namen, wenn Sie eine DB2 UDB-Datenbank mit dem Befehl `dlfm add_db` registrieren. Die internen Verbindungen des DB2-Servers zum DLFM schlagen fehl, wenn in diesem Befehl andere Aliasnamen verwendet werden.

Ein Data Links-Server wird mit dem DB2-Befehl "add datalinks manager for datenbank datenbank_alias using node host_name port anschlussnummer" in einer DB2-Datenbank registriert.

Der Host-Name ist der Name des Data Links-Servers. In diesem Befehl kann jeder gültige Aliasname des Data Links-Servers verwendet werden. DATALINK-Werte, die Verweise auf diesen Data Links-Server sind, müssen den Host-Namen im URL-Wert angeben, d. h. der exakte Name, der im Befehl "add datalinks manager" verwendet wurde, muss beim Zuordnen von URL-Werten zu DATALINK-Spalten verwendet werden. Wird ein anderer Aliasname verwendet, schlägt die SQL-Anweisung fehl.

Arbeiten mit Data Links File Manager: Bereinigung nach dem Löschen von DB2 Data Links Manager aus einer DB2-Datenbank

Wenn ein DB2 Data Links Manager mit dem Befehl `DROP DATALINKS MANAGER` aus einer Datenbank gelöscht wird, bereinigt der Befehl selbst nicht die entsprechenden Informationen zum DB2 Data Links Manager. Benutzer können explizit die Aufhebung der Verbindung von mit der Datenbank verknüpften Dateien und die Garbage Collection von Sicherungsinformationen einleiten. Dies erfolgt mit dem Befehl `dlfm drop_dlm`. Dieser Befehl leitet die asynchrone Löschung aller Informationen für eine bestimmte Datenbank ein. DB2 Data Links Manager muss aktiv sein, damit dieser Befehl erfolgreich ausgeführt werden kann. Es ist äußerst wichtig, dass dieser Befehl nur nach dem Löschen eines DB2 Data Links Manager verwendet wird, da sonst wichtige Informationen zum DB2 Data Links Manager verloren gehen und nicht wiederhergestellt werden können.

Gehen Sie wie folgt vor, um die Aufhebung der Verbindung und die Garbage Collection von Sicherungsinformationen für eine bestimmte Datenbank einzuleiten:

1. Melden Sie sich am System als DB2 Data Links Manager-Administrator an.
2. Setzen Sie den folgenden Befehl ab:

```
dlfm drop_dlm datenbank exemplar host-name
```

Dabei gilt Folgendes:

datenbank ist der Name der fernen DB2 UDB-Datenbank.
exemplar ist der Name des Exemplars, in dem die Datenbank sich befindet.
host-name ist der Host-Name des DB2 UDB-Servers, auf dem sich die Datenbank befindet.

3. Melden Sie sich ab.

Ein vollständiges Syntaxszenario, das den Kontext für die Verwendung dieses Befehls zeigt, finden Sie im Handbuch *Command Reference*.

Für diesen Befehl wurde ein neuer Fehlercode generiert (siehe „DLFM1001E (Neue Fehlermeldung)“ auf Seite 34).

DLFM1001E (Neue Fehlernachricht)

DLFM1001E: Fehler bei der Verarbeitung von "drop_dlm".

Ursache:

Der Data Links Manager konnte die Aufhebung der Verbindung und die Garbage Collection für die angegebene Datenbank nicht einleiten. Dies kann eine der folgenden Ursachen haben:

- Der Data Links Manager ist nicht aktiv.
- Im Befehl wurde eine ungültige Kombination aus Datenbank-, Exemplar- und Host-Name angegeben.
- In einem der Komponentenservices des Data Links Manager trat ein Fehler auf.

Aktion:

Führen Sie folgende Schritte aus:

1. Stellen Sie sicher, dass der Data Links Manager aktiv ist. Starten Sie den Data Links Manager, wenn er nicht bereits aktiv ist.
2. Stellen Sie sicher, dass die Kombination aus Datenbank-, Exemplar- und Host-Name eine registrierte Datenbank bezeichnet. Verwenden Sie dazu den Befehl "dlfm list registered databases" des Data Links Manager.
3. Wenn der Fehler weiterhin auftritt, lesen Sie die Informationen in der Datei db2diag.log, um zu sehen, ob Komponentenservices (z. B. der Connection Management Service, der Transaction Management Service usw.) fehlgeschlagen sind. Sehen Sie den Fehlercode in der Datei db2diag.log an, und ergreifen Sie die entsprechenden, unter diesem Fehlercode vorgeschlagenen Maßnahmen.

DLFM-Konfigurationsdateioption

Die Option dlfm setup dlfm.cfg wurde entfernt. Alle in der Dokumentation enthaltenen Verweise darauf sollten ignoriert werden.

Fehler beim Ausführen von des Data Links/DFS-Scripts "dmapp_prestart" unter AIX

Wenn der Befehl

```
/usr/sbin/cfgdmepi -a "/usr/lib/drivers/dmlfs.ext"
```

mit einem Rückkehrcode von 1 beim Ausführen des Data Links/DFS-Scripts dmapp_prestart fehlschlägt, installieren Sie PTF-Set 1 für DFS 3.1, um den Befehl "cfgdmepi" zu korrigieren.

Integration von Tivoli Space Manager in Data Links

DB2 Data Links Manager kann jetzt die Vorteile der Funktionalität von Tivoli Space Manager nutzen. Das Client-Programm Hierarchical Storage Manager (HSM) von Tivoli Space Manager migriert automatisch auswählbare Dateien in den Speicher, um bestimmte Stufen von freiem Speicher in den lokalen Dateisystemen zu verwalten. Es ruft migrierte Dateien automatisch wieder ab, wenn sie benötigt werden und ermöglicht dem Benutzer das Migrieren und erneute Abrufen bestimmter Dateien.

Diese neue Funktion bietet Vorteile für Kunden, die Dateisysteme mit großen Dateien haben, die in regelmäßigen Abständen in tertiären Speicher versetzt werden müssen, in dem der Speicherbereich des Dateisystems in regelmäßigen Abschnitten verwaltet werden muss. Für viele Kunden bietet Tivoli Space Manager zurzeit die Möglichkeit, ihren tertiären Speicher zu verwalten. Die neue Unterstützung von DB2 Data Links Manager in Tivoli Space Manager bietet größere Flexibilität beim Verwalten des Speicherbereichs für DATALINK-Dateien. Mit Tivoli Space Manager muss der Speicher für alle Dateien, die möglicherweise im DB2 Data Links Manager-Dateisystem gespeichert werden, nicht mehr im Voraus zugeordnet werden. Statt dessen kann der zugeordnete Speicherbereich für das Dateisystem über einen Zeitraum hinweg angepasst werden, ohne dass die Gefahr besteht, dass das Dateisystem während normaler Belegung unbeabsichtigt gefüllt wird.

Hinzufügen der Unterstützung von Data Links und HSM zu einem Dateisystem

Wenn Sie ein Dateisystem mit Hierarchical Storage Management (HSM) registrieren, registrieren Sie es *zuerst* mit HSM und danach mit DataLinks File Manager.

1. Registrieren Sie das Dateisystem mit HSM mit dem Befehl **"dsmmigfs add /fs"**.
2. Registrieren Sie das Dateisystem mit DLM mit dem Befehl **"dlfmfsmd /fs"**.

Die Unterstützung von Data Links für ein Dateisystem spiegelt sich für ein HSM-Dateisystem in den folgenden Einträgen der Zeilengruppe in `/etc/filesystems` wieder:

```
vfs = dlfs
mount = false
options = rw,Basefs=fsm
nodename = -
```

Hinzufügen der Unterstützung von Data Links zu einem vorhandenen HSM-Dateisystem

Registrieren Sie das Dateisystem mit DLM mit dem Befehl **"dlfmfsmd /fs"**.

Hinzufügen der Unterstützung von HSM zu einem vorhandenen Data Links-Dateisystem

1. Registrieren Sie das Dateisystem mit HSM mit dem Befehl **"dsmmigfs add /fs"**.
2. Registrieren Sie das Dateisystem mit DLM mit dem Befehl **"dlfmfsmd /fs"**.

Entfernen der Unterstützung von Data Links aus einem Data Links-HSM-Dateisystem

Entfernen Sie die Unterstützung von Data Links mit dem Befehl `"dlmfsmd -j /fs"`.

Entfernen der Unterstützung von HSM aus einem Data Links-HSM-Dateisystem

1. Entfernen Sie die Unterstützung von HSM mit dem Befehl `"dsmmigfs remove /fs"`.
2. Entfernen Sie die Unterstützung von Data Links mit dem Befehl `"dlmfsmd -j /fs"`.
3. Registrieren Sie das Dateisystem mit DLM mit dem Befehl `"dlmfsmd /fs"`.

Entfernen der Unterstützung von Data Links und HSM aus einem Data Links-HSM-Dateisystem

1. Entfernen Sie die Unterstützung von HSM mit dem Befehl `"dsmmigfs remove /fs"`.
2. Entfernen Sie die Unterstützung von Data Links mit dem Befehl `"dlmfsmd -j /fs"`.

Einschränkungen und Begrenzungen

Diese Funktion wird zurzeit nur unter AIX unterstützt.

Die selektive Migration (dsmmigrate) und das Abrufen einer verbundenen FC-Datei (Datenbank mit Lesezugriff) sollte nur von einem Root-Benutzer ausgeführt werden.

Die selektive Migration kann nur durch den Dateieigner ausgeführt werden. Im Fall von Datenbankdateien mit Lesezugriff ist dies der Data-Link Manager Administrator (dlfm). Ein Token von der Hostdatenbankseite ist erforderlich, um auf eine solche Datei zugreifen zu können. Der einzige Benutzer, der kein Token benötigt, ist der Benutzer "Root". Für den Benutzer "Root" ist es einfacher, die selektive Migration und das Abrufen von Datenbankdateien mit Lesezugriff auszuführen. Der Benutzer "dlfm" kann nur beim ersten Mal eine FC-Datei migrieren, indem er ein gültiges Token verwendet. Beim zweiten Versuch (nach einem erneuten Aufruf) schlägt die Migration fehl, und die Nachricht ANS1028S wird ausgegeben, mit dem Hinweis auf einen internen Programmfehler und der Aufforderung, sich an den IBM Ansprechpartner zu wenden. Die Ausführung des Befehls **dsmmigrate** für eine FC-Datei durch einen Benutzer, der keine Root-Berechtigungen hat, schlägt fehl. Diese Einschränkung ist geringfügig, weil es normalerweise Administratoren sind, die auf die Dateien auf dem Dateiserver zugreifen.

Die Systemaufrufe stat und statfs zeigen den vfs-Typ mit fsm anstelle von dlfs an, obwohl dlfs über fsm angehängt ist.

Das oben aufgeführte Verhalten ist die normale Funktionalität der **dsmrecalld**-Dämonen, die den Befehl „statfs“ für das Dateisystem ausführen, um zu überprüfen, ob sein vfs-Typ "fsm" ist oder nicht.

Der Befehl "dsmls" zeigt keine Ausgabe an, wenn eine Datei mit der minimalen I-Node-Nummer FC-verbunden ist (Datenbank mit Lesezugriff)

Der Befehl **dsmls** funktioniert ähnlich wie der Befehl **ls** und listet die Dateien auf, die von TSM verwaltet werden. Es ist keine Benutzeraktion erforderlich.

Kapitel 4. Installieren und Konfigurieren von DB2 Data Links Manager für AIX

Allgemeine Installationsvoraussetzungen

Migration von DB2 File Manager Version 5.2 auf DB2 Data Links Manager Version 7

Die Informationen in Schritt 3 sind nicht korrekt. Schritt 3 muss folgendermaßen lauten:

"3. Führen Sie als DLFM-Administrator den Befehl
`/usr/lpp/db2_07_01/adm/db2dlmmg` aus.

Installation und Konfiguration: Ergänzung

Kapitel 5. Installieren von DB2-Clients auf UNIX-Betriebssystemen

Kernel-Konfigurationsparameter für HP-UX

Die Empfehlung zum Einstellen von Kernel-Parametern für HP-UX besagt fälschlicherweise, dass `msgmbn` und `msgmax` auf 65535 oder höher gesetzt werden sollten. Beide Parameter müssen genau auf 65535 gesetzt werden.

Kapitel 12. Ausführen Ihrer eigenen Anwendungen

Binden von Datenbankdienstprogrammen mit Run-Time Client

Datenbankdienstprogramme (`import`, `export`, `reorg`, Befehlszeilenprozessor) und DB2 CLI-Bindedateien können nicht mit DB2 Run-Time Client an die einzelnen Datenbanken gebunden werden, um mit diesen Datenbanken eingesetzt zu werden. Verwenden Sie stattdessen DB2 Administration Client oder DB2 Application Development Client.

Diese Datenbankdienstprogramme und DB2 CLI-Bindedateien müssen an die einzelnen Datenbanken gebunden werden, bevor sie mit diesen verwendet werden können. Wenn Sie in einer Netzwerkkumgebung mehrere Clients verwenden, die unter verschiedenen Betriebssystemen ausgeführt werden oder unterschiedliche Versionen oder Service-Stufen von DB2 aufweisen, müssen Sie die Dienstprogramme für jede Kombination aus Betriebssystem und DB2-Version einmal binden.

Client-Zugriff auf DB2 mit ODBC unter UNIX

In Kapitel 12, "Ausführen Ihrer eigenen Anwendungen", wird angegeben, dass Sie die Datei `odbcinst.ini` aktualisieren müssen, wenn Sie einen ODBC-Treiber-Manager mit Ihrer ODBC-Client-Anwendung oder mit ODBC SDK installieren wollen. Dies ist nur zum Teil richtig. Sie müssen `odbcinst.ini` nicht aktualisieren, wenn Sie einen Merant-ODBC-Treiber-Manager installieren.

Kapitel 24. Einrichten eines Systems zusammengesetzter Datenbanken für den Zugriff auf Einstellen eines Systems auf mehrere Datenquellen

Systeme zusammengesetzter Datenbanken

Ein DB2-System zusammengesetzter Datenbanken ist ein besonderer Typ eines Verwaltungssystems für verteilten Datenbanken. Ein System zusammengesetzter Datenbanken erlaubt Ihnen die Abfrage und das Abrufen von Daten, die sich in anderen Datenbankverwaltungssystemen wie z. B. Oracle, Sybase und Microsoft SQL Server befinden. SQL-Anweisungen können sich auf mehrere Datenbankverwaltungssysteme oder einzelne Datenbanken in einer einzigen Anweisung beziehen. Sie können z. B. Daten verknüpfen, die sich in einer DB2 Universal Database-Tabelle, einer Oracle-Tabelle und einer Sybase-Sicht befinden. Die unterstützten Datenbankverwaltungssysteme schließen Oracle, Sybase, Microsoft SQL

Server (für Windows NT) und Mitglieder der DB2 Universal Database-Produktfamilie ein (wie z. B. DB2 für OS/390, DB2 für AS/4000 und DB2 für Windows).

Ein DB2-System zusammengesetzter Datenbanken besteht aus einem Server mit einem DB2-Exemplar (eine Datenbank, die als zusammengesetzte Datenbank verwendet wird) und mindestens einer Datenquelle. Die zusammengesetzte Datenbank enthält Katalogeinträge, die die Datenquellen und deren Kenndaten angeben. Eine Datenquelle besteht aus einem Datenbankverwaltungssystem und Daten. DB2 Universal Database hat Protokolle, die sogenannten Oberflächen, die Sie für den Zugriff auf diese verwenden können. Oberflächen sind die Mittel, mit denen Server mit zusammengesetzten Datenbanken mit Datenquellen kommunizieren und Daten von Datenquellen abrufen. Kurznamen werden verwendet, um auf Tabellen und Sichten zu verweisen, die sich in den Datenquellen befinden. Anwendungen verbinden sich mit der zusammengesetzten Datenbank wie mit jeder anderen DB2-Datenbank. Die Oberfläche, die Sie verwenden, hängt von der Plattform ab, auf der DB2 Universal Database ausgeführt wird.

Nachdem ein System zusammengesetzter Datenbanken installiert ist, kann der Zugriff auf die Informationen in Datenquellen erfolgen, als ob sie in einer großen Datenbank stehen würden. Benutzer und Anwendungen senden Abfragen an eine zusammengesetzte Datenbank, die die Daten aus den Datenquellen abrufen.

Für ein DB2-System zusammengesetzter Datenbanken gelten einige Einschränkungen. In DB2 Version 7 sind verteilte Anforderungen auf Operationen mit Leszugriff beschränkt. Außerdem können Sie keine Dienstprogrammoperationen (LOAD, REORG, REORGCHK, IMPORT, RUNSTATS usw.) für Kurznamen ausführen. Sie können allerdings eine Durchgriffsfunktion verwenden, um DDL- und DML-Anweisungen direkt an Datenbankverwaltungssysteme zu übermitteln, indem Sie die SQL-Programmversion verwenden, die der Datenquelle zugeordnet ist.

Einschränkung

Die neuen Oberflächen in Version 7.2 (wie z. B. Oracle unter Linux und Solaris, Sybase unter AIX und Solaris und Microsoft SQL Server unter NT und AIX) sind in FixPak 3 nicht verfügbar. Sie müssen DB2 Relational Connect Version 7.2 erwerben.

Installieren von DB2 Relational Connect

Dieser Abschnitt gibt Anweisungen, wie Sie DB2 Relational Connect auf dem Server installieren, den Sie als Server für Ihr System zusammengesetzter Datenbanken verwenden wollen.

Installieren von DB2 Relational Connect auf Windows NT-Servern
Überprüfen Sie Folgendes, bevor Sie DB2 Relational Connect auf Ihrem Server mit zusammengesetzten Datenbanken unter Windows NT installieren:

- Stellen Sie sicher, dass Sie entweder DB2 Universal Database Enterprise Edition oder DB2 Universal Database Enterprise — Extended Edition auf dem Server mit zusammengesetzten Datenbanken installiert haben. Wenn Sie Datenbanken der DB2-Systemfamilie in Ihre verteilten Anforderungen integrieren wollen, müssen Sie die Datenquellenoption **Verteilte Verknüpfung für DB2-Datenquellen** ausgewählt haben, als Sie DB2 Universal Database installiert haben. Überprüfen Sie, ob der Parameter FEDERATED auf YES gesetzt ist, um zu bestätigen, dass diese Option implementiert wurde. Sie können diese Einstellung überprüfen, indem Sie den Befehl **GET DATABASE MANAGER CONFIGURATION** absetzen, der alle Parameter und ihre aktuelle Einstellung anzeigt.

- Stellen Sie sicher, dass Sie die Client-Software für die Datenquelle (wie z. B. Sybase Open Client) auf dem Server mit zusammengeschlossenen Datenbanken installiert haben.
1. Melden Sie sich an dem System mit dem Benutzerprofil an, das Sie erstellt haben, um die Installation auszuführen.
 2. Beenden Sie alle aktiven Programme, damit das Installationsprogramm Dateien, falls erforderlich, aktualisieren kann.
 3. Rufen Sie das Installationsprogramm auf. Sie können das Installationsprogramm entweder automatisch oder manuell aufrufen. Wenn der automatische Start des Installationsprogramms fehlschlägt oder Sie die Installation in einer anderen Sprache ausführen wollen, rufen Sie das Installationsprogramm manuell auf.
 - Legen Sie die CD von DB2 Relational Connect in das Laufwerk ein, um das Installationsprogramm automatisch aufzurufen. Die Autorun-Funktion startet das Installationsprogramm automatisch. Die Systemsprache wird ermittelt, und das Installationsprogramm für diese Sprache wird gestartet.
 - Gehen Sie wie folgt vor, um das Installationsprogramm manuell aufzurufen:
 - a. Klicken Sie **Start** an, und wählen Sie die Option **Ausführen** aus.
 - b. Geben Sie den folgenden Befehl in das Feld **Öffnen** ein:


```
x:\setup /i sprache
```

Dabei gilt Folgendes:

x: Steht für Ihr CD-ROM-Laufwerk.

sprache Steht für den Landescode für Ihre Sprache (z. B. DE für Deutsch).
 - c. Klicken Sie **OK** an.

Der Installationsassistent wird geöffnet.

4. Klicken Sie **Installieren** an, um den Installationsprozess zu starten.
5. Folgen Sie den Eingabeaufforderungen im Installationsprogramm.

Wenn das Programm beendet ist, ist DB2 Relational Connect in Ihrem Installationsverzeichnis mit Ihren anderen DB2-Produkten installiert.

Installieren von DB2 Relational Connect auf Servern mit AIX, Linux und der Solaris-Betriebsumgebung

Überprüfen Sie Folgendes, bevor Sie DB2 Relational Connect auf Ihrem Server mit zusammengeschlossenen Datenbanken unter AIX, Linux und der Solaris-Betriebsumgebung installieren:

- Stellen Sie sicher, dass Sie entweder DB2 Universal Database Enterprise Edition oder DB2 Universal Database Enterprise — Extended Edition auf dem Server mit zusammengeschlossenen Datenbanken installiert haben. Wenn Sie Datenbanken der DB2-Systemfamilie in Ihre verteilten Anforderungen integrieren wollen, müssen Sie die Datenquelloption **Verteilte Verknüpfung für DB2** ausgewählt haben, als Sie DB2 Universal Database installiert haben. Überprüfen Sie, ob der Parameter FEDERATED auf YES gesetzt ist, um zu bestätigen, dass diese Option implementiert wurde. Sie können diese Einstellung überprüfen, indem Sie den Befehl **GET DATABASE MANAGER CONFIGURATION** absetzen, der alle Parameter und ihre aktuelle Einstellung anzeigt.
- Stellen Sie sicher, dass Sie die Client-Software für die Datenquelle (wie z. B. Sybase Open Client) auf dem Server mit zusammengeschlossenen Datenbanken installiert haben.

Verwenden Sie das Dienstprogramm `db2setup`, um DB2 Relational Connect auf Ihren Servern mit AIX, Linux und der Solaris-Betriebsumgebung zu installieren:

1. Melden Sie sich als Benutzer mit Root-Berechtigung an.
2. Legen Sie Ihre DB2-Produkt-CD-ROM ein, und hängen Sie sie an. Informationen dazu, wie Sie eine CD-ROM anhängen, finden Sie in dem Handbuch *Quick Beginnings for AIX*.
3. Wechseln Sie in das Verzeichnis, in dem die CD-ROM angehängt ist, indem Sie den Befehl `cd /cd-rom` eingeben. Dabei gilt Folgendes: `cd-rom` ist der Mount-Punkt Ihrer Produkt-CD-ROM.
4. Geben Sie den Befehl `./db2setup` ein. Nach einigen Momenten wird das Fenster des DB2-Installationsdienstprogramms geöffnet.
5. Wählen Sie **Installieren** aus. Das Fenster „DB2 Version 7 installieren“ wird geöffnet.
6. Navigieren Sie zu dem Produkt DB2 Relational Connect für Ihren Client, z. B. Relational Connect for Sybase, und drücken Sie die Leertaste, um es auszuwählen. Ein Stern wird neben der Option angezeigt, wenn Sie ausgewählt ist.
7. Wählen Sie **OK** aus. Das Fenster „DB2-Services erstellen“ wird geöffnet.
8. Sie können auswählen, dass Sie ein DB2-Exemplar erstellen wollen. Wählen Sie **OK** aus. Das Fenster mit dem Ergebnisbericht für die Installation wird geöffnet. Zwei Elemente werden automatisch installiert: die verteilte Verknüpfung für Oracle und die Programmkennung für DB2 Relational Connect. Die Programmkennung ist erforderlich, damit Sie eine Verbindung zu Sybase-Datenquellen herstellen können.
9. Wählen Sie **Weiter** aus. Ein Fenster wird angezeigt, um Ihnen mitzuteilen, dass dies die letzte Möglichkeit ist, die Installation von Relational Connect zu stoppen. Wählen Sie **OK** aus, um mit der Installation fortzufahren. Das Durchführen der Installation kann einige Minuten dauern.
10. Wenn ein Hinweis angezeigt wird, dass die Installation erfolgreich beendet ist, wählen Sie **OK** aus. Das Fenster „Ergebnisbericht“ wird geöffnet, um den Erfolg oder Fehlschlag der einzelnen installierten Optionen anzuzeigen. Wählen Sie noch einmal **OK** aus.

Wenn die Installation beendet ist, ist DB2 Relational Connect in dem Verzeichnis mit Ihren anderen DB2-Produkten installiert. Unter AIX ist dies das Verzeichnis `/usr/lpp/db2_07_01`. Unter Solaris ist dies das Verzeichnis `/opt/IBMdb2/V7.1`. Unter Linux ist dies das Verzeichnis `/usr/IBMdb2/V7.1`.

Kapitel 26. Zugriff auf Oracle-Datenquellen

Zusätzlich zur Unterstützung von Oberflächen unter AIX und Windows NT unterstützt DB2 Universal Database jetzt auch die Oracle-Oberfläche unter Linux und der Solaris-Betriebsumgebung. Diese Unterstützung begrenzt sich auf Oracle Version 8. Sie müssen die CD von DB2 Relational Connect Version 7.2 einlegen und DB2 Relational Connect für Oracle-Datenquellen auswählen, um auf die Oberflächen für diese Plattformen zuzugreifen.

Nachdem Sie DB2 Relational Connect installiert haben, können Sie einem Server mit zusammengeschlossenen Datenbanken wie folgt eine Oracle-Datenquelle hinzufügen:

1. Installieren und konfigurieren Sie die Oracle-Client-Software auf dem DB2-Server mit zusammengeschlossenen Datenbanken.
2. Legen Sie die Umgebungsvariablen für die Datenquellen fest, indem Sie die Datei `db2dj.ini` ändern und den Befehl `db2set` absetzen.

3. Führen Sie bei DB2-Servern mit zusammengeschlossenen Datenbanken, die auf UNIX-Plattformen aktiv sind, das Script `djxlink` aus, um die Bibliothek Oracle SQL*Net oder Net8 über eine Programmverbindung mit Ihrem DB2-Server mit zusammengeschlossenen Datenbanken zu verbinden.
4. Stellen Sie sicher, dass die SQL*Net- oder Net8-Datei `tnsnames.ora` aktualisiert wird.
5. Starten Sie das DB2-Exemplar erneut.
6. Erstellen Sie die Oberfläche.
7. Optional: Legen Sie die Umgebungsvariable `DB2_DJ_COMM` fest.
8. Erstellen Sie einen Server.
9. Erstellen Sie eine Benutzerzuordnung.
10. Erstellen Sie Kurznamen für Tabellen und Sichten.

Genaue Anweisungen für diese Schritte, einschließlich der Einstellung der Umgebungsvariablen, finden Sie in Kapitel 26, Einrichten eines Systems zusammengeschlossener Datenbanken für den Zugriff auf Oracle-Datenquellen, im Handbuch *DB2 Installation und Konfiguration - Ergänzung*.

Fehler in der Dokumentation

Der Abschnitt „Hinzufügen von Oracle-Datenquellen zu einem System zusammengeschlossener Datenbanken“ enthält folgende Fehler:

- In der Prozedur fehlt ein Schritt. Die korrekte Schrittabfolge ist:

1. Installieren und konfigurieren Sie die Oracle-Client-Software auf dem DB2-Server mit zusammengeschlossenen Datenbanken mit Hilfe der von Oracle zur Verfügung gestellten Dokumentation.

2.

Definieren Sie Umgebungsvariablen für Datenquellen durch eine Änderung der Datei `db2dj.ini` und durch Absetzen des Befehls **db2set**. Der Befehl **db2set** aktualisiert die DB2-Profilregistrierdatenbank mit Ihren Einstellungen.

Ausführliche Anweisungen zum Einstellen der Umgebungsvariablen finden Sie in Kapitel 26, Einrichten eines Systems zusammengeschlossener Datenbanken für den Zugriff auf Oracle-Datenquellen, des Handbuchs *DB2 Installation und Konfiguration: Ergänzung*.

3. Führen Sie bei DB2-Servern mit zusammengeschlossenen Datenbanken, die auf UNIX-Plattformen aktiv sind, das Script `djxlink` aus, um die Bibliothek Oracle SQL*Net oder Net8 über eine Programmverbindung mit Ihrem DB2-Server mit zusammengeschlossenen Datenbanken zu verbinden. Abhängig von Ihrer Plattform befindet sich das Script `djxlink` in einem der folgenden Verzeichnisse:

```
/usr/lpp/db2_07_01/bin unter AIX
/opt/IBMdb2/V7.1/bin unter Solaris
/usr/IBMdb2/V7.1/bin unter Linux
```

Führen Sie das Script `djxlink` erst nach der Installation der Oracle-Client-Software auf dem DB2-Server mit zusammengeschlossenen Datenbanken aus.

- Laut Dokumentation soll folgende Einstellung vorgenommen werden:

```
DB2_DJ_INI = sql1lib/cfg/db2dj.ini
```

Dies ist falsch. Die Einstellung muss wie folgt lauten:

```
DB2_DJ_INI = $INSTHOME/sql1lib/cfg/db2dj.ini
```

Zugriff auf Sybase-Datenquellen (neues Kapitel)

Bevor Sie einem Server mit zusammengeschlossenen Datenbanken Sybase-Datenquellen hinzufügen, müssen Sie die Sybase Open Client-Software auf dem DB2-Server mit zusammengeschlossenen Datenbanken installieren und konfigurieren. In der Installationsbeschreibung in der Dokumentation, die mit der Sybase-Datenbanksoftware geliefert wird, finden Sie genaue Einzelangaben zur Installation der Open Client-Software. Stellen Sie sicher, dass die im Katalog gespeicherten Prozeduren und die Bibliotheken von Sybase Open Client installiert werden.

Sie müssen die folgenden Operationen ausführen, um Ihren Server mit zusammengeschlossenen Datenbanken zu definieren, damit er auf Daten zugreifen kann, die in Sybase-Datenquellen gespeichert sind:

1. Installieren Sie DB2 Relational Connect Version 7.2. Siehe „Installieren von DB2 Relational Connect“ auf Seite 40.
2. Fügen Sie Ihrem Server mit zusammengeschlossenen Datenbanken Sybase-Datenquellen hinzu.
3. Geben Sie die Sybase-Codepages an.

Dieses Kapitel geht auf die Schritte 2 und 3 ein.

Die Anweisungen in diesem Kapitel gelten für Windows NT, AIX und die Solaris-Betriebsumgebung. Die plattformspezifischen Unterschiede werden erwähnt, wo Sie zutreffen.

Hinzufügen von Sybase-Datenquellen zu einem Server mit zusammengeschlossenen Datenbanken

Sie müssen die folgenden Operationen ausführen, um einem Server mit zusammengeschlossenen Datenbanken eine Sybase-Datenquelle hinzuzufügen:

1. Legen Sie die Umgebungsvariablen fest, und aktualisieren Sie die Profilregistrierdatenbank.
2. Verknüpfen Sie DB2 mit der Sybase-Client-Software (nur AIX und Solaris).
3. Starten Sie das DB2-Exemplar erneut.
4. Erstellen und definieren Sie eine Schnittstellendatei.
5. Erstellen Sie die Oberfläche.
6. Optional: Legen Sie die Umgebungsvariable DB2_DJ_COMM fest.
7. Erstellen Sie einen Server.
8. Optional: Legen Sie die Serveroption CONNECTSTRING fest.
9. Erstellen Sie eine Benutzerzuordnung.
10. Erstellen Sie Kurznamen für Tabellen und Sichten.

Diese Schritte werden in diesem Abschnitt genau erklärt.

Schritt 1: Festlegen der Umgebungsvariablen und Aktualisieren der Profilregistrierdatenbank

Definieren Sie Umgebungsvariablen für Datenquellen durch eine Änderung der Datei `db2dj.ini` und durch Absetzen des Befehls `db2set`. Die Datei `db2dj.ini` enthält Konfigurationsdaten zu der Sybase-Client-Software, die auf Ihrem Server mit zusammengeschlossenen Datenbanken installiert ist. Der Befehl `db2set` aktualisiert die DB2-Profilregistrierdatenbank mit Ihren Einstellungen.

In einem partitionierten Datenbanksystem können Sie eine einzige Datei `db2dj.ini` für alle Knoten in einem bestimmten Exemplar verwenden, oder Sie können eine eindeutig Datei `db2dj.ini` für einen odere mehrere Knoten in einem bestimmten Exemplar verwenden. Ein nicht-partitioniertes Datenbanksystem kann nur eine Datei `db2dj.ini` für jedes Exemplar haben.

Gehen Sie wie folgt vor, um die Umgebungsvariablen festzulegen:

1. Editieren Sie die Datei `db2dj.ini`, die sich im Verzeichnis `sql1lib/cfg` befindet, und legen Sie die folgende Umgebungsvariable fest:

```
SYBASE= "<sybase-ausgangsverzeichnis>"
```

Dabei ist `<sybase-ausgangsverzeichnis>` das Verzeichnis, in dem der Sybase-Client installiert ist.

2. Aktualisieren Sie die Datei `.profile` für das DB2-Exemplar mit der Sybase-Umgebungsvariable. Sie können dies tun, indem Sie den folgenden Befehl absetzen:

```
export PATH=„$SYBASE/bin:$PATH“
export SYBASE="<sybase-ausgangsverzeichnis>"
```

Dabei gilt ist `<sybase-ausgangsverzeichnis>` das Verzeichnis, in dem der Sybase-Client installiert ist.

3. Führen Sie die Datei `.profile` auf dem DB2-Exemplar aus, indem Sie Folgendes eingeben:

```
. .profile
```

4. Setzen Sie den Befehl `db2set` ab, um die DB2-Profilregistrierdatenbank mit Ihren Änderungen zu aktualisieren. Die Syntax des Befehls `db2set` hängt von Ihrer Datenbanksystemstruktur ab. Dieser Schritt ist nur erforderlich, wenn Sie die Datei `db2dj.ini` in einer der folgenden Datenbanksystemstrukturen verwenden:

Wenn Sie die Datei `db2dj.ini` in einem nicht-partitionierten Datenbanksystem verwenden oder wenn Sie wollen, dass die Datei `db2dj.ini` nur auf den aktuellen Knoten angewendet wird, setzen Sie den folgenden Befehl ab:

```
db2set DB2_DJ_INI = sql1lib/cfg/db2dj.ini
```

Wenn Sie die Datei `db2dj.ini` in einem partitionierten Datenbanksystem verwenden und Sie wollen, dass die Werte in der Datei `db2dj.ini` für alle Knoten innerhalb dieses Exemplars gelten sollen, setzen Sie den folgenden Befehl ab:

```
db2set -g DB2_DJ_INI = sql1lib/cfg/db2dj.ini
```

Wenn Sie die Datei `db2dj.ini` in einem partitionierten Datenbanksystem verwenden und Sie wollen, dass die Werte in der Datei `db2dj.ini` für einen bestimmten Knoten gelten sollen, setzen Sie den folgenden Befehl ab:

```
db2set -i EXEMPX 3 DB2_DJ_INI = sql1lib/cfg/knoten3.ini
```

Dabei gilt Folgendes:

EXEMPX

Der Name des Exemplars.

3 Die Knotennummer, die in der Datei `db2nodes.cfg` aufgelistet wird.

knoten3.ini

Die geänderte und umbenannte Version der Datei `db2dj.ini`.

Schritt 2: Verbinden von DB2 mit der Sybase-Client-Software (nur AIX und Solaris)

Es muss eine Programmverbindung zwischen dem DB2-Server mit zusammengeschlossenen Datenbanken und den Client-Bibliotheken hergestellt werden, um den Zugriff auf Sybase-Datenquellen zu aktivieren. Durch das Herstellen einer Programmverbindung wird eine Oberfläche für alle Datenquellen erstellt, über die der Server mit zusammengeschlossenen Datenbanken kommunizieren wird. Wenn Sie das Script `djxlink` ausführen, erstellen Sie die Oberflächenbibliothek. Geben Sie Folgendes ein, um das Script `djxlink` abzusetzen:

```
djxlink
```

Schritt 3: DB2-Exemplar erneut starten

Starten Sie das DB2-Exemplar erneut, um sicherzustellen, dass die Umgebungsvariablen in dem Programm festgelegt sind. Wenn Sie das Exemplar erneut starten, aktualisieren Sie das DB2-Exemplar so, dass es die vorgenommenen Änderungen übernimmt.

Setzen Sie die folgenden Befehle ab, um das DB2-Exemplar erneut zu starten:

In DB2 für Windows NT-Server:

```
NET STOP exemplarname  
NET START exemplarname
```

Auf DB2 für AIX- und Solaris-Servern:

```
db2stop  
db2start
```

Schritt 4: Erstellen und Definieren einer Schnittstellendatei

Sie müssen die Datei erstellen und den Zugriff auf sie erlauben, um eine Schnittstellendatei zu definieren.

1. Verwenden Sie das Sybase-Dienstprogramm, um eine Schnittstellendatei zu erstellen, die die Daten für alle Sybase Open Server enthält, auf die Sie zugreifen wollen. Weitere Informationen zur Verwendung dieses Dienstprogramms finden Sie in der Sybase-Installationsdokumentation.

Windows NT nennt diese Datei normalerweise `sql.ini`. Benennen Sie die Datei, die gerade erstellt haben, von `sql.ini` in `interfaces.um`, um die Datei auf allen Plattformen gleich zu nennen. Wenn Sie die Datei `sql.ini` nicht in `interfaces` umbenennen wollen, müssen Sie den Parameter `IFILE` oder die Option `CONNECTSTRING` verwenden, der/die in Schritt 8 erklärt wird.

Auf AIX- und Solaris-Systemen heißt diese Datei
<exemplarausgangsverzeichnis>/sqllib/interfaces.

2. Erlauben Sie DB2 den Zugriff auf die Schnittstellendatei.

Auf DB2 für Windows NT-Servern:

Stellen Sie die Datei in das Verzeichnis `%DB2PATH%` des DB2-Exemplars.

Auf DB2 für AIX- und Solaris-Servern:

Stellen Sie die Datei in das Verzeichnis `$HOME/sqllib` des DB2-Exemplars. Verwenden Sie den Befehl `ln`, um eine Verknüpfung zu der Datei in dem Verzeichnis `$HOME/sqllib` des DB2-Exemplars herzustellen. Beispiel:

```
ln -s -f /home/sybase/interfaces /home/db2djinst1/sqllib
```

Schritt 5: Erstellen der Oberfläche

Verwenden Sie die Anweisung `CREATE WRAPPER`, um die Oberfläche anzugeben, die verwendet werden soll, um auf Sybase-Datenquellen zuzugreifen. Oberflächen sind die Mittel, mit denen Server mit zusammengeschlossenen Datenbanken mit

Datenquellen kommunizieren und Daten von Datenquellen abrufen. DB2 enthält zwei Oberflächen für Sybase: CTLIB und DBLIB. Das folgende Beispiel zeigt eine CREATE WRAPPER-Anweisung:

```
CREATE WRAPPER CTLIB
```

Dabei gilt Folgendes: CTLIB ist der Standardoberflächenname, der mit der Sybase Open Client-Software verwendet wird. Die Oberfläche CTLIB kann auf Windows NT-, AIX- und Solaris-Servern verwendet werden.

Sie können den Standardoberflächennamen mit einem Namen ihrer Wahl ersetzen. Wenn Sie dies machen, müssen Sie allerdings auch den Parameter LIBRARY und den Namen der Oberflächenbibliothek für Ihren Server mit zusammengesetzten Datenbanken in der Anweisung CREATE WRAPPER angeben. Weitere Informationen zu Oberflächenbibliotheksnamen finden Sie bei der Anweisung CREATE WRAPPER im Handbuch *SQL Reference*.

Schritt 6: Optional: Festlegen der Umgebungsvariablen DB2_DJ_COMM

Setzen Sie die Umgebungsvariable DB2_DJ_COMM, um die Leistung für den Zugriff auf die Sybase-Datenquelle zu erhöhen. Diese Variable stellt fest, ob eine Oberfläche geladen ist, wenn der Server mit zusammengesetzten Datenbanken initialisiert wird. Legen Sie die Umgebungsvariable DB2_DJ_COMM fest, damit die Oberflächenbibliothek integriert wird, die zu der Oberfläche gehört, die Sie in dem vorherigen Schritt angegeben haben. Beispiel:

Auf DB2 für Windows NT-Servern:

```
db2set DB2_DJ_COMM='ctlib.dll'
```

Auf DB2 für AIX-Servern:

```
db2set DB2_DJ_COMM='libctlib.a'
```

Auf DB2 für Solaris-Servern:

```
db2set DB2_DJ_COMM='libctlib.so'
```

Stellen Sie sicher, dass sich auf keiner Seite des Gleichheitszeichens (=) Leerzeichen befinden.

Weitere Informationen zu den Oberflächenbibliotheksnamen finden Sie im Handbuch *SQL Reference*. Weitere Informationen zu der Umgebungsvariable DB2_DJ_COMM finden Sie im Handbuch *Systemverwaltung*.

Schritt 7: Erstellen des Servers

Verwenden Sie die Anweisung CREATE SERVER, um alle Sybase-Server zu definieren, auf die Sie zugreifen wollen. Beispiel:

```
CREATE SERVER SYBSERVER TYPE SYBASE VERSION 12.0 WRAPPER CTLIB  
OPTIONS (NODE 'syb-knoten', DBNAME 'syb-db')
```

Dabei gilt Folgendes:

SYBSERVER

Der Name, den Sie einem Sybase-Server zuordnen. Dieser Name muss eindeutig sein.

SYBASE

Der Datenquellentyp, für den Sie den Zugriff konfigurieren. Sybase ist die einzige Datenquelle, die unterstützt wird.

12.0 Die Sybase-Version, auf die Sie zugreifen. Die unterstützten Versionen sind 10.0, 11.0, 11.1, 11.5, 11.9 und 12.0.

CTLIB Der Oberflächenname, den Sie in der Anweisung CREATE WRAPPER angegeben haben.

'syb-knoten'

Der Name des Knotens, auf dem sich SYBSERVER befindet. Entnehmen Sie den Knotenwert der Schnittstellendatei. Beachten Sie bei diesem Wert die Groß-/Kleinschreibung.

Obwohl der Name des Knotens als Option angegeben wird, ist er für Sybase-Datenquellen erforderlich. Informationen zu zusätzlichen Optionen finden Sie im Handbuch *SQL Reference*.

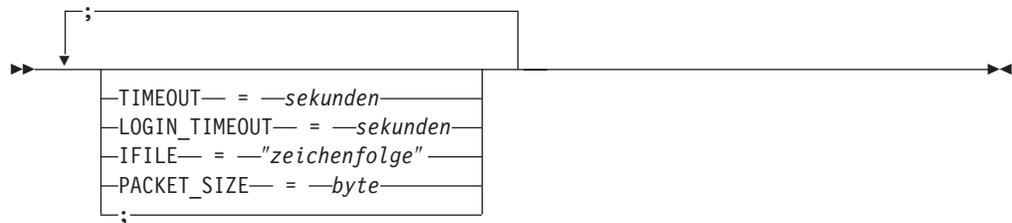
'syb-db'

Der Name der Sybase-Datenbank, auf die Sie zugreifen wollen.

Optional: Schritt 8: Festlegen der Serveroption CONNECTSTRING

Geben Sie die Zeitlimitschwelle, den Pfad und den Namen der Schnittstellendatei sowie die Paketgröße der Schnittstellendatei an. Sybase Open Client verwendet Zeitlimitschwellen, um Abfragen und Antworten zu unterbrechen, die zu lange ausgeführt werden. Sie können diese Schwellen in DB2 mit der Option CONNECTSTRING in der Anweisung CREATE SERVER OPTION DDL festlegen. Verwenden Sie die Option CONNECTSTRING, um Folgendes anzugeben:

- Zeitlimitdauer für SQL-Abfragen
- Zeitlimitdauer für Anmeldeantworten
- Pfad und Name der Schnittstellendatei
- Paketgröße



TIMEOUT

Gibt die Anzahl an Sekunden an, die DB2 Universal Database auf eine Antwort von Sybase Open Client für eine beliebige SQL-Anweisung wartet. Der Wert von *sekunden* ist eine positive ganze Zahl aus dem Integerbereich von DB2 Universal Database. Der Zeitlimitwert, den Sie angeben, hängt von der Oberfläche ab, die Sie verwenden. Windows NT-, AIX- und Solaris-Server können die Oberfläche DBLIB verwenden. Der Standardwert für die Oberfläche DBLIB ist 0. Auf Windows NT-, AIX- und Solaris-Servern veranlasst der Standardwert, dass DB2 Universal Database unendlich lang auf eine Antwort wartet.

LOGIN_TIMEOUT

Gibt die Anzahl an Sekunden an, die DB2 Universal Database auf eine Antwort von Sybase Open Client für eine Anmeldeanfrage wartet. Die Standardwerte sind dieselben wie für TIMEOUT.

IFILE

Gibt den Pfad und Namen der Schnittstellendatei von Sybase Open Client an. Der in *zeichenfolge* angegebene Pfad muss in doppelte Anführungszeichen (") eingeschlossen sein. Auf Windows NT-Servern ist der Standardpfad

%DB2PATH%. Auf AIX- und Solaris-Servern ist der Standardwert `sqllib/interfaces` im Ausgangsverzeichnis Ihres DB2 Universal Database-Exemplars.

PACKET_SIZE

Gibt die Paketgröße der Schnittstellendatei in Byte an. Wenn die Datenquelle die angegebene Paketgröße nicht unterstützt, schlägt die Verbindung fehl. Das Erhöhen der Paketgröße für sehr große Datensätze (z. B. beim Einfügen von Zeilen in große Tabellen) verbessert die Leistung deutlich. Die Bytegröße ist ein numerischer Wert. Weitere Informationen finden Sie in den Sybase-Referenzhandbüchern.

Beispiele:

Verwenden Sie den folgenden Befehl auf Windows NT-Servern, um den Zeitlimitwert auf 60 Sekunden und die Schnittstellendatei auf `C:\etc\interfaces` zu setzen:

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1
SETTING 'TIMEOUT=60;LOGIN_TIMEOUT=5;IFILE="C:\etc\interfaces";'
```

Verwenden Sie den folgenden Befehl auf AIX- und Solaris-Servern, um den Zeitlimitwert auf 60 Sekunden und die Schnittstellendatei auf `/etc/interfaces` zu setzen:

```
CREATE SERVER OPTION connectstring FOR SERVER sybase1
SETTING 'TIMEOUT=60;PACKET_SIZE=4096;IFILE="/etc/interfaces";'
```

Schritt 9: Erstellen einer Benutzerzuordnung

Wenn sich die Benutzer-ID oder das Kennwort auf dem Server mit zusammengesetzten Datenbanken von einer Benutzer-ID oder einem Kennwort in einer Sybase-Datenquelle unterscheiden, verwenden Sie die Anweisung `CREATE USER MAPPING`, um die lokale Benutzer-ID der Benutzer-ID und dem Kennwort zuzuordnen, die/das in der Sybase-Datenquelle definiert ist. Beispiel:

```
CREATE USER MAPPING FOR DB2USER SERVER SYBSERVER
OPTIONS ( REMOTE_AUTHID 'syb-benutzer', REMOTE_PASSWORD
'day11te')
```

Dabei gilt Folgendes:

DB2USER

Die lokale Benutzer-ID, die Sie einer Benutzer-ID zuordnen, die in einer Sybase-Datenquelle definiert ist.

SYBSERVER

Der Name der Sybase-Datenquelle, die Sie in der Anweisung `CREATE SERVER` definiert haben.

'syb-benutzer'

Die Benutzer-ID in der Sybase-Datenquelle, der Sie *DB2USER* zuordnen. Beachten Sie bei diesem Wert die Groß-/Kleinschreibung.

'day11te'

Das Kennwort, das *'syb-benutzer'* zugeordnet ist. Beachten Sie bei diesem Wert die Groß-/Kleinschreibung.

Weitere Informationen zu zusätzlichen Optionen finden Sie im Handbuch *SQL Reference*.

Schritt 10: Erstellen von Kurznamen für Tabellen und Sichten

Ordnen Sie jeder Sicht oder Tabelle, die sich in Ihrer Sybase-Datenquelle befindet, einen Kurznamen zu. Sie verwenden diese Kurznamen, wenn Sie die Sybase-

Datenquelle abfragen. Beachten Sie bei Sybase-Kurznamen die Groß-/Kleinschreibung. Schließen Sie sowohl die Schemen- als auch die Tabellennamen in doppelte Anführungszeichen (") ein. Das folgende Beispiel zeigt eine CREATE NICKNAME-Anweisung:

```
CREATE NICKNAME SYBSALES FOR  
SYBSERVER."verkaufsdaten"."europa"
```

Dabei gilt Folgendes:

SYBSALES

Ein eindeutiger Kurzname für die Sybase-Tabelle oder -Sicht.

SYBSERVER."verkaufsdaten"."europa"

Ein dreiteiliger Bezeichner, der das folgende Format hat:

datenquellename."ferner-schemenname"."ferner-tabellenname"

Wiederholen Sie diesen Schritt für alle Tabellen oder Sichten, für die Sie Kurznamen erstellen wollen. Wenn Sie den Kurznamen erstellen, verwendet DB2 die Verbindung, um den Datenquellenkatalog abzufragen. Diese Abfrage testet Ihre Verbindung zu der Datenquelle. Wenn die Verbindung nicht zustande kommt, empfangen Sie eine Fehlermeldung.

Weitere Informationen zu der Anweisung CREATE NICKNAME finden Sie im Handbuch *SQL Reference*. Weitere Informationen zu Kurznamen im Allgemeinen und zum Prüfen von Datentypzuordnungen finden Sie im Handbuch *DB2 Systemverwaltung*.

Angeben von Sybase-Codepages

Dieser Schritt ist nur notwendig, wenn der DB2-Server mit zusammengeschlossenen Datenbanken und der Sybase-Server unterschiedliche Codepages ausführen. Datenquellen, die denselben codierten Zeichensatz wie DB2 verwenden, erfordern keine Übersetzung. In der folgenden Tabelle finden Sie gleichbedeutende Sybase-Optionen für gemeinsame Codepages mit Unterstützung der Landessprache (National Language Support, NLS). Entweder müssen Sie Ihre Sybase-Datenquellen so konfigurieren, dass sie mit diesen Werten übereinstimmen, oder der Client-Code muss in der Lage sein, die Abweichung zu entdecken und sie als Fehler zu markieren oder die Daten mit seiner eigenen Semantik zuzuordnen. Wenn keine Konvertierungstabelle von der Quellen-Codepage zu der Ziel-Codepage gefunden werden kann, setzt DB2 eine Fehlermeldung ab. Weiter Informationen finden Sie in Ihrer Sybase-Dokumentation.

Tabelle 1. Sybase-Codepage-Optionen

Codepage	Gleichbedeutende Sybase-Option
850	cp850
897	sjis
819	iso_1
912	iso_2
1089	iso_6
813	iso_7
916	iso_8
920	iso_9

Zugriff auf Microsoft SQL Server-Datenquellen mit ODBC (neues Kapitel)

Bevor Sie einem DB2-Server mit zusammengeschlossenen Datenbanken Microsoft SQL Server-Datenquellen hinzufügen, müssen Sie den ODBC-Treiber auf dem Server mit zusammengeschlossenen Datenbanken installieren und konfigurieren. Genaue Einzelangaben zur Installation der ODBC-Treibers finden Sie in den Installationsanleitungen in der Dokumentation, die mit dem ODBC-Treiber ausgeliefert wird.

Sie müssen die folgenden Operationen ausführen, um Ihren Server mit zusammengeschlossenen Datenbanken für den Zugriff auf Daten zu definieren, die in Microsoft SQL Server-Datenquellen gespeichert sind:

1. Installieren und konfigurieren Sie den ODBC-Treiber auf dem Server mit zusammengeschlossenen Datenbanken. Genaue Einzelangaben zur Installation der ODBC-Treibers finden Sie in den Installationsanweisungen in der Dokumentation, die mit dem ODBC-Treiber ausgeliefert wird.

Auf DB2 für Windows NT-Servern:

Konfigurieren Sie einen Systemdatenquellennamen mit dem ODBC-Einheitenmanager.

Auf DB2 für AIX-Servern:

Installieren Sie die Thread-Version der Bibliotheken, die von MERANT angeboten werden, geben Sie das Bibliotheksverzeichnis von MERANT als ersten Eintrag in der Variable LIBPATH an, und definieren Sie die Datei `.odbc.ini`. Erstellen Sie die Datei `.odbc.ini` im Ausgangsverzeichnis.

2. Installieren Sie DB2 Relational Connect Version 7.2. Siehe „Installieren von DB2 Relational Connect“ auf Seite 40.
3. Fügen Sie Ihrem Server mit zusammengeschlossenen Datenbanken Microsoft SQL Server-Datenquellen hinzu.
4. Geben Sie die Microsoft SQL Server-Codepages an.

Dieses Kapitel geht auf die Schritte 3 und 4 ein.

Die Anweisungen in diesem Kapitel gelten für Windows NT- und AIX-Plattformen. Die plattformspezifischen Unterschiede werden erwähnt, wo Sie zutreffen.

Hinzufügen von Microsoft SQL Server-Datenquellen zu einem Server mit zusammengeschlossenen Datenbanken

Nachdem Sie den ODBC-Treiber und DB2 Relational Connect installiert haben, fügen Sie Ihrem Server mit zusammengeschlossenen Datenbanken Microsoft SQL Server-Datenquellen hinzu, indem Sie die folgenden Schritte ausführen:

1. Legen Sie die Umgebungsvariablen fest (nur AIX).
2. Führen Sie das Shell-Script aus (nur AIX).
3. Optional: Legen Sie die Umgebungsvariable `DB2_DJ_COMM` fest.
4. Starten Sie das DB2-Exemplar erneut (nur AIX).
5. Erstellen Sie die Oberfläche.
6. Erstellen Sie den Server.
7. Erstellen Sie eine Benutzerzuordnung.
8. Erstellen Sie Kurznamen für die Tabellen und Sichten.
9. Optional: Rufen Sie die ODBC-Traces ab.

Diese Schritte werden in den folgenden Abschnitten genau erklärt.

Schritt 1: Festlegen der Umgebungsvariablen (nur AIX)

Definieren Sie Umgebungsvariablen für Datenquellen durch eine Änderung der Datei `db2dj.ini` und durch Absetzen des Befehls `db2set`. Die Datei `db2dj.ini` enthält Konfigurationsdaten zur Herstellung einer Verbindung zu Microsoft SQL Server-Datenquellen. Der Befehl `db2set` aktualisiert die DB2-Profilregistrierdatenbank mit Ihren Einstellungen.

In einem partitionierten Datenbanksystem können Sie eine einzige Datei `db2dj.ini` für alle Knoten in einem bestimmten Exemplar verwenden, oder Sie können eine eindeutig Datei `db2dj.ini` für einen odere mehrere Knoten in einem bestimmten Exemplar verwenden. Ein nicht-partitioniertes Datenbanksystem kann nur eine Datei `db2dj.ini` für jedes Exemplar haben.

Gehen Sie wie folgt vor, um die Umgebungsvariablen festzulegen:

1. Editieren Sie die Datei `db2dj.ini`, die sich im Verzeichnis `$HOME/sqllib/cfg` befindet, und legen Sie die folgenden Umgebungsvariable fest:

```
ODBCINI=$HOME/.odbc.ini
DJX_ODBC_LIBRARY_PATH=<pfad-zum-merant-treiber>/lib
LIBPATH=<pfad-zum-merant-treiber>/lib
DB2ENVLIST=LIBPATH
```

Setzen Sie den Befehl `db2set` ab, um die DB2-Profilregistrierdatenbank mit Ihren Änderungen zu aktualisieren. Die Syntax des Befehls `db2set` hängt von Ihrer Datenbanksystemstruktur ab.

- Wenn Sie die Datei `db2dj.ini` in einem nicht-partitionierten Datenbanksystem oder in einem partitionierten Datenbanksystem verwenden und die Datei `db2dj.ini` nur auf den aktuellen Knoten angewendet werden soll, setzen Sie den folgenden Befehl ab:

```
db2set DB2_DJ_INI=<pfad-zur-ini-datei>/db2dj.ini
```

- Wenn Sie die Datei `db2dj.ini` in einem partitionierten Datenbanksystem verwenden und die Werte in der Datei `db2dj.ini` für alle Knoten innerhalb dieses Exemplars gelten sollen, setzen Sie den folgenden Befehl ab:

```
db2set -g
DB2_DJ_INI=<pfad-zur-ini-datei>/db2dj.ini
```

- Wenn Sie die Datei `db2dj.ini` in einem partitionierten Datenbanksystem verwenden und die Werte in der Datei `db2dj.ini` für einen bestimmten Knoten gelten sollen, setzen Sie den folgenden Befehl ab:

```
db2set -i EXEMPX 3 DB2_DJ_INI=$HOME/sqllib/cfg/knoten3.ini
```

Dabei gilt Folgendes:

EXEMPX

Der Name des Exemplars.

3 Die Knotennummer, die in der Datei `db2nodes.cfg` aufgelistet wird.

knoten3.ini

Die geänderte und umbenannte Version der Datei `db2dj.ini`.

Schritt 2: Ausführen des Shell-Scripts (nur AIX)

Das Shell-Script `djxlink.sh` verknüpft die Client-Bibliotheken mit den Oberflächenbibliotheken. Geben Sie Folgendes ein, um das Shell-Script auszuführen:

```
djxlink
```

Schritt 3: Optional: Festlegen der Umgebungsvariablen DB2_DJ_COMM

Wenn Sie meinen, der Zugriff auf die Microsoft SQL Server-Datenquelle dauert ungewöhnlich lang, können Sie die Leistung verbessern, indem Sie die Umgebungsvariable DB2_DJ_COMM festlegen, damit die Oberfläche geladen wird, wenn der Server mit zusammengeschlossenen Datenbanken initialisiert wird und nicht dann, wenn Sie versuchen, auf die Datenquelle zuzugreifen. Legen Sie die Umgebungsvariable DB2_DJ_COMM fest, damit die Oberflächenbibliothek integriert wird, die zu der Oberfläche gehört, die Sie in Schritt 5 angegeben haben. Beispiel:

Auf DB2 für Windows NT-Servern:

```
db2set DB2_DJ_COMM=djxmssql3.d11
```

Auf DB2 für AIX-Servern:

```
db2set DB2_DJ_COMM=libmssql3.a
```

Stellen Sie sicher, dass sich auf keiner Seite des Gleichheitszeichens (=) Leerzeichen befinden.

Weitere Informationen zu den Oberflächenbibliotheksnamen finden Sie im Handbuch *SQL Reference*.

Schritt 4: DB2-Exemplar erneut starten (nur AIX)

Starten Sie das DB2-Exemplar erneut, um sicherzustellen, dass die Umgebungsvariablen in dem Programm festgelegt sind. Wenn Sie das Exemplar erneut starten, aktualisieren Sie das DB2-Exemplar so, dass es die vorgenommenen Änderungen übernimmt. Starten Sie das DB2-Exemplar mit den folgenden Befehlen erneut:

```
db2stop  
db2start
```

Schritt 5: Erstellen der Oberfläche

DB2 Universal Database hat zwei verschiedene Protokolle, die sogenannten Oberflächen, die Sie verwenden können, um auf Microsoft SQL Server-Datenquellen zuzugreifen. Oberflächen sind die Mittel, mit denen Server mit zusammengeschlossenen Datenbanken mit Datenquellen kommunizieren und Daten von Datenquellen abrufen. Die Oberfläche, die Sie verwenden, hängt von der Plattform ab, auf der DB2 Universal Database ausgeführt wird. Verwenden Sie Tabelle 2 als Richtlinie zur Auswahl der passenden Oberfläche.

Tabelle 2. ODBC-Treiber

ODBC-Treiber	Plattform	Oberflächenname
ODBC 3.0-Treiber (oder höher)	Windows NT	DJXMSSQL3
MERANT DataDirect Connect ODBC 3.6-Treiber	AIX	MSSQLODBC3

Verwenden Sie die Anweisung CREATE WRAPPER, um die Oberfläche anzugeben, die verwendet werden soll, um auf Microsoft SQL Server-Datenquellen zuzugreifen. Das folgende Beispiel zeigt eine CREATE WRAPPER-Anweisung:

```
CREATE WRAPPER DJXMSSQL3
```

Dabei ist DJXMSSQL3 der Standardoberflächenname, der auf einem DB2-Server für Windows NT (mit dem ODBC 3.0-Treiber) verwendet wird. Wenn Sie einen DB2-Server für AIX haben, würden Sie den Oberflächennamen MSSQLODBC3 angeben.

Sie können den Standardoberflächennamen mit einem Namen ihrer Wahl ersetzen. Wenn Sie dies machen, müssen Sie allerdings den Parameter LIBRARY und den Namen der Oberflächenbibliothek für Ihre Serverplattform mit zusammengesetzten Datenbanken in der Anweisung CREATE WRAPPER angeben. Beispiel:

Auf DB2 für Windows NT-Servern:

```
CREATE WRAPPER oberflächenname LIBRARY  
'djmssql3.dll'
```

Dabei ist *oberflächenname* der Name, den Sie der Oberfläche geben wollen, und 'djmssql3.dll' ist der Bibliotheksname.

Auf DB2 für AIX-Servern:

```
CREATE WRAPPER oberflächenname  
LIBRARY 'libmssql3.a'
```

Dabei ist *oberflächenname* der Name, den Sie der Oberfläche geben wollen, und 'libdjmssql.a' ist der Bibliotheksname.

Weitere Informationen zu Oberflächenbibliotheksnamen finden Sie bei der Anweisung CREATE WRAPPER im Handbuch *SQL Reference*.

Schritt 6: Erstellen des Servers

Verwenden Sie die Anweisung CREATE SERVER, um alle Microsoft SQL Server-Server zu definieren, zu denen Sie eine Verbindung herstellen wollen. Beispiel:

```
CREATE SERVER sqlserver TYPE MSSQLSERVER VERSION 7.0 WRAPPER djmssql3  
OPTIONS (NODE 'sql-knoten', DBNAME 'datenbankname')
```

Dabei gilt Folgendes:

sqlserver

Der Name, den Sie dem Microsoft SQL Server-Server zuordnen. Dieser Name muss eindeutig sein.

MSSQLSERVER

Der Datenquellentyp, für den Sie den Zugriff konfigurieren.

7.0 Die Microsoft SQL Server-Version, auf die Sie zugreifen. DB2 Universal Database unterstützt die Versionen 6.5 und 7.0 von Microsoft SQL Server.

DJXMSSQL3

Der Oberflächenname, den Sie in der Anweisung CREATE WRAPPER definiert haben.

'*sql-knoten*'

Der Systemdatenquellentyp, der auf die Version von Microsoft SQL Server verweist, auf die Sie zugreifen. Beachten Sie bei diesem Wert die Groß-/Kleinschreibung. DB2 Universal Database unterstützt die Versionen 6.5 und 7.0 von Microsoft SQL Server.

Obwohl der Name des Knotens (Systemdatenquellentyp) als Option in der Anweisung CREATE SERVER angegeben wird, ist er für Microsoft SQL Server-Datenquellen erforderlich.

Zusätzliche Optionen, die Sie mit der Anweisung CREATE WRAPPER verwenden können, finden Sie im Handbuch *SQL Reference*.

'*datenbankname*'

Der Name der Datenbank, zu der Sie eine Verbindung herstellen.

Obwohl der Name der Datenbank als Option in der Anweisung CREATE SERVER angegeben wird, ist er für Microsoft SQL Server-Datenquellen erforderlich.

Schritt 7: Erstellen einer Benutzerzuordnung

Wenn sich die Benutzer-ID oder das Kennwort auf dem Server mit zusammengesetzten Datenbanken von einer Benutzer-ID oder einem Kennwort in einer Microsoft SQL Server-Datenquelle unterscheiden, verwenden Sie die Anweisung CREATE USER MAPPING, um die lokale Benutzer-ID der Benutzer-ID und dem Kennwort zuzuordnen, die/das in der Microsoft SQL Server-Datenquelle definiert ist. Beispiel:

```
CREATE USER MAPPING FOR db2-benutzer SERVER servername
OPTIONS (REMOTE_AUTHID 'ms-sql-benutzer', REMOTE_PASSWORD
'daylite')
```

Dabei gilt Folgendes:

db2-benutzer

Die lokale Benutzer-ID, die Sie einer Benutzer-ID zuordnen, die in einer Microsoft SQL Server-Datenquelle definiert ist.

servername

Der Name des Servers, den Sie in der Anweisung CREATE SERVER definiert haben.

'ms-sql-benutzer'

Die Benutzer-ID in der Microsoft SQL Server-Datenquelle, der Sie *db2-benutzer* zuordnen. Beachten Sie bei diesem Wert die Groß-/Kleinschreibung.

'daylite'

Das Kennwort, das *'ms-sql-benutzer'* zugeordnet ist. Beachten Sie bei diesem Wert die Groß-/Kleinschreibung.

Zusätzliche Optionen, die Sie mit der Anweisung CREATE USER MAPPING verwenden können, finden Sie im Handbuch *SQL Reference*.

Schritt 8: Erstellen von Kurznamen für Tabellen und Sichten

Ordnen Sie jeder Sicht oder Tabelle, die sich in Ihrer Microsoft SQL Server-Datenquelle befindet und auf die Sie zugreifen wollen, einen Kurznamen zu. Sie verwenden diese Kurznamen, wenn Sie die Microsoft SQL Server-Datenquelle abfragen. Verwenden Sie die Anweisung CREATE NICKNAME, um einen Kurznamen zuzuweisen. Beachten Sie bei Kurznamen die Groß-/Kleinschreibung. Das folgende Beispiel zeigt eine CREATE NICKNAME-Anweisung:

```
CREATE NICKNAME ms-sql-verkäufe FOR servername.verkaufsdaten.europa
```

Dabei gilt Folgendes:

ms-sql-verkäufe

Ein eindeutiger Kurzname für die Microsoft SQL Server-Tabelle oder -Sicht.

servername.verkaufsdaten.europa

Ein dreiteiliger Bezeichner, der das folgende Format hat:

datenquellenservername.ferner-schemenname.ferner-tabellenname

Für die Abschnitte *ferner-schemenname* und *ferner-tabellenname* werden doppelte Anführungszeichen empfohlen.

Wenn Sie einen Kurznamen erstellen, versucht DB2 auf die Datenquellenkatalogtabellen zuzugreifen (Microsoft SQL Server bezeichnet diese als Systemtabellen). Diese Abfrage testet die Verbindung zu der Datenquelle. Wenn die Verbindung fehlschlägt, empfangen Sie eine Fehlermeldung.

Wiederholen Sie diesen Schritt für alle Datenbanktabellen und -sichten, für die Sie Kurznamen erstellen wollen.

Weitere Informationen zu der Anweisung CREATE NICKNAME finden Sie im Handbuch *SQL Reference*. Weitere Informationen zu Kurznamen im Allgemeinen und zum Prüfen von Datentypzuordnungen finden Sie im Handbuch *DB2 Systemverwaltung*.

Schritt 9: Optional: Abrufen von ODBC-Traces

Wenn beim Zugriff auf die Datenquelle Probleme auftreten, können Sie ODBC-Trace-Informationen abrufen, um diese Probleme zu analysieren und zu lösen. Verwenden Sie das Trace-Tool, das von ODBC Data Source Administrator angeboten wird, um sicherzustellen, dass der ODBC-Trace richtig funktioniert. Das Aktivieren des Trace beeinträchtigt Ihre Systemleistung, deshalb sollten Sie den Trace ausschalten, wenn Sie die Probleme gelöst haben.

Übersicht über Microsoft SQL Server-Codepages

Microsoft SQL Server unterstützt viele gemeinsame Codepageoptionen für Unterstützung der Landessprache, die DB2 UDB unterstützt. Datenquellen, die denselben codierten Zeichensatz wie DB2 verwenden, erfordern keine Übersetzung. In Tabelle 3 finden Sie eine Liste der Codepages, die sowohl von DB2 Universal Database als auch Microsoft SQL Server unterstützt werden.

Tabelle 3. DB2 UDB- und Microsoft SQL Server-Codepage-Optionen

Codepage	Unterstützte Sprache
1252	ISO-Zeichensatz
850	Mehrsprachig
437	Amerikanisches Englisch
874	Thailändisch
932	Japanisch
936	Vereinfachtes Chinesisch
949	Koreanisch
950	Traditionelles Chinesisch
1250	Mitteleuropäisch
1251	Kyrillisch
1253	Griechisch
1254	Türkisch
1255	Hebräisch
1256	Arabisch

Wenn der DB2Server mit zusammengeschlossenen Datenbanken und der Microsoft SQL Server-Server mit zusammengeschlossenen Datenbanken unterschiedliche Codepages zur Unterstützung der Landessprache ausführen, müssen Sie entweder Ihre Microsoft SQL Server-Datenquellen so konfigurieren, dass sie mit diesen Werten übereinstimmen, oder der Client-Code muss in der Lage sein, die Abweichung zu entdecken und sie als Fehler zu markieren oder die Daten mit seiner eigenen Semantik zuzuordnen. Wenn keine Konvertierungstabelle von der Quellen-Codepage zu der Ziel-Codepage gefunden werden kann, setzt DB2 eine Fehlermeldung ab. Weiter Informationen finden Sie in Ihrer Microsoft SQL Server-Dokumentation.

Teil 3. Verwaltung

Systemverwaltung: Konzept	63
Kapitel 8. Entwerfen der physischen Datenbank	63
Partitionierungsschlüssel	63
Entwerfen von Knotengruppen	63
Kapitel 9. Entwerfen verteilter Datenbanken	63
Aktualisieren mehrerer Datenbanken	63
Kapitel 13. Hohe Verfügbarkeit in der Windows NT-Umgebung	64
Vor Ausführung von Dienstprogramm DB2MSCS Warmstart der Maschine erforderlich	64
Kapitel 14. DB2 und hohe Verfügbarkeit unter Sun Cluster 2.2.	64
Veritas-Unterstützung unter Solaris	64
Anhang B. Namenskonventionen	66
Hinweise zu Benutzer-IDs und Schemennamen mit mehr als 8 Zeichen	66
Benutzer-IDs und Kennwörter	66
Anhang D. Inkompatibilitäten zwischen Releases.	66
Windows NT DLFS inkompatibel mit Norton Uti- lities.	66
SET CONSTRAINTS ersetzt durch SET INTE- GRITY	67
Anhang E. Unterstützung von Landessprachen (NLS)	67
Landessprachenversionen von DB2 Version 7	67
Dateigruppen für Steuerzentrale und Doku- mentation	69
Einstellung der Länderangaben für den DB2- Verwaltungs-Server.	70
DB2 UDB Supports the Baltic Rim Code Page (MS-1257) on Windows Platforms	71
Ableiten der Werte von Codepages	71
Unterstützung von Landescodes und Codepages Zeichensätze	71
Systemverwaltung: Implementierung	73
Hinzufügen oder Erweitern von DMS-Behältern (Neuer Prozess)	73
Kapitel 1. Verwalten von DB2 mit den Tools für die grafische Benutzerschnittstelle	73
Kapitel 3. Erstellen einer Datenbank	74
Erstellen eines Tabellenbereichs.	74
Unformatierte Ein-/Ausgabe bei Linux	74
Erstellen einer Sequenz	75
Vergleichen von IDENTITY-Spalten und Sequen- zen	77
Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation	77
Kapitel 4. Ändern einer Datenbank	78
Hinzufügen eines Behälters zu einem SMS- Tabellenbereich in einer Partition	78
Ändern einer IDENTITY-Spalte.	78
Ändern einer Sequenz.	79
Löschen einer Sequenz	79
Wechseln des Status eines Tabellenbereichs.	80
Modifizieren von Behältern in einem DMS- Tabellenbereich	80
Kapitel 5. Steuern des Datenbankzugriffs	80
Sequenzzugriffsrechte	80
Datenverschlüsselung	80
Kapitel 8. Wiederherstellen einer Datenbank	82
Verwendung der zurückgestellten Ein-/Ausgabe Teilsicherung und -wiederherstellung.	82
Wiederherstellen von Teilsicherungsimages.	84
Parallele Wiederherstellung	86
Sicherung auf benannte Pipes	87
Sicherung von einem geteilten Image.	87
Protokollarchivierung nach Bedarf.	89
Protokollspiegelung	90
Unterstützung von plattformübergreifendem Sichern und Wiederherstellen unter Sun Solaris und HP.	91
Überlegungen zum DB2 Data Links Manager/Überlegungen zum Dienstprogramm BACKUP	91
Überlegungen zum DB2 Data Links Manager/Überlegungen zu den Dienst- programmen RESTORE und ROLLFORWARD.	92
Wiederherstellen von Datenbanken von einer Offlinesicherung ohne aktualisierendes Wieder- herstellen	94
Wiederherstellen von Datenbanken und Tabellen- bereichen und aktualisierendes Wiederherstellen bis zum Ende der Protokolle.	95
DB2 Data Links Manager und Interaktionen bei Wiederherstellung	95
Erkennen von Situationen, die Abstimmung erfordern.	100
Anhang C. Benutzer-Exit zur Datenbank- wiederherstellung	101
Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitionsserver	102
Anhang I. Knotenübergreifende Hochgeschwindig- keitskommunikation	102
Aktivieren von DB2 zur Ausführung mit VI	102
Systemverwaltung: Optimierung	105
Kapitel 3. Überlegungen zu Anwendungen	105
Angaben der Isolationsstufe	105
Anpassen der Optimierungsklasse	105
Dynamische Compound-Anweisungen	106
Kapitel 4. Überlegungen zur Umgebung	107
Größere Indexschlüssel	107
Kapitel 5. Systemkatalogstatistiken	107
Erfassen und Verwenden von Verteilungs- statistiken	107
Regeln zur Aktualisierung von Katalog- statistiken	108
Subelementstatistik	108
Kapitel 6. Der SQL-Compiler	111
Replizierte Übersichtstabellen	111

Datenzugriffskonzepte und Optimierung	113
Kapitel 8. Leistung bei der Ausführung.	114
Verwalten des Datenbankpufferpools	114
Verwalten mehrerer Datenbankpufferpools	115
Kapitel 9. Verwenden von Governor.	116
Kapitel 13. Konfigurieren von DB2	116
Zwischenspeicher für Sortierlisten (sorheap)	116
Schwellenwert für Sortierspeicher (sheapthres)	116
Maximale Anzahl Sperren pro Anwendung (maxlocks)	117
Konfigurieren von DB2/DB2 Data Links Manager/Data Links-Token-Verfallsintervall (dl_expint)	117
Datenbankkonfigurationsparameter MIN_DEC_DIV_3	118
Maximaler Zwischenspeicher für Anwendungssteuerung (app_ctl_heap_sz)	119
Zwischenspeicher für den Datenbanksystemmonitor (mon_heap_sz)	119
Maximale Anzahl aktiver Anwendungen (max_appls)	119
Vor dem bedingten Prüfpunkt zu schreibende Protokollsätze (softmax)	119
Geänderte Seiten protokollieren (trackmod)	120
Datenbankprotokollpfad ändern (newlogpath)	120
Pfad zu Protokolldateien (logpath)	121
Maximaler Speicher für Sperrenliste (locklist)	121
Anhang A. DB2-Registrierungsvariablen und DB2-Umgebungsvariablen.	122
Tabelle der neuen und geänderten Registrierungsvariablen	122
Anhang C. EXPLAIN-Programme (SQL)	127

Administering Satellites Guide and Reference	129
Setting up Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition as Satellites.	129
Prerequisites.	129
Installation Considerations	130
Configuring the Version 7.2 System for Synchronization	131
Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System	132
Upgrading Version 6 DB2 Enterprise Edition for Use as the DB2 Control Server	132
Upgrading a Version 6 Steuerzentrale and Satellitenverwaltungszentrale	139

Command Reference	141
db2batch - Benchmark Tool.	141
db2cap (new command).	141
db2cap - CLI/ODBC Static Package Binding Tool	141
db2ckrst (new command)	143
db2ckrst - Check Incremental Restore Image Sequence	143
db2gncol (new command)	145
db2gncol - Update Generated Column Values	145
db2inidb - Initialize a Mirrored Database	146
db2look - DB2 Statistics Extraction Tool	147
db2updv7 - Update Database to Version 7 Current Fix Level	147

New Command Line Processor Option (-x, Suppress printing of column headings)	148
True Type Font Requirement for DB2 CLP.	148
ADD DATALINKS MANAGER	149
ARCHIVE LOG (new command)	149
Archive Log	149
BACKUP DATABASE	151
Syntax Diagram	151
DB2 Data Links Manager Considerations	152
BIND	152
CALL	153
DROP DATALINKS MANAGER (new command)	154
DROP DATALINKS MANAGER	154
EXPORT	157
GET DATABASE CONFIGURATION	158
GET ROUTINE (new command)	159
GET ROUTINE.	159
GET SNAPSHOT	159
IMPORT	160
LIST HISTORY	160
LOAD.	162
PING (new command)	163
PING	163
PUT ROUTINE (new command)	165
PUT ROUTINE.	165
RECONCILE	166
REORGANIZE TABLE	166
RESTORE DATABASE	167
Syntax.	167
DB2 Data Links Manager Considerations	168
ROLLFORWARD DATABASE	168
Documentation Error in CLP Return Codes	168

Versetzen von Daten Dienstprogramme und Referenz	169
Kapitel 2. Import	169
Verwenden von IMPORT mit gepufferten INSERT-Operationen	169
Kapitel 3. Load	169
Statusangaben für anstehende Aktionen nach einer Ladeoperation	169
LOAD-Rahmenbedingungen und -Einschränkungen	169
Dateitypänderungswert "totalreespace"	169
Kapitel 4. AutoLoader	170
"rexecd" erforderlich für Ausführung von Auto-loader bei aktivierter Authentifizierung.	170

Replikation Referenzhandbuch.	171
Replikations-Server und Nicht-IBM Server.	171
Replikation unter Windows 2000	171
Bekannter Fehler beim Speichern von SQL-Dateien	171
DB2-Verwaltung	171
Dienstprogramm "Data Difference" im Internet	171
Kapitel 3. Beispielszenario für die Datenreplikation	171
Replikationsszenarios.	171
Kapitel 5. Planung der Replikationsumgebung	173
Tabellen- und Spaltennamen	173
DATALINK-Replikation	173
LOB-Einschränkungen	173
Planen der Replikation	173

Kapitel 6. Einrichten der Replikationsumgebung	174	disable_MQFunctions.	206
Vorbedingung für beliebige Tabellenreplikation	174	disable_MQFunctions.	206
Einrichten Ihrer Replikationsumgebung	174		
Kapitel 8. Fehlerbestimmung	174		
Kapitel 9. Capture und Apply für AS/400	176		
Kapitel 10. Capture und Apply für OS/390	176		
Vorbedingungen für DB2 DataPropagator für OS/390	176		
UNICODE- und ASCII-Schemen für Codeumsetzung unter OS/390	177		
Auswählen eines Schemas für Codeumsetzung	177		
Einstellen von Schemen für Codeumsetzung	177		
Kapitel 11. Capture und Apply für UNIX-Plattformen	179		
Einstellen der Umgebungsvariablen für Capture und Apply unter UNIX und Windows	179		
Kapitel 14. Tabellenstrukturen	179		
Kapitel 15. Nachrichten der Programme Capture und Apply	179		
Anhang A. Starten der Programme Capture und Apply aus einer Anwendung heraus	180		
System Monitor Guide and Reference	181		
db2ConvMonStream	181		
Troubleshooting Guide	183		
Starting DB2 on Windows 95, Windows 98, and Windows ME When the User Is Not Logged On.	183		
Chapter 2. Troubleshooting the DB2 Universal Database Server	186		
Verwenden von DB2 Universal Database auf 64-Bit-Plattformen	187		
Kapitel 5. Konfiguration	187		
LOCKLIST	187		
shmsys:shminfo_shmmax	187		
Kapitel 6. Einschränkungen.	187		
XML Extender Verwaltung und Programmierung	189		
MQSeries	191		
Installation und Konfiguration für die DB2 MQSeries-Funktionen	191		
Installieren von MQSeries	191		
Installieren von MQSeries AMI	192		
Aktivieren der DB2 MQSeries-Funktionen	192		
MQSeries - Nachrichtendarstellung	193		
Nachrichtenstruktur	193		
MQSeries - Funktionsübersicht	194		
Einschränkungen	195		
Fehlercodes	195		
Einsatzszenarios	195		
Basis-Nachrichtenübertragung	195		
Senden von Nachrichten	197		
Nachrichten abrufen	197		
Anwendungsübergreifende Konnektivität	199		
R/R-Kommunikation	200		
P/S.	201		
enable_MQFunctions	204		
enable_MQFunctions	204		

Systemverwaltung: Konzept

Kapitel 8. Entwerfen der physischen Datenbank

Partitionierungsschlüssel

Im Unterabschnitt „Überlegungen zum Aufbau von Knotengruppen“ des Abschnitts zum Entwerfen von Knotengruppen muss der folgende Text aus dem Abschnitt, in dem die bei der Definition von Partitionierungsschlüsseln zu beachtenden Punkte beschrieben werden, nur dann **gelöscht** werden, wenn DB2_UPDATE_PART_KEY=ON angegeben wurde:

Anmerkung: Bei DB2_UPDATE_PART_KEY=OFF gelten die Einschränkungen weiterhin.

Anmerkung: Ab FixPak 3 lautet der Standardwert ON.

- Sie können den Spaltenwert des Partitionierungsschlüssels für eine Zeile in der Tabelle nicht aktualisieren.
- Sie können Spaltenwerte für den Partitionierungsschlüssel nur löschen oder einfügen.

Entwerfen von Knotengruppen

Im Abschnitt „Entwerfen von Knotengruppen“ muss der letzte Satz der Unterabschnitte „Überlegungen zum Aufbau von Knotengruppen“ und „Replizierte Übersichtstabellen“ gelöscht werden:

Das Schlüsselwort REPLICATED kann nur für eine Übersichtstabelle angegeben werden, die mit der Option REFRESH DEFERRED wurde.

Kapitel 9. Entwerfen verteilter Datenbanken

Aktualisieren mehrerer Datenbanken

Im Abschnitt "Aktualisieren mehrerer Datenbanken" enthält die Liste der Einrichtungsschritte eine Ungenauigkeit. Schritt 4 lautet bisher:

Vorkompilieren Ihres Anwendungsprogramms, um eine Verbindung des Typs 2, d. h. CONNECT 2 im Befehl PRECOMPILE PROGRAM, sowie das einphasige Festschreiben, d. h. SYNCPOINT ONEPHASE im Befehl PRECOMPILE PROGRAM, anzugeben (siehe die Beschreibung im Application Development Guide.

Dies sollte geändert werden in:

Vorkompilieren Ihres Anwendungsprogramms, um eine Verbindung des Typs 2, d. h. CONNECT 2 im Befehl PRECOMPILE PROGRAM, sowie das einphasige Festschreiben, d. h. SYNCPOINT TWOPHASE im Befehl PRECOMPILE PROGRAM, anzugeben (siehe die Beschreibung im Application Development Guide.

Kapitel 13. Hohe Verfügbarkeit in der Windows NT-Umgebung

Vor Ausführung von Dienstprogramm DB2MSCS Warmstart der Maschine erforderlich

Mit dem Dienstprogramm DB2MSCS wird die erforderliche Konfiguration durchgeführt, um in DB2 die Funktionsübernahmeunterstützung in der Microsoft Cluster Service-Umgebung zu aktivieren. Damit das Dienstprogramm DB2MSCS erfolgreich ausgeführt werden kann, muss der Cluster Service die Ressourcen-DLL, `db2wolf.dll`, die sich im Verzeichnis `%ProgramFiles%\SQLLIB\bin` befindet, lokalisieren können.

Das Installationsprogramm von DB2 UDB Version 7 definiert die Systemumgebungsvariable `PATH` so, dass sie auf das Verzeichnis `%ProgramFiles%\SQLLIB\bin` zeigt. Nach der Installation ist jedoch kein Warmstart der Maschine erforderlich, wenn Sie das Betriebssystem Windows 2000 verwenden.

Wenn Sie das Dienstprogramm DB2MSCS ausführen wollen, müssen Sie einen Warmstart der Maschine durchführen, damit die Umgebungsvariable `PATH` für den Cluster Service aktualisiert wird.

Kapitel 14. DB2 und hohe Verfügbarkeit unter Sun Cluster 2.2

DB2 Connect wird unter Sun Cluster 2.2 unterstützt, wenn Folgendes gegeben ist:

- Das Protokoll für den Host ist TCP/IP (nicht SNA).
- Es wird keine zweiphasige Festschreibung verwendet. Diese Einschränkung wird gelockert, wenn der Benutzer das SPM-Protokoll so konfiguriert, dass es sich auf einem gemeinsamen Datenträger befindet (mit Hilfe des Konfigurationsparameters `spm_log_path` des Datenbankmanagers), und die Maschine für die Funktionsübernahme dieselbe TCP/IP-Konfiguration hat (denselben Host-Namen, dieselbe IP-Adresse, usw.).

Veritas-Unterstützung unter Solaris

DB2 unterstützt jetzt Veritas, das Cluster-Unterstützung für DB2 High Availability unter Solaris bietet.

Beschreibung

Setzt ein DB2 UDB-Exemplar online oder offline und überwacht es.

Eingangspunkte

Online

Aktivieren Sie ein Exemplar mit `db2start`.

Offline

Inaktivieren Sie ein Exemplar mit `db2stop`.

Monitor

Bestimmt, ob ein bestimmtes DB2-Exemplar aktiv ist. Verwendet die entsprechende Prozessüberwachung und (optional) Datenbanküberwachung.

Clean Entfernt DB2-Exemplarressourcen.

Attribut	Typ	Definition
probeDatabase	string	Zu überwachende Datenbank
instanceOwner	string	Name des Exemplareigners
instanceHome	string	Ausgangsverzeichnis des Exemplareigners
probeTable	string	Zu überwachende Tabelle in probeDatabase
monitorLevel	integer	1 für Prozessüberwachung, 2 für DB-Überwachung
nodeNumber	integer	Knotennummer des zu startenden Exemplars (nicht gesetzt: EE)

Typdefinition

```

type DB2UDB (
    static int CleanTimeout = 240
    static int MonitorTimeout = 30
    static int OfflineTimeout = 240
    static int OnlineRetryLimit = 2
    static int OnlineTimeout = 120
    static int OnlineWaitLimit = 1
    static int RestartLimit = 3
    static int ToleranceLimit = 1
    static str ArgList[] =
{ probeDatabase, instanceOwner, instanceHome, probeTable,
  monitorLevel, nodeNumber }
    NameRule = resource.db2udb
    str probeDatabase
    str instanceOwner
    str instanceHome
    str probeTable
    int monitorLevel
    int nodeNumber
)

```

Beispielkonfiguration

```

DB2UDB db2_resource_n0 (
    probeDatabase = sample
    probeTable = vcstable
    instanceOwner = stevera
    instanceHome = "/export/home/stevera"
    monitorLevel = 2
)

```

Installation

- Erstellen Sie das Verzeichnis /opt/VRTSvcs/bin/DB2UDB.
- Kopieren Sie die Dateien online, offline, monitor, clean, DB2UDBAgent in /opt/VRTSvcs/bin/DB2UDB, und stellen Sie sicher, dass sie als ausführbar markiert sind.
- Kopieren Sie die Datei db2udb.type.cf in /etc/VRTSvcs/conf/config.
- Stoppen Sie den Cluster (z. B. hastop -all).
- Fügen Sie die Zeile include db2udb.type.cf in der Datei main.cf nach der Zeile include types.cf hinzu.
- Prüfen Sie mit /opt/VRTSvcs/bin/hacf -verify /etc/VRTSvcs/conf/config, ob die Cluster-Konfiguration gültig ist.

Sie können jetzt die notwendigen DB2-Ressourcen zur Steuerung von DB2-Exemplaren erstellen.

Anhang B. Namenskonventionen

Hinweise zu Benutzer-IDs und Schemennamen mit mehr als 8 Zeichen

- DB2-Produkte der Version 7 unterstützen auf Windows-32-Bit-Plattformen Benutzer-IDs mit bis zu 30 Zeichen. Durch die native Unterstützung von Windows NT und Windows 2000 sind Benutzer-IDs jedoch praktisch auf 20 Zeichen beschränkt.
- DB2 Version 7 unterstützt auch 32-Bit-Clients anderer Betriebssysteme, die mit Benutzer-IDs mit mehr als 8 Zeichen Verbindungen zu Windows NT und Windows 2000 herstellen, wenn die Benutzer-ID und das Kennwort explizit angegeben sind. Dies schließt Verbindungen mit Client- oder DCE-Authentifizierung aus.
- Für die DCE-Authentifizierung gilt weiterhin auf allen Plattformen die Beschränkung der Benutzer-ID auf 8 Zeichen.
- Die im SQL-Kommunikationsbereich von einer erfolgreichen CONNECT- oder ATTACH-Operation zurückgegebene Authentifizierungs-ID wird auf 8 Zeichen abgeschnitten. Die SQLWARN-Felder enthalten Warnungen, wenn die Authentifizierungs-ID abgeschnitten wird. Zusätzliche Informationen finden Sie in der Beschreibung der Anweisung CONNECT im Handbuch *SQL Reference*.
- Die vom Befehlszeilenprozessor bei einer erfolgreichen CONNECT- oder ATTACH-Operation zurückgegebene Authentifizierungs-ID wird auf 8 Zeichen abgeschnitten. Ein Auslassungszeichen (...) wird an die Authentifizierungs-ID angehängt, um das Abschneiden darzustellen.
- DB2 Version 7 unterstützt Schemennamen mit bis zu 30 Byte. Dabei gelten allerdings die folgenden Ausnahmen:
 - Tabellen mit Schemennamen mit mehr als 18 Byte können nicht repliziert werden.
 - Benutzerdefinierte Typen dürfen nur Schemennamen mit maximal 8 Byte haben.

Benutzer-IDs und Kennwörter

Ändern Sie im Abschnitt „Benutzer-IDs und Kennwörter“ die Angabe „A - Z“ wie folgt:

Lateinische Single-Byte-Groß- und Kleinbuchstaben (A...Z, a...z). Die Unterstützung anderer Buchstaben und Zeichen hängt von der verwendeten Codepage ab. Weitere Informationen zur Codepage-Unterstützung finden Sie im Anhang „Unterstützung von Landessprachen“.

Anhang D. Inkompatibilitäten zwischen Releases

Windows NT DLFS inkompatibel mit Norton Utilities

Windows NT Data Links File System (DLFS) ist inkompatibel mit Norton Utilities. Wenn eine Datei aus einem DLFS-gesteuerten Laufwerk gelöscht wird, tritt eine Kernel-Ausnahme auf: Fehler 0x1E (Kernel Mode Exception Not Handled). Die Ausnahmebedingung ist 0xC0000005 (ungültiger Zugriff). Dieser ungültige Zugriff tritt auf, weil der Treiber der Norton Utilities nach dem Treiber des DLFS-Filters geladen wird.

Eine vorübergehende Notlösung besteht darin, den DLFS-Dreiber nach dem Treiber der Norton Utilities zu laden. Hierfür muss der Start des DLFS-Dreibers manuell erfolgen. Klicken Sie **Start** an, und wählen Sie **Einstellungen**—> **Systemsteuerung**—> **Geräte**—> **DLFS** aus. Geben Sie die Auswahl **manuell** an.

Sie können eine Stapeldatei erstellen, die dem Systemstartordner hinzugefügt werden kann und die den DLFS-Dreiber und den DLFM-Service beim Systemstart lädt. Der Inhalt der Stapeldatei sieht wie folgt aus:

```
net start dlfsd
net start "dlfm service"
```

Nennen Sie die Datei start_dlfs.bat und kopieren Sie sie in das Verzeichnis C:\WINNT\Profiles\Administrator\Startmenü\Programme\Autostart.

Nur der Administrator hat das Zugriffsrecht, den DLFS-Filtertreiber und den DLFM-Service zu laden.

SET CONSTRAINTS ersetzt durch SET INTEGRITY

Die Anweisung SET CONSTRAINTS wurde durch die Anweisung SET INTEGRITY ersetzt. Beide Anweisungen werden in DB2 UDB V7 akzeptiert, um Abwärtskompatibilität zu gewährleisten.

Anhang E. Unterstützung von Landessprachen (NLS)

Landessprachenversionen von DB2 Version 7

DB2 Version 7 ist verfügbar in Englisch, Französisch, Deutsch, Italienisch, Spanisch, brasilianischem Portugiesisch, Japanisch, Koreanisch, vereinfachtem Chinesisch, traditionellem Chinesisch, Dänisch, Finnisch, Norwegisch, Schwedisch, Tschechisch, Niederländisch, Ungarisch, Polnisch, Türkisch, Russisch, Bulgarisch und Slowenisch.

Auf UNIX-basierten Plattformen können die DB2-Produktnachrichten und die Bibliothek in unterschiedlichen Sprachen installiert werden. Das DB2-Installationsprogramm speichert die Nachrichtenkatalog-Dateigruppen in dem in der Regel verwendeten Verzeichnis für die Ländereinstellung für eine bestimmte Plattform (siehe Tabellen unten). Tabelle 4 bietet Informationen zu AIX, HP-UX und Solaris. Tabelle 5 auf Seite 68 bietet Informationen zu Linux, Linux/390, SGI und Dynix.

Tabelle 4. AIX, HP-UX, Solaris

Betriebssystem	AIX		HP-UX		Solaris	
Sprache	Länderspez. Angaben	Code-page	Länderspez. Angaben	Code-page	Länderspez. Angaben	Code-page
Französisch	fr_FR	819	fr_FR.iso88591	819	fr	819
	Fr_FR	850	fr_FR.roman8	1051		
Deutsch	de_DE	819	de_DE.iso88591	819	de	819
	De_DE	850	de_DE.roman8	1051		
Italienisch	it_IT	819	it_IT.iso88591	819	it	819
	It_IT	850	it_IT.roman8	1051		

Tabelle 4. AIX, HP-UX, Solaris (Forts.)

Spanisch	es_ES	819	es_ES.iso88591	819	es	819
	Es_ES	850	es_ES.roman8	1051		
Brasilianisches Portugiesisch	pt_BR	819			pt_BR	819
Japanisch	ja_JP	954	ja_JP.eucJP	954	ja	954
	Ja_JP	932				
Koreanisch	ko_KR	970	ko_KR.eucKR	970	ko	970
Vereinfachtes Chinesisch	zh_CN	1383	zh_CN.hp15CN	1383	zh	1383
	Zh_CN.GBK	1386				
Traditionelles Chinesisch	zh_TW	964	zh_TW.eucTW	964	zh_TW	964
	Zh_TW	950	zh_TW.big5	950	zh_TW.BIG5	950
Dänisch	da_DK	819	da_DK.iso88591	819	da	819
	Da_DK	850	da_DK.roman8	1051		
Finnisch	fi_FI	819	fi_FI.iso88591	819	fi	819
	Fi_FI	850	fi_FI.roman8	1051		
Norwegisch	no_NO	819	no_NO.iso88591	819	no	819
	No_NO	850	no_NO.roman8	1051		
Schwedisch	sv_SE	819	sv_SE.iso88591	819	sv	819
	Sv_SE	850	sv_SE.roman8	1051		
Tschechisch	cs_CZ	912				
Ungarisch	hu_HU	912				
Polnisch	pl_PL	912				
Niederländisch	nL_NL	819				
	NI_NL	850				
Türkisch	tr_TR	920				
Russisch	ru_RU	915				
Bulgarisch	bg_BG	915	bg_BG.iso88595	915		
Slowenisch	sl_SI	912	sl_SI.iso88592	912	sl_SI	912

Tabelle 5. Linux, Linux/390, SGI, Dynix

Betriebssystem	Linux		Linux/390		SGI		Dynix	
	Länderspez. Angaben	Code-page						
Französisch	fr	819	fr	819			fr	819
Deutsch	de	819	de	819			de	819
Italienisch							es	819
Spanisch								
Brasilianisches Portugiesisch								
Japanisch	ja_JP.ujis	954	ja_JP.ujis	954			ja_JPEUC	954

Tabelle 5. Linux, Linux/390, SGI, Dynix (Forts.)

Koreanisch	ko	970	ko	970	ko_KO.euc	970		
Vereinfachtes Chinesisch	zh zh_CN.GBK	1386	zh zh_CN.GBK	1386				
Traditionelles Chinesisch	zh_TW.Big5	950	zh_TW.Big5	950				
Dänisch								
Finnisch								
Norwegisch								
Schwedisch								
Tschechisch								
Ungarisch								
Polnisch								
Niederländisch							nl	819
Türkisch								
Russisch								
Bulgarisch								
Slowenisch								

Wenn Ihr System dieselben Codepages, jedoch andere Namen für die Ländereinstellung verwendet, können Sie dennoch die übersetzten Nachrichten sehen. Dazu müssen Sie eine Verknüpfung zum entsprechenden Nachrichtenverzeichnis herstellen.

Wenn z. B. Ihre AIX-Maschine standardmäßig die Ländereinstellung ja_JP.IBM-eucJP verwendet und die Codepage von ja_JP.IBM-eucJP 954 ist, können Sie mit dem folgenden Befehl eine Verknüpfung von /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP zu /usr/lpp/db2_07_01/msg/ja_JP erstellen:

In -s /usr/lpp/db2_07_01/msg/ja_JP /usr/lpp/db2_07_01/msg/ja_JP.IBM-eucJP

Nach der Ausführung dieses Befehls werden alle DB2-Nachrichten in Japanisch angezeigt.

Dateigruppen für Steuerzentrale und Dokumentation

Die Dateigruppen für die Steuerzentrale, die Hilfefunktion der Steuerzentrale und die Dokumentation werden in die folgenden Verzeichnisse auf der Ziel-Workstation kopiert:

- DB2 für AIX:
 - /usr/lpp/db2_07_01/cc/%L
 - /usr/lpp/db2_07_01/java/%L
 - /usr/lpp/db2_07_01/doc/%L
 - /usr/lpp/db2_07_01/qp/\$L
 - /usr/lpp/db2_07_01/spb/%L
- DB2 für HP-UX:
 - /opt/IBMd2/V7.1/cc/%L
 - /opt/IBMd2/V7.1/java/%L

- /opt/IBMd2/V7.1/doc/%L
- DB2 für Linux:
 - /usr/IBMd2/V7.1/cc/%L
 - /usr/IBMd2/V7.1/java/%L
 - /usr/IBMd2/V7.1/doc/%L
- DB2 für Solaris:
 - /opt/IBMd2/V7.1/cc/%L
 - /usr/IBMd2/V7.1/java/%L
 - /opt/IBMd2/V7.1/doc/%L

Dateigruppen der Steuerzentrale liegen in der Unicode-Codepage vor. Die Dateigruppen für die Dokumentation und die Hilfe der Steuerzentrale liegen in einem von Browsern erkannten codierten Zeichensatz vor. Wenn Ihr System einen anderen als den angegebenen Namen für die Ländereinstellung verwendet, können Sie trotzdem noch die übersetzte Version der Steuerzentrale ausführen und die übersetzte Version der Hilfe anzeigen. Dazu müssen Sie Verknüpfungen zu den entsprechenden Sprachverzeichnissen erstellen.

Wenn z. B. Ihre AIX-Maschine standardmäßig die Ländereinstellung ja_JP.IBM-eucJP verwendet, könne Sie mit den folgenden Befehlen Verknüpfungen von /usr/lpp/db2_07_01/cc/ja_JP.IBM-eucJP zu /usr/lpp/db2_07_01/cc/ja_JP und von /usr/lpp/db2_07_01/doc/ja_JP.IBM-eucJP zu /usr/lpp/db2_07_01/doc/ja_JP erstellen:

- `ln -s /usr/lpp/db2_07_01/cc/ja_JP /usr/lpp/db2_07_01/cc/ja_JP.IBM-eucJP`
- `ln -s /usr/lpp/db2_07_01/doc/ja_JP /usr/lpp/db2_07_01/doc/ja_JP.IBM-eucJP`

Nach der Ausführung dieser Befehle werden die Steuerzentrale und der Hilfetext in Japanisch angezeigt.

Anmerkung: Die native Ausführung der Web-Steuerzentrale unter Linux/390 oder NUMA-Q wird nicht unterstützt. Die Web-Steuerzentrale kann jedoch auf einer Client-Workstation zur Verwaltung von Datenbanken auf diesen Plattformen eingesetzt werden.

Einstellung der Länderangaben für den DB2-Verwaltungs-Server

Stellen Sie sicher, dass die Ländereinstellung des DB2-Verwaltungs-Server-Exemplars kompatibel mit der Ländereinstellung des DB2-Exemplars ist. Andernfalls kann das DB2-Exemplar nicht mit dem DB2-Verwaltungs-Server kommunizieren.

Wenn die Umgebungsvariable LANG im Benutzerprofil des DB2-Verwaltungs-Servers nicht gesetzt ist, wird der DB2-Verwaltungs-Server mit den Standardeinstellungen für die Ländereinstellung des Systems gestartet. Wenn die Standardeinstellungen für die Ländereinstellung des Systems nicht definiert sind, wird der DB2-Verwaltungs-Server mit Codepage 819 gestartet. Wenn das DB2-Exemplar eine DBCS-Ländereinstellung verwendet und der DB2-Verwaltungs-Server mit Codepage 819 gestartet wird, kann das Exemplar nicht mit dem DB2-Verwaltungs-Server kommunizieren. Die Ländereinstellung des DB2-Verwaltungs-Servers und die des DB2-Exemplars müssen kompatibel sein. Auf einem Linux-System für vereinfachtes Chinesisch z. B. sollte LANG=zh_CN im Benutzerprofil des DB2-Verwaltungs-Servers gesetzt sein.

DB2 UDB Supports the Baltic Rim Code Page (MS-1257) on Windows Platforms

DB2 UDB supports the Baltic Rim code page, MS-1257, on Windows 32-bit operating systems. This code page is used for Latvian, Lithuanian, and Estonian.

Ableiten der Werte von Codepages

Ändern Sie im Abschnitt „Ableiten der Werte von Codepages“ den ersten Absatz:

Es ist jedoch nicht notwendig, die Registrierungsvariable DB2CODEPAGE zu setzen, da DB2 den geeigneten Wert für die Codepage aus dem Betriebssystem ermittelt.

Er lautet jetzt: In der Regel müssen Sie die Registrierungsvariable DB2CODEPAGE nicht einstellen, da DB2 die Codepage-Informationen automatisch vom Betriebssystem ableitet.

Unterstützung von Landescodes und Codepages

Fügen Sie im Abschnitt „Unterstützung von Landescodes und Codepages“ die folgenden Angaben zur Tabelle hinzu:

Umsetz- tab.	Gruppe	Codierter Zeichens.	Geb.	Landes- code	Länderspez. Angaben	BS	Länderspez. Länderspez.
-----	-----	-----	----	-----	-----	---	-----
943	D-1	IBM-943	JP	81	ja_JP.PCK	Sun	Japan

Zeichensätze

Ersetzen Sie im Abschnitt „Zeichensätze“ unter „Zeichensatz für Kennungen“ die letzten beiden Sätze im ersten Absatz durch folgenden Text:

Verwenden Sie die Sonderzeichen #, @ und \$ vorsichtig in einer NLS-Umgebung, da sie nicht im unveränderlichen Zeichensatz des NLS-Host (EBCDIC) enthalten sind. Zeichen aus dem erweiterten Zeichensatz können je nach verwendeter Codepage ebenfalls verwendet werden. Wenn Sie die Datenbank in einer Umgebung mit mehreren Codepages einsetzen, müssen Sie sicherstellen, dass alle Codepages alle Elemente aus dem zu verwendenden Zeichensatz unterstützen.

Systemverwaltung: Implementierung

Hinzufügen oder Erweitern von DMS-Behältern (Neuer Prozess)

Das Hinzufügen (während oder nach Erstellung des Tabellenbereichs) oder Erweitern von DMS-Behältern (Dateibehälter und Behälter für unformatierte Einheiten) erfolgt jetzt parallel durch die Bereitstellungsfunktionen. Damit die Parallelität dieser Operationen zum Erstellen/Vergrößern von Behältern verbessert wird, kann die Anzahl der auf dem System aktiven Bereitstellungsfunktionen erhöht werden. Die einzigen Prozesse, die nicht parallel erfolgen, sind das Protokollieren dieser Funktionen und - im Fall der Behältererstellung - das Kennzeichnen der Behälter.

Anmerkung: Parallelität von CREATE TABLESPACE / ALTER TABLESPACE (in Hinblick auf das Hinzufügen neuer Behälter in einem vorhandenen Tabellenbereich) nimmt nicht mehr zu, wenn die Anzahl der Bereitstellungsfunktionen der Anzahl der hinzugefügten Behälter entspricht.

Kapitel 1. Verwalten von DB2 mit den Tools für die grafische Benutzerschnittstelle

Entfernen Sie die letzten beiden Sätze des Abschnitts „Alert-Zentrale“.

Entfernen Sie im Abschnitt „Performance Monitor“ den zweiten Listenpunkt der Liste „Definition von Leistungsvariablen“ im Unterabschnitt „Überwachen der Leistung zu einem bestimmten Zeitpunkt“.

Im selben Unterabschnitt sollten die letzten Absätze folgendermaßen umgeschrieben werden:

Zu allen Objekten kann eine Reihe verschiedener Leistungsvariablen überwacht werden. In der Hilfe zu den Leistungsvariablen, die im Menü **Hilfe** eines beliebigen Fensters von Snapshot Monitor verfügbar ist, sind alle Leistungsvariablen beschrieben. Diese Variablen werden in Kategorien eingeteilt. Es gibt folgende Kategorien:

- Exemplar: Agenten, Verbindungen, Sortieren
- Datenbank: Sperre und gegenseitiges Sperren, Pufferpool und E/A, Verbindungen, Sortieren, SQL-Anweisungen
- Tabelle: Tabelle
- Tabellenbereich: Pufferpool und E/A
- Datenbankverbindungen: Pufferpool und E/A, Sperre und gegenseitiges Sperren, Sortieren, SQL-Cursor, SQL-Anweisungen

Genaue Informationen zum Generieren von Momentaufnahmen finden Sie in der Onlinehilfefunktion.

Entfernen Sie in demselben Abschnitt den letzten Satz des Unterabschnitts „Erforderliche Aktion nach der Anzeige eines Objekts in der Alert-Zentrale“.

Kapitel 3. Erstellen einer Datenbank

Erstellen eines Tabellenbereichs

Unformatierte Ein-/Ausgabe bei Linux

Linux hat einen Pool von Knoten mit unformatierten Einheiten, die an eine Blockeinheit gebunden werden müssen, bevor unformatierte Ein-/Ausgabe darauf ausgeführt werden kann. Es gibt einen Controller für unformatierte Einheiten, der als zentrales Repository für die Informationen zur Bindung von unformatierten Einheiten mit Blockeinheiten dient. Die Bindung erfolgt mittels eines Dienstprogramms namens `raw`, das in der Regel vom Linux-Anbieter erhältlich ist.

Für die unformatierte Ein-/Ausgabe auf Linux gelten folgende Voraussetzungen:

- Mindestens eine freie IDE- oder SCSI-Plattenpartition
- Linux-Kernel 2.4.0 oder höher (einige Linux-Distributionen bieten jedoch unformatierte Ein-/Ausgabe bei Kernen der Version 2.2)
- Einen Controller für unformatierte Einheiten namens `/dev/rawctl` oder `/dev/raw`. Wenn dieser nicht verfügbar ist, erstellen Sie eine symbolische Verbindung:

```
# ln -s /dev/your_raw_dev_ctrl /dev/rawctl
```
- Das Dienstprogramm `raw`, das normalerweise in Verbindung mit der Linux-Distribution bereitgestellt wird
- DB2 Version 7.1 FixPak 3 oder höher

Anmerkung: Bei den Distributionen, die derzeit unformatierte Ein-/Ausgabe unterstützen, ist die Benennung von Knoten mit unformatierten Einheiten unterschiedlich:

Distribution	Knoten	Controller
RedHat 6.2	/dev/raw/raw1 bis 255	/dev/rawctl
SuSE 7.0	/dev/raw1 bis 63	/dev/raw

DB2 unterstützt einen der beiden oben aufgeführten Controller für unformatierte Einheiten und die meisten anderen Namen für Knoten mit unformatierten Einheiten. Unformatierte Einheiten werden von DB2 unter Linux/390 nicht unterstützt.

Gehen Sie wie folgt vor, um die unformatierte Ein-/Ausgabe unter Linux zu konfigurieren:

In diesem Beispiel wird als unformatierte Partition `/dev/sda5` verwendet. Sie sollte keine wichtigen Daten enthalten.

Schritt 1. Berechnen Sie die Anzahl von 4096-Byte-Seiten in dieser Partition, und runden Sie dabei gegebenenfalls ab. Beispiel:

```
# fdisk /dev/sda
Command (m for help): p

Disk /dev/sda: 255 heads, 63 sectors, 1106 cylinders
Units = cylinders of 16065 * 512 bytes

   Device Boot   Start    End  Blocks  Id System
 /dev/sda1             1     523   4200997  83  Linux
 /dev/sda2           524   1106   4682947+  5  Extended
 /dev/sda5           524   1106   4682947  83  Linux

Command (m for help): q
#
```

Die Seitenanzahl in /dev/sda5 beträgt:

```
num_pages = floor( ((1106-524+1)*16065*512)/4096 )
num_pages = 11170736
```

Schritt 2. Binden Sie einen nicht verwendeten Knoten für unformatierte Einheiten an diese Partition. Diese Aktion muss bei jedem Neustart des Geräts erfolgen und setzt Root-Zugriff voraus. Zeigen Sie mit `raw -a an`, welche Knoten für unformatierte Einheiten bereits verwendet werden:

```
# raw /dev/raw/raw1 /dev/sda5
/dev/raw/raw1: bound to major 8, minor 5
```

Schritt 3. Setzen Sie die globalen Leseberechtigungen auf dem Controller für unformatierte Einheiten und der Plattenpartition. Setzen Sie die globalen Lese- und Schreibberechtigungen auf der unformatierten Einheit:

```
# chmod a+r /dev/rawctl
# chmod a+r /dev/sdb1
# chmod a+rw /dev/raw/raw1
```

Schritt 4. Erstellen Sie den Tabellenbereich in DB2. Geben Sie dabei die unformatierte Einheit anstelle der Plattenpartition an. Beispiel:

```
CREATE TABLESPACE dms1
MANAGED BY DATABASE
USING (DEVICE '/dev/raw/raw1' 11170736)
```

Tabellenbereiche auf unformatierten Einheiten werden auch für alle anderen von DB2 unterstützten Seitengrößen unterstützt.

Erstellen einer Sequenz

Fügen Sie nach dem Abschnitt „Definieren einer Identitätsspalte in einer neuen Tabelle“ folgenden Abschnitt, „Erstellen einer Sequenz“, hinzu:

Eine *Sequenz* ist ein Datenbankobjekt, das die automatische Generierung von Werten erlaubt. Sequenzen sind hervorragend für die Generierung eindeutiger Schlüsselwerte geeignet. Anwendungen können Sequenzen nutzen, um Probleme durch gemeinsamen Zugriff oder Leistungsprobleme zu vermeiden, die durch die Generierung eines eindeutigen Zählers außerhalb der Datenbank entstehen könnten.

Im Gegensatz zu einem IDENTITY-Spaltenattribut ist eine Sequenz weder mit einer bestimmten Tabellenspalte verbunden noch an eine eindeutige Tabellenspalte gebunden und nur durch diese Tabellenspalte abrufbar.

Eine Sequenz kann erstellt oder geändert werden, so dass sie Werte auf die folgende Art und Weise generiert:

- Monotones Erhöhen oder Vermindern ohne Bindung
- Monotones Erhöhen oder Vermindern mit einer benutzerdefinierten Begrenzung
- Monotones Erhöhen oder Vermindern mit einer benutzerdefinierten Begrenzung und Rücksprung an den Anfang

Im Folgenden sehen Sie ein Beispiel für die Erstellung eines Sequenzobjekts:

```
CREATE SEQUENCE order_seq
START WITH 1
INCREMENT BY 1
NOMAXVALUE
NOCYCLE
CACHE 24
```

In diesem Beispiel heißt die Sequenz `order_seq`. Sie beginnt bei 1 und erhöht sich ohne obere Begrenzung um 1. Es gibt keinen Grund, an den Anfang zurückzuspringen und wieder bei 1 zu beginnen, da keine obere Begrenzung zugeordnet ist. Die Zahl für den Parameter `CACHE` gibt die maximale Anzahl von Sequenzwerten an, die der Datenbankmanager vorgibt und im Speicher hält.

Die generierten Sequenznummern müssen die folgenden Merkmale haben:

- Werte können jeden Exact Numeric-Datentyp mit null Kommastellen haben. Dies sind die Datentypen `SMALLINT`, `BIGINT`, `INTEGER` und `DECIMAL`.
- Aufeinander folgende Werte können sich durch einen beliebigen definierten ganzzahligen Schrittwert unterscheiden. Standardmäßig wird der Inkrementwert 1 verwendet.
- Der Zählerwert ist wiederherstellbar. Er kann aus Protokollen wiederhergestellt werden, wenn eine Fehlerbehebung erforderlich ist.
- Werte können zur Leistungssteigerung zwischengespeichert werden. Durch die Vorgabe und Speicherung von Werten im Cache wird die synchrone E/A im Protokoll verringert, wenn Werte für die Sequenz generiert werden. Bei einem Systemausfall gelten alle nicht festgeschriebenen zwischengespeicherten Werte als verloren und werden nie verwendet. Der für `CACHE` angegebene Wert ist die maximale Anzahl von Sequenzwerten, die verloren gehen könnten.

Wenn eine Datenbank mit mindestens einer Sequenz bis zu einem früheren Zeitpunkt wiederhergestellt wird, könnte dies zur Generierung doppelter Werte für einige Sequenzen führen. Zur Vermeidung möglicher Doppelwerte sollte eine Datenbank mit Sequenzen nicht bis zu einem früheren Zeitpunkt wiederhergestellt werden.

Sequenzen werden nur in einer Datenbank mit einem einzigen Knoten unterstützt.

In Verbindung mit einer Sequenz werden zwei Ausdrücke verwendet.

Der Ausdruck `PREVVAL` gibt den zuletzt generierten Wert für die angegebene Sequenz für eine vorherige Anweisung in der aktuellen Sitzung zurück.

Der Ausdruck `NEXTVAL` gibt den nächsten Wert für die angegebene Sequenz zurück. Eine neue Sequenznummer wird generiert, wenn in einem Ausdruck `NEXTVAL` der Name der Sequenz angegeben ist. Wenn jedoch mehrere Exemplare eines Ausdrucks `NEXTVAL` denselben Sequenznamen innerhalb einer Abfrage angeben, wird der Zähler für die Sequenz für jede Ergebniszeile jeweils nur einmal erhöht.

Dieselbe Sequenznummer kann als eindeutiger Schlüsselwert in zwei separaten Tabellen durch Verweis auf die Sequenznummer innerhalb eines Ausdrucks `NEXTVAL` für die erste Tabelle und eines Ausdrucks `PREVVAL` für alle weiteren Tabellen verwendet werden.

Beispiel:

```
INSERT INTO order (orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1)
```

Die Ausdrücke NEXTVAL oder PREVVAL können an folgenden Stellen verwendet werden:

- Anweisung INSERT, Klausel VALUES
- Anweisung SELECT, SELECT-Liste
- Zuordnungsanweisung SET
- Anweisung UPDATE, Klausel SET
- Anweisung VALUES oder VALUES INTO

Vergleichen von IDENTITY-Spalten und Sequenzen

Fügen Sie den folgenden neuen Abschnitt nach dem neuen Abschnitt „Erstellen einer Sequenz“ hinzu:

IDENTITY-Spalten und Sequenzen weisen sowohl Ähnlichkeiten als auch Unterschiede auf. Die Merkmale der beiden können beim Entwurf Ihrer Datenbank und Ihrer Anwendungen genutzt werden.

Eine IDENTITY-Spalte hat die folgenden Merkmale:

- Eine IDENTITY-Spalte kann nur während der Tabellenerstellung als Teil einer Tabelle definiert werden. Sobald eine Tabelle erstellt ist, können Sie sie nicht ändern, um eine IDENTITY-Spalte hinzuzufügen. (Die Merkmale vorhandener IDENTITY-Spalten können jedoch geändert werden.)
- Eine IDENTITY-Spalte generiert automatisch Werte für eine einzige Tabelle.
- Wenn eine IDENTITY-Spalte als GENERATED ALWAYS definiert ist, werden die verwendeten Werte immer vom Datenbankmanager generiert. Es ist nicht zulässig, dass Anwendungen während der Änderung des Tabelleninhalts ihre eigenen Werte angeben.

Ein Sequenzobjekt hat die folgenden Merkmale:

- Ein Sequenzobjekt ist ein Datenbankobjekt, das nicht mit einer einzigen Tabelle verbunden ist.
- Ein Sequenzobjekt generiert sequenzielle Werte, die in jeder beliebigen SQL-Anweisung verwendet werden können.
- Da ein Sequenzobjekt von jeder Anwendung genutzt werden kann, gibt es zwei Ausdrücke, die den Abruf des nächsten Werts in der angegebenen Sequenz bzw. des zuletzt vor der derzeit ausgeführten Anweisung generierten Werts steuern. Der Ausdruck PREVVAL gibt den zuletzt generierten Wert für die angegebene Sequenz für eine vorherige Anweisung in der aktuellen Sitzung zurück. Der Ausdruck NEXTVAL gibt den nächsten Wert für die angegebene Sequenz zurück. Die Verwendung dieser Ausdrücke erlaubt es, denselben Wert über mehrere SQL-Anweisungen hinweg in mehreren Tabellen zu nutzen.

Anhand dieser nicht umfassenden Liste von Merkmalen der beiden Elemente können Sie ermitteln, welches Element in Abhängigkeit Ihres Datenbankentwurfs und der Anwendungen, die die Datenbank nutzen, einzusetzen ist.

Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation

Fügen Sie im Abschnitt „Erstellen eines Index, einer Indexerweiterung oder einer Indexspezifikation“ den folgenden Hinweis dem Absatz hinzu, der mit folgendem Satz beginnt: „Alle Spalten, die Teil eines Indexschlüssels sind, sind auf 255 Byte begrenzt.“

Anmerkung: Mit Hilfe der Registrierungsvariablen DB2_INDEX_2BYTEVARLEN können Spalten mit mehr als 255 Byte als Teil eines Indexschlüssels angegeben werden.

Kapitel 4. Ändern einer Datenbank

Im Abschnitt „Ändern eines Tabellenbereichs“ müssen folgende Abschnitte hinzugefügt werden:

Hinzufügen eines Behälters zu einem SMS-Tabellenbereich in einer Partition

Sie können einem SMS-Tabellenbereich in einer Partition oder auf einem Knoten, die/der momentan keine Behälter enthält, Behälter hinzufügen.

Der Inhalt des Tabellenbereichs wird über alle Behälter neu verteilt. Während dieser Neuverteilung wird der Zugriff auf den Tabellenbereich nicht eingeschränkt. Wenn mehr als ein Behälter hinzugefügt werden muss, sollten alle Behälter gleichzeitig hinzugefügt werden.

Geben Sie in der Befehlszeile Folgendes ein, um einem SMS-Tabellenbereich einen Behälter hinzuzufügen:

```
ALTER TABLESPACE <name>
  ADD ('<pfad>')
  ON NODE (<partitionsnummer>)
```

Die durch die Nummer angegebene Partition muss ebenso wie sämtliche Partitionen (oder Knoten) im Partitionsbereich in der Knotengruppe vorhanden sein, in welcher der Tabellenbereich definiert ist. Eine Partitionsnummer darf nur in genau einer *ON NODE-Klausel* der Anweisung explizit oder innerhalb eines Bereichs angegeben werden.

Im folgenden Beispiel wird gezeigt, wie auf einem UNIX-basierten Betriebssystem der Partition Nummer 3 der vom Tabellenbereich „plans“ verwendeten Knotengruppe ein neuer Behälter hinzugefügt wird:

```
ALTER TABLESPACE plans
  ADD ('/dev/rhdisk0')
  ON NODE (3)
```

Fügen Sie nach dem Abschnitt „Ändern von Tabellenattributen“ folgende Abschnitte hinzu:

Ändern einer IDENTITY-Spalte

Ändern Sie die Attribute einer vorhandenen IDENTITY-Spalte mit der Anweisung ALTER TABLE. Weitere Informationen zu dieser Anweisung und ihrer Syntax finden Sie im Handbuch *SQL Reference*.

Es gibt verschiedene Möglichkeiten, um eine IDENTITY-Spalte so zu ändern, dass sie einige der Merkmale von Sequenzen erhält.

Einige Tasks gibt es nur bei ALTER TABLE und der IDENTITY-Spalte:

- RESTART setzt die Sequenz für die IDENTITY-Spalte auf den Wert zurück, der bei der ursprünglichen Erstellung der IDENTITY-Spalte implizit oder explizit als Startwert angegeben wurde.
- RESTART WITH <numerische_konstante> setzt die Sequenz für die IDENTITY-Spalte auf den Exact Numeric-Konstantenwert zurück. Die numerische Konstante könnte jeder positive oder negative Wert ohne Dezimalstellen sein, der der IDENTITY-Spalte zugeordnet werden könnte.

Ändern einer Sequenz

Ändern Sie die Attribute einer vorhandenen Sequenz mit der Anweisung ALTER SEQUENCE. Weitere Informationen zu dieser Anweisung und ihrer Syntax finden Sie im Handbuch *SQL Reference*.

Sequenzattribute können folgendermaßen geändert werden:

- Ändern des Inkrements zwischen künftigen Werten
- Definieren neuer Mindest- oder Maximalwerte
- Ändern der Anzahl zwischengespeicherter Sequenznummern
- Ändern des Rücksprungverhaltens der Sequenz
- Ändern der Einstellung, ob die Sequenznummer in der Reihenfolge der Anforderung generiert werden müssen
- Erneutes Starten der Sequenz

Zwei Tasks sind nicht als Teil der Sequenzerstellung zu finden. Dies sind:

- RESTART setzt die Sequenz auf den Wert zurück, der bei der Erstellung der Sequenz implizit oder explizit als Startwert angegeben wurde.
- RESTART WITH <numerische_konstante> setzt die Sequenz auf den Exact Numeric-Konstantenwert zurück. Die numerische Konstante kann jeder positive oder negative Wert ohne Dezimalstellen sein.

Nach dem Neustart einer Sequenz oder nach der Aktivierung des Rücksprungs mit CYCLE können doppelte Sequenznummern generiert werden. Nur künftige Sequenznummern sind von der Anweisung ALTER SEQUENCE betroffen.

Der Datentyp einer Sequenz kann nicht geändert werden. Stattdessen müssen Sie die aktuelle Sequenz löschen und dann eine neue Sequenz mit dem neuen Datentyp erstellen.

Alle zwischengespeicherten Sequenzwerte, die nicht von DB2 verwendet werden, gehen beim Ändern einer Sequenz verloren.

Löschen einer Sequenz

Verwenden Sie zum Löschen einer Sequenz die Anweisung DROP. Weitere Informationen zu dieser Anweisung und ihrer Syntax finden Sie im Handbuch *SQL Reference*.

Eine bestimmte Sequenz kann folgendermaßen gelöscht werden:

```
DROP SEQUENCE sequenzname
```

Dabei ist sequenzname der Name der zu löschenden Sequenz und enthält den impliziten oder expliziten Schemennamen zur exakten Bezeichnung einer vorhandenen Sequenz.

Sequenzen, die vom System für IDENTITY-Spalten erstellt werden, können nicht mit der Anweisung DROP SEQUENCE gelöscht werden.

Sobald eine Sequenz gelöscht wurde, sind auch alle Zugriffsrechte für die Sequenz gelöscht.

Wechseln des Status eines Tabellenbereichs

Mit der Klausel SWITCH ONLINE der Anweisung ALTER TABLESPACE kann ein Tabellenbereich vom Status OFFLINE in den Status ONLINE versetzt werden, wenn der Zugriff auf die diesem Tabellenbereich zugeordneten Behälter möglich ist. Der Tabellenbereich wird in den Status ONLINE versetzt, während die übrige Datenbank aktiv und in Benutzung bleibt.

Alternativ zur Verwendung dieser Klausel können Sie die Verbindung aller Anwendungen zur Datenbank trennen und anschließend erneut herstellen. Dadurch wird der Tabellenbereich vom Status OFFLINE in den Status ONLINE versetzt.

Geben Sie zum Versetzen eines Tabellenbereichs in den Status ONLINE Folgendes in die Befehlszeile ein:

```
ALTER TABLESPACE <name>  
SWITCH ONLINE
```

Modifizieren von Behältern in einem DMS-Tabellenbereich

DMS-Tabellenbereiche werden jetzt parallel erstellt und in der Größe geändert, wodurch die Leistung verbessert wird. Der Grad der Parallelität entspricht der Anzahl von Bereitstellungsfunktionen plus 1.

Kapitel 5. Steuern des Datenbankzugriffs

Fügen Sie nach dem Abschnitt „Zugriffsrechte für Indizes“ folgenden Abschnitt hinzu:

Sequenzzugriffsrechte

Der Ersteller einer Sequenz erhält automatisch das Zugriffsrecht USAGE. Dieses Zugriffsrecht ist erforderlich, um die Ausdrücke NEXTVAL und PREVVAL für die Sequenz nutzen zu können. Damit andere Benutzer berechtigt sind, die Ausdrücke NEXTVAL und PREVVAL zu nutzen, müssen PUBLIC Sequenzzugriffsrechte erteilt werden. Dadurch können Benutzer die Ausdrücke in Verbindung mit der angegebenen Sequenz verwenden.

Fügen Sie nach dem Abschnitt „Überwachen des Zugriffs auf Daten mit Hilfe der Audit-Einrichtung“ folgenden Abschnitt hinzu:

Datenverschlüsselung

Teil Ihres Sicherheitsplan kann die Verschlüsselung Ihrer Daten sein. Dazu können Sie die integrierten Ver- und Entschlüsselungsfunktionen ENCRYPT, DECRYPT_BIN, DECRYPT_CHAR und GETHINT verwenden. Weitere Informationen zu diesen Funktionen und ihrer Syntax finden Sie im Abschnitt „SQL Reference“ dieser Release-Informationen.

Die Funktion ENCRYPT verschlüsselt Daten mit einer kennwortbasierten Verschlüsselungsmethode. Mit diesen Funktionen können Sie auch einen Kennworttipp einbinden. Der Kennworttipp ist in die verschlüsselten Daten eingebettet. Sobald die Daten verschlüsselt sind, können sie nur mit Hilfe des korrekten Kennworts entschlüsselt werden. Entwickler, die diese Funktionen verwenden wollen, sollten die Verwaltung vergessener Kennwörter und nicht nutzbarer Daten einplanen.

Das Ergebnis der Funktion ENCRYPT hat denselben Datentyp wie das erste Argument.

Nur VARCHAR-Daten können verschlüsselt werden.

Für die deklarierte Länge des Ergebnisses gilt Folgendes:

- Sie entspricht der Länge des Datenarguments plus 42, wenn der optionale Tipp-Parameter angegeben ist.
- Sie entspricht der Länge des Datenarguments plus 10, wenn der optionale Tipp-Parameter nicht angegeben ist.

Die Funktionen DECRYPT_BIN und DECRYPT_CHAR entschlüsseln Daten mit kennwortbasierter Entschlüsselung.

Das Ergebnis der Funktionen DECRYPT_BIN und DECRYPT_CHAR hat denselben Datentyp wie das erste Argument.

Die deklarierte Länge des Ergebnisses entspricht der Länge der ursprünglichen Daten.

Die Funktion GETHINT gibt einen eingebundenen Kennworttipp zurück. Ein Kennworttipp ist ein Ausdruck, der den Dateneignern hilft, sich an Kennwörter zu erinnern. Beispielsweise kann das Wort „Ozean“ als Tipp verwendet werden, um sich an das Kennwort „Pazifik“ zu erinnern.

Das Kennwort, das zur Verschlüsselung der Daten verwendet wird, wird auf zwei Arten festgelegt:

- Kennwortargument. Das Kennwort ist eine Zeichenfolge, die explizit beim Aufruf der Funktion ENCRYPT übergeben wird. Die Daten werden mit dem angegebenen Kennwort ver- und entschlüsselt.
- Sonderregisterkennwort. Die Anweisung SET ENCRYPTION PASSWORD verschlüsselt den Kennwortwert und sendet das verschlüsselte Kennwort an den Datenbankmanager, der es in einem Sonderregister speichert. Die Funktionen ENCRYPT, DECRYPT_BIN und DECRYPT_CHAR, die ohne einen Kennwortparameter aufgerufen werden, verwenden den Wert im Sonderregister ENCRYPTION PASSWORD.

Der Anfangs- oder Standardwert für das Sonderregister ist eine leere Zeichenfolge.

Kennwörter dürfen 6 bis 127 Zeichen lang sein. Kennworttipps dürfen 0 bis 32 Zeichen lang sein.

Wenn das Sonderregister ENCRYPTION PASSWORD vom Client gesetzt ist, wird das Kennwort auf dem Client verschlüsselt, zum Datenbankserver gesendet und dann entschlüsselt. Damit das Kennwort nicht lesbar bleibt, wird es auch auf dem Datenbankserver wieder verschlüsselt. Die Funktionen DECRYPT_BIN und DECRYPT_CHAR müssen das Sonderregister vor der Verwendung entschlüsseln. Der Wert in ENCRYPTION PASSWORD bleibt ebenfalls nicht lesbar. Gateway-Sicherheit wird nicht unterstützt.

Kapitel 8. Wiederherstellen einer Datenbank

Verwendung der zurückgestellten Ein-/Ausgabe

In Kapitel 8, „Wiederherstellen einer Datenbank“, muss der folgende neue Abschnitt zur Verwendung der Funktion für zurückgestellte Ein-/Ausgabe hinzugefügt werden:

Anmerkung: Die unten stehenden Informationen zum Dienstprogramm "db2inidb" haben Vorrang vor den Informationen im Handbuch *Neue Funktionen* der Version 7.2.

db2inidb ist ein neues Tool, das mit DB2 ausgeliefert wird und Wiederherstellung nach einem Systemabsturz ausführen, eine Datenbank in den Status „aktualisierende Wiederherstellung anstehend“ versetzen kann.

Zurückgestellte Ein-/Ausgabe unterstützt die fortlaufende Systemverfügbarkeit durch eine volle Implementierung der Onlineteilung von Spiegeln, das heißt das Teilen eines Spiegels, ohne die Datenbank herunterfahren zu müssen. Wenn es einem Kunden nicht möglich ist, Offline- oder Onlinesicherungen in einer großen Datenbank auszuführen, können Sicherungen oder Systemkopien mit Hilfe von zurückgestellter Ein-/Ausgabe und Spiegelteilung über ein Spiegelimage ausgeführt werden.

Je nach Art der Spiegelung der Speichereinheiten ist die Verwendung von db2inidb unterschiedlich. Im Folgenden wird davon ausgegangen, dass die gesamte Datenbank im Speichersystem konsistent gespiegelt wird.

In einer Umgebung mit mehreren Knoten muss das Tool **db2inidb** auf jeder Partition ausgeführt werden, bevor das geteilte Image von einer der Partitionen aus verwendet werden kann. Das Tool **db2inidb** kann gleichzeitig auf allen Partitionen ausgeführt werden.

1. Erstellen einer Klondatenbank

Ziel ist die Erstellung eines Klons der Primärdatenbank für den Lesezugriff. Die folgende Prozedur beschreibt, wie eine Klondatenbank erstellt werden kann:

- a. Stellen Sie die Ein-/Ausgabe auf dem primären System mit dem folgenden Befehl zurück:

```
db2 set write suspend for database
```

- b. Trennen Sie den Spiegel mit Hilfe des Betriebssystembefehls von der primären Datenbank.
- c. Nehmen Sie die Ein-/Ausgabe auf dem primären System mit dem folgenden Befehl wieder auf:

```
db2 set write resume for database
```

Nach Ausführung des Befehls sollte sich die Datenbank auf dem primären System wieder in einem normalen Status befinden.

- d. Stellen Sie von einer anderen Maschine aus eine Verbindung zur gespiegelten Datenbank her.
- e. Starten Sie das Datenbankexemplar mit dem folgenden Befehl:

```
db2start
```

- f. Starten Sie die DB2-Wiederherstellung nach Systemabsturz mit dem folgenden Befehl:

```
db2inidb datenbankname AS SNAPSHOT
```

Anmerkung: Dieser Befehl nimmt die durch Transaktionen, die zum Teilungszeitpunkt aktiv waren, vorgenommenen Änderungen zurück.

Sie können diesen Prozess auch für eine Offlinesicherung verwenden. Wenn die Sicherung jedoch auf dem primären System wiederhergestellt wird, kann sie nicht für eine aktualisierende Wiederherstellung verwendet werden, da die Protokollkette nicht übereinstimmt.

2. Verwendung des geteilten Spiegels als Bereitschaftsdatenbank

Während die gespiegelte Datenbank (Bereitschaftsdatenbank) eine fortlaufende aktualisierende Wiederherstellung über die Protokolle durchführt, werden neue Protokolle, die die primäre Datenbank erstellt, ständig aus dem primären System abgerufen. Die folgende Prozedur zeigt, wie der geteilte Spiegel als Bereitschaftsdatenbank verwendet werden kann:

- a. Stellen Sie die Ein-/Ausgabe auf der primären Datenbank zurück.
- b. Spalten Sie den Spiegel vom primären System ab.
- c. Nehmen Sie die Ein-/Ausgabe auf der primären Datenbank wieder auf, so dass der Normalbetrieb der Datenbank fortgesetzt wird.
- d. Ordnen Sie die gespiegelte Datenbank einem anderen Exemplar zu.
- e. Versetzen Sie den Spiegel in den Status „aktualisierende Wiederherstellung anstehend“, und führen Sie eine aktualisierende Wiederherstellung des Spiegels aus. Führen Sie das Tool `db2inidb` aus (`db2inidb <db_alias>` im Bereitschaftsmodus), um den Status „Schreiben zurückstellen“ aufzuheben, und versetzen Sie die gespiegelte Datenbank in den Status „aktualisierende Wiederherstellung anstehend“.
- f. Kopieren Sie Protokolle mit Hilfe eines Benutzer-Exit-Programms, das Protokolldateien aus dem primären System abrufen, um sicherzustellen, dass die neuesten Protokolle für diese gespiegelte Datenbank zur Verfügung stehen.
- g. Führen Sie eine aktualisierende Wiederherstellung der Datenbank bis zum Ende der Protokolle aus.
- h. Gehen Sie zurück zu Schritt f, und wiederholen Sie diesen Prozess, bis die primäre Datenbank inaktiv ist.

3. Verwendung des geteilten Spiegels als Sicherungsimage

Die folgende Prozedur beschreibt, wie das gespiegelte System als Sicherungsimage zum Wiederherstellen über das primäre System verwendet wird:

- a. Kopieren Sie die gespiegelten Daten und Protokolle mit Betriebssystembefehlen über das primäre System.
- b. Starten Sie das Datenbankexemplar mit dem folgenden Befehl:
`db2start`
- c. Führen Sie den folgenden Befehl aus, um die gespiegelte Datenbank in den Status „aktualisierende Wiederherstellung anstehend“ zu versetzen und heben Sie den Status „Schreiben zurückstellen“ auf:
`db2inidb
datenbankaliasname AS MIRROR`
- d. Führen Sie eine aktualisierende Wiederherstellung der Datenbank bis zum Ende der Protokolle aus.

Teilsicherung und -wiederherstellung

In Kapitel 8, „Wiederherstellen einer Datenbank“, wird folgender neuer Abschnitt zu Teilsicherungen und -wiederherstellungen eingefügt:

Da die Größe von Datenbanken und insbesondere von Warehouses weiter in den Terabyte- und Petabyte-Bereich steigt, wächst auch die erforderliche Zeit für die Sicherung und Wiederherstellung dieser Datenbanken beträchtlich. Vollsicherungen von Datenbanken und Tabellenbereichen sind nicht immer die beste Vorgehensweise bei der Handhabung großer Datenbanken, da die Speicheranforderungen für mehrere Kopien solcher Datenbanken enorm sind. Berücksichtigen Sie folgende Punkte:

- Wenn sich nur ein kleiner Teil der Daten im Warehouse ändert, sollte es nicht notwendig sein, die gesamte Datenbank zu sichern.
- Das Anhängen von Tabellenbereichen an vorhandene Datenbanken und die anschließende Sicherung nur der Tabellenbereiche ist riskant, da möglicherweise Daten außerhalb der gesicherten Tabellenbereiche geändert wurden.

DB2 unterstützt jetzt die Teilsicherung und -wiederherstellung (jedoch nicht von Langfelddaten oder LOB-Daten). Eine *Teilsicherung* ist ein Sicherungsimagem, das nur die Seiten enthält, die seit der letzten Sicherung aktualisiert wurden. Zusätzlich zu den aktualisierten Daten und Indexseiten enthält jedes Teilsicherungsimagem auch alle der ursprünglichen Datenbank-Metadaten (z. B. Datenbankkonfiguration, Tabellenbereichsdefinitionen, Datenbankprotokoll usw.), die normalerweise in Gesamtsicherungsimagem gespeichert werden.

Es werden zwei Arten von Teilsicherung unterstützt:

- *Inkrementell*. Ein inkrementelles Sicherungsimagem ist eine Kopie aller Datenbankdaten, die seit der letzten erfolgreichen Gesamtsicherung geändert wurden. Dies wird auch als kumulatives Sicherungsimagem bezeichnet, da Teilsicherungen, die im Lauf der Zeit erstellt werden, jeweils den Inhalt des vorherigen Teilsicherungsimagem enthalten. Der Vorgänger eines Teilsicherungsimagem ist immer die letzte erfolgreiche Gesamtsicherung desselben Objekts.
- *Delta*. Ein Delta- oder inkrementelles Delta-Sicherungsimagem ist eine Kopie aller Datenbankdaten, die seit der letzten erfolgreichen Sicherung (Gesamt-, Inkrementell- oder Deltasicherung) des entsprechenden Tabellenbereichs geändert wurden. Dies wird auch als Differenzial- oder nicht kumulatives Sicherungsimagem bezeichnet. Der Vorgänger eines Delta-Sicherungsimagem ist die letzte erfolgreiche Sicherung mit einer Kopie aller Tabellenbereiche im Delta-Sicherungsimagem.

Der Hauptunterschied zwischen inkrementellen und Delta-Sicherungsimagem ist ihr Verhalten, wenn aufeinanderfolgende Sicherungen eines Objekts erfolgen, das im Lauf der Zeit verändert wird. Jedes inkrementelle Image enthält den gesamten Inhalt des vorherigen inkrementellen Image sowie alle Daten, die seit der letzten Sicherung geändert wurden oder neu hinzugekommen sind. Delta-Sicherungsimagem enthalten nur die Seiten, die seit der Erstellung des letzten Image geändert wurden.

Kombinationen aus Datenbank- und Tabellenbereichs-Teilsicherungen sind sowohl online als auch offline zulässig. Planen Sie Ihre Sicherungsstrategie sorgfältig, da die Kombination von Datenbank- und Tabellenbereichs-Teilsicherungen impliziert, dass der Vorgänger einer Datenbanksicherung (oder einer Tabellenbereichs-sicherung mehrerer Tabellenbereiche) nicht notwendigerweise ein einzelnes Image

ist, sondern eine eindeutige Menge vorheriger Datenbank- und Tabellenbereichsicherungen sein könnte, die zu unterschiedlichen Zeiten durchgeführt worden sind.

Zur Wiederherstellung der Datenbank oder des Tabellenbereichs auf einen konsistenten Zustand muss der Wiederherstellungsprozess mit einem konsistenten Image des gesamten wiederherzustellenden Objekts (Datenbank oder Tabellenbereich) beginnen und dann alle passenden Teilsicherungsimages in der unten beschriebenen Reihenfolge anwenden (siehe Abschnitt zur Wiederherstellungsmethode).

DB2 unterstützt jetzt einen neuen Datenbankkonfigurationsparameter, TRACKMOD, um die Verfolgung von Datenbankaktualisierungen zu ermöglichen. Dieser Parameter kann zwei Werte annehmen:

- NO. Teilsicherung ist bei dieser Konfiguration nicht zulässig. Aktualisierungen von Datenbankseiten werden weder verfolgt noch aufgezeichnet.
- YES. Teilsicherung ist bei dieser Konfiguration zulässig. Wenn die Aktualisierungsverfolgung aktiviert ist, wird die Änderung bei der ersten erfolgreichen Verbindung zu einer Datenbank im Exemplar wirksam. Eine Datenbankgesamticherung ist notwendig, bevor eine Teilsicherung erfolgen kann.

Die Standardeinstellung von TRACKMOD für vorhandene Datenbanken lautet NO; bei neuen Datenbanken lautet sie YES.

Die Unterteilung der Verfolgung erfolgt auf der Ebene der Tabellenbereiche für SMS- und DMS-Tabellenbereiche.

Die Verfolgung von Aktualisierungen an der Datenbank kann eine wenn auch minimale Auswirkung auf die Laufzeitleistung von Transaktionen haben, die Daten aktualisieren oder einfügen.

Wiederherstellen von Teilsicherungsimages

Eine Wiederherstellungsoperation von Teilsicherungsimages umfasst immer die folgenden Schritte:

1. Angeben des Zielimage. Der Datenbankadministrator muss zuerst festlegen, welches Endimage wiedergestellt werden soll, und dann eine Teiltrückschreibung über das DB2-Dienstprogramm **restore** anfordern. Dieses Image wird als Zielimage der Teiltrückschreibung bezeichnet, da es das letzte wiederherzustellende Image ist. Ein Teiltrückschreibungsbefehl für dieses Image kann die Erstellung einer neuen Datenbank mit der Konfiguration und den Tabellenbereichsdefinitionen dieses Zielimage einleiten. Das Teiltrückschreibungsziel wird mit dem Parameter TAKEN AT im Befehl RESTORE DATABASE angegeben.
2. Wiederherstellen des neuesten Gesamtsicherungsimage der Datenbank bzw. des Tabellenbereichs, um eine Ausgangsbasis zu erhalten, auf die nachfolgende Teilsicherungsimages angewendet werden können.
3. Wiederherstellen der erforderlichen Gesamt- oder Teilsicherungsimages von Tabellenbereichen in der Reihenfolge ihrer Erstellung aufbauend auf dem in Schritt 2 wiederhergestellten Grundimage.
4. Wiederholen von Schritt 3, bis das Zielimage aus Schritt 1 ein zweites Mal gelesen wird. In einer kompletten Teiltrückschreibung wird immer zweimal auf das Zielimage zugegriffen. Beim ersten Zugriff werden nur die Initialisierungsdaten vom Image, jedoch noch keine Benutzerdaten gelesen. Das vollständige Image wird dann während des zweiten Zugriffs gelesen und verarbeitet.

Auf das Zielimage der Teiltrückschreibung muss zweimal zugegriffen werden, um sicherzustellen, dass die Datenbank anfangs mit dem korrekten Protokoll,

mit der korrekten Datenbankkonfiguration und mit den korrekten Tabellenbereichsdefinitionen für die Datenbank konfiguriert wird, die während der Wiederherstellung erzeugt wird. Wenn ein Tabellenbereich seit der anfänglichen Gesamtsicherung der Datenbank gelöscht wurde, werden die Tabellenbereichsdaten für dieses Image aus den Sicherungsimagen gelesen, jedoch während der Teilrückschreibung ignoriert.

Beispiel:

1. db2 restore database sample incremental taken at <ts>

Dabei gilt Folgendes:

<ts> zeigt auf das letzte wiederherzustellende Teilsicherungsimage.

2. db2 restore database sample incremental taken at <ts1>

Dabei gilt Folgendes:

<ts1> zeigt auf das ursprüngliche Gesamtsicherungsimage der Datenbank (bzw. des Tabellenbereichs).

3. db2 restore database sample incremental taken at <tsX>

Dabei gilt Folgendes:

<tsX> zeigt auf jedes Teilsicherungsimage in der Erstellungsfolge.

4. Wiederholen Sie Schritt 3, wodurch jedes Teilsicherungsimage bis einschließlich Image <ts> wiederhergestellt wird.

Wenn eine Datenbankwiederherstellung versucht wird und Teilsicherungsimages von Tabellenbereichen erstellt wurden, müssen die Tabellenbereichsimages in der Reihenfolge ihrer Sicherungszeitmarken wiederhergestellt werden.

Parallele Wiederherstellung

DB2 verwendet jetzt mehrere Agenten für die Wiederherstellung nach einem Systemabsturz und die aktualisierende Datenbankwiederherstellung. Sie können jetzt bei diesen Operationen eine bessere Leistung erwarten, insbesondere bei symmetrischen Mehrprozessor-Maschinen (SMP). Durch den Einsatz mehrerer Agenten bei der Datenbankwiederherstellung werden die zusätzlichen CPUs genutzt, die auf SMP-Maschinen verfügbar sind.

Der neue Agententyp, der in dieser Erweiterung eingeführt wurde, heißt „db2agnsc“. DB2 wählt die Anzahl der für die Datenbankwiederherstellung zu verwendenden Agenten anhand der Anzahl der CPUs der Maschine aus. Bei SMP-Maschinen ist die Anzahl von Agenten gleich der Anzahl der CPUs + 1. Bei einer Maschine mit einer CPU werden drei Agenten für das effizientere Lesen von Protokollen, das Verarbeiten von Protokollsätzen und das Bereitstellen von Datenseiten verwendet.

DB2 verteilt Protokollsätze so an diese Agenten, dass sie gegebenenfalls gleichzeitig erneut angewendet werden können. Die Verarbeitung von Protokollsätzen wird parallel auf Seitenebene geschrieben (Protokollsätze auf derselben Datenseite werden vom selben Agenten verarbeitet). Dadurch wird die Leistung verbessert, selbst wenn alle Aktionen an einer Tabelle erfolgen.

Sicherung auf benannte Pipes

Die Datenbanksicherung auf lokale benannte Pipes auf UNIX-basierten Systemen und die Datenbankwiederherstellung davon werden jetzt unterstützt. Eingabe- und Ausgabeprogramm der benannten Pipe müssen sich auf derselben Maschine befinden. Die Pipe muss vorhanden sein und sich in einem lokalen Dateisystem befinden. Da die benannte Pipe als lokale Einheit behandelt wird, muss nicht angegeben werden, dass das Ziel eine benannte Pipe ist. Im Folgenden sehen Sie ein Beispiel für AIX:

1. Erstellen Sie eine benannte Pipe:
`mkfifo /u/dbuser/mypipe`
2. Verwenden Sie diese Pipe als Ziel für eine Datenbanksicherung:
`db2 backup db sample to /u/dbuser/mypipe`
3. Stellen Sie die Datenbank wieder her:
`db2 restore db sample into mynewdb from /u/dbuser/mypipe`

Sicherung von einem geteilten Image

DB2 unterstützt jetzt Offline-Gesamtdatenbanksicherungen auf der geteilten gespiegelten Kopie einer Datenbank. Onlinesicherung wird nicht unterstützt und ist auch nicht notwendig, da die Datenbank, die sich im Status „aktualisierende Wiederherstellung anstehend“ befindet, nicht verfügbar ist. Wenn ein geteiltes gespiegeltes Sicherungsimage wiederhergestellt wird, muss es aktualisierend wiederhergestellt werden, da während der Teilung aktive Transaktionen aufgetreten sein können.

Anmerkung: Bei DB2 Version 7.1 FixPak 3 und DB2 Version 7.2 ist diese Unterstützung auf Datenbanken beschränkt, die nur DMS-Tabellenbereiche enthalten. Wenn versucht wird, eine Datenbank nach einer Teilung zu sichern und die Datenbank SMS-Tabellenbereiche enthält, schlägt die Sicherung fehl.

Sobald eine Datenbank geteilt wurde, muss das Dienstprogramm **db2inidb** verwendet werden, um eine der folgenden Optionen anzugeben:

- Snapshot. Dies leitet eine Wiederherstellung nach einem Systemabsturz ein, so dass die Datenbank wieder konsistent wird. Eine neue Protokollkette beginnt und die Datenbank kann Protokolle der ursprünglichen Datenbank nicht mehr aktualisierend wiederherstellen. Die Datenbank ist für alle Operationen einschließlich Sicherung verfügbar.
- Standby. Dies setzt die Datenbank in den Status „aktualisierende Wiederherstellung anstehend“. Es wird *keine* Wiederherstellung nach einem Systemabsturz durchgeführt, und die Datenbank bleibt inkonsistent.
- Mirror. Dies erzeugt eine gespiegelte Kopie der Datenbank, die die ursprüngliche Datenbank ersetzt. Die Datenbank wird in den Status „aktualisierende Wiederherstellung anstehend“ gesetzt, und der Status WRITE SUSPEND wird inaktiviert. Es wird *keine* Wiederherstellung nach einem Systemabsturz durchgeführt und die Datenbank bleibt inkonsistent.

Im Folgenden sehen Sie einige Nutzungsszenarios:

- Erstellen eines Datenbankklons.

Ziel hier ist, einen schreibgeschützten Klon der primären Datenbank zu haben, der z. B. zum Erstellen von Berichten verwendet werden kann. Führen Sie dazu folgende Schritte aus:

1. Stellen Sie die Ein-/Ausgabe auf dem primären System zurück:
`db2 set write suspend for database`

2. Teilen Sie den Spiegel. Trennen Sie den Spiegel mit Hilfe von Betriebssystembefehlen von der primären Datenbank.
3. Nehmen Sie die Ein-/Ausgabe auf dem primären System wieder auf:

```
db2 set write resume for database
```

Die Datenbank auf dem primären System sollte sich wieder in einem normalen Status befinden.

4. Hängen Sie die geteilten Spiegel der Datenbank an einen anderen Host an.
5. Starten Sie das Exemplar:

```
db2start
```
6. Starten Sie die Wiederherstellung nach einem Systemabsturz von DB2:

```
db2inidb <db_name> as snapshot
```

Sie können diesen Prozess auch für eine Offlinesicherung verwenden. Wenn die Sicherung jedoch auf dem primären System wiederhergestellt wird, kann sie nicht für eine aktualisierende Wiederherstellung verwendet werden, da die Protokollkette nicht übereinstimmt.

- Verwendung des geteilten Spiegels als Bereitschaftsdatenbank

Während die gespiegelte Datenbank (Bereitschaftsdatenbank) eine fortlaufende aktualisierende Wiederherstellung über die Protokolle durchführt, werden neue Protokolle, die die primäre Datenbank erstellt, ständig aus dem primären System abgerufen. Gehen Sie wie folgt vor, um den geteilten Spiegel als Bereitschaftsdatenbank zu verwenden:

1. Stellen Sie die Ein-/Ausgabe auf dem primären System zurück:

```
db2 set write suspend for database
```
2. Teilen Sie den Spiegel. Trennen Sie den Spiegel mit Hilfe von Betriebssystembefehlen von der primären Datenbank.
3. Nehmen Sie die Ein-/Ausgabe auf dem primären System wieder auf:

```
db2 set write resume for database
```

Die Datenbank auf dem primären System sollte sich wieder in einem normalen Status befinden.

4. Hängen Sie die geteilten Spiegel der Datenbank an einen anderen Host an.
5. Heben Sie den Status „Schreiben zurückstellen“ auf, und setzen Sie die gespiegelte Datenbank in den Status „aktualisierende Wiederherstellung anstehend“.

```
db2inidb <db_name> as standby
```
6. Kopieren Sie Protokolle. Richten Sie ein Benutzer-Exit-Programm so ein, dass es Protokolldateien aus dem Archiv des primären Systems abrufen, so dass die neuesten Protokolle für diese gespiegelte Datenbank zur Verfügung stehen.
7. Führen Sie eine aktualisierende Wiederherstellung des Spiegels bis zum Ende der Protokolle aus.

```
db2 rollforward db <db_name> to end of logs
```
8. Wiederholen Sie den Prozess ab Schritt 6, bis die primäre Datenbank inaktiv ist.

- Verwendung des geteilten Spiegels zur Wiederherstellung des primären Systems
 Die folgende Prozedur beschreibt, wie das gespiegelte System als Sicherungsbild zum Wiederherstellen des primären Systems verwendet wird:

1. Überschreiben durch Kopieren. Kopieren Sie die gespiegelten Daten und Protokolle mit Betriebssystembefehlen über das primäre System.

2. Starten Sie das Exemplar:

```
db2start
```

3. Setzen Sie den wiederhergestellten Spiegel in den Status „aktualisierende Wiederherstellung anstehend“, und führen Sie eine aktualisierende Wiederherstellung des Spiegels bis zum Ende der Protokolle aus.

```
db2inidb <dbname> as mirror
```

- Erstellen einer Sicherung ohne Wiederherstellung nach einem Systemabsturz
Wenn Sie eine Offlinesicherung ohne Wiederherstellung nach einem Systemabsturz durchführen, können Sie dieses Sicherungsimago über das primäre System wiederherstellen. Führen Sie dazu folgende Schritte aus:

1. Stellen Sie die Ein-/Ausgabe auf dem primären System zurück:

```
db2 set write suspend for database
```

2. Teilen Sie den Spiegel. Trennen Sie den Spiegel mit Hilfe von Betriebssystembefehlen von der primären Datenbank.

3. Nehmen Sie die Ein-/Ausgabe auf dem primären System wieder auf:

```
db2 set write resume for database
```

Die Datenbank auf dem primären System sollte sich wieder in einem normalen Status befinden.

4. Hängen Sie die geteilten Spiegel der Datenbank an einen anderen Host an.

5. Starten Sie das Exemplar:

```
db2start
```

6. Setzen Sie die gespiegelte Datenbank in den Status „aktualisierende Wiederherstellung anstehend“:

```
db2inidb <db_name> as standby
```

7. Rufen Sie eine Datenbanksicherung auf:

```
db2 backup database <db_name>
```

Dies führt zu einer impliziten Datenbankverbindung, leitet jedoch keine Wiederherstellung nach einem Systemabsturz von DB2 ein.

Protokollarchivierung nach Bedarf

DB2 unterstützt jetzt den Abschluss (und bei aktivierter Benutzer-Exit-Option die Archivierung) des aktiven Protokolls für eine wiederherstellbare Datenbank zu jedem beliebigen Zeitpunkt. Dadurch können Sie einen vollständigen Satz von Protokolldateien bis zu einem bestimmten Zeitpunkt sammeln und dann mit Hilfe dieser Protokolldateien eine Bereitschaftsdatenbank aktualisieren.

Anmerkung: Die Protokollierung nach Bedarf garantiert nicht, dass die Protokolldateien sofort archiviert werden. Sie schneidet die Protokolldatei ab und setzt eine Archivierungsanforderung ab, unterliegt jedoch noch den Verzögerungen im Zusammenhang mit dem Benutzer-Exit-Programm.

Sie können die Protokollierung nach Bedarf mit dem neuen DB2-Befehl ARCHIVE LOG oder über die neue API **db2ArchiveLog** einleiten.

Protokollspiegelung

In Kapitel 8, „Wiederherstellen einer Datenbank“, wird folgender Abschnitt zur Verwendung der zurückgestellten E/A-Funktion hinzugefügt:

DB2 unterstützt jetzt Protokollspiegelung auf Datenbankebene. Die Spiegelung von Protokolldateien hilft, die Datenbank vor folgenden Risiken zu schützen:

- Versehentliches Löschen einer aktiven Protokolldatei
- Datenverlust durch Hardwarefehler

Wenn Sie vermeiden wollen, dass Ihre aktiven Protokolle beschädigt werden (z. B. durch einen Plattenabsturz), sollten Sie die Verwendung einer neuen DB2-Registrierungsvariablen, `DB2_NEWLOGPATH2`, in Betracht ziehen, mit der Sie einen sekundären Pfad für die Datenbank zur Verwaltung von Kopien des aktiven Protokolls angeben können, so dass die Datenträger gespiegelt werden, auf denen die Protokolle gespeichert sind.

Die Registrierungsvariable `DB2_NEWLOGPATH2` erlaubt es der Datenbank, eine identische Zweitkopie von Protokolldateien in einen anderen Pfad zu schreiben. Es empfiehlt sich, den sekundären Protokollpfad auf eine physikalisch getrennte Platte zu legen (vorzugsweise auf eine Platte, die an einen anderen Plattencontroller angeschlossen ist). Auf diese Weise kann der Plattencontroller nicht zu einer kritischen Komponente werden, die zu einem gesamten Systemausfall führen kann.

Anmerkung: Da Windows NT und OS/2 das „Anhängen“ einer Einheit unter einem beliebigen Pfadnamen nicht unterstützen, ist es (auf diesen Plattformen) nicht möglich, einen sekundären Pfad auf einer separaten Einheit anzugeben.

`DB2_NEWLOGPATH2` kann aktiviert (auf 1 gesetzt) oder inaktiviert (auf 0 gesetzt) sein. Der Standardwert ist Null. Wenn diese Variable auf 1 gesetzt ist, ist der Name des sekundären Pfads der aktuelle Wert der Variablen `LOGPATH` verkettet mit dem Zeichen 2. Wenn z. B. in einer SMP-Umgebung `LOGPATH` `/u/dbuser/sqllogdir/logpath` lautet, heißt der sekundäre Protokollpfad `/u/dbuser/sqllogdir/logpath2`. Wenn `LOGPATH` in einer MPP-Umgebung `/u/dbuser/sqllogdir/logpath` lautet, hängt DB2 den Knotenanzeiger an den Pfad an und verwendet `/u/dbuser/sqllogdir/logpath/NODE0000` als primären Protokollpfad. In diesem Fall heißt der sekundäre Protokollpfad `/u/dbuser/sqllogdir/logpath2/NODE0000`.

Wenn `DB2_NEWLOGPATH2` zum ersten Mal aktiviert wird, kommt es erst zur Verwendung, wenn die aktuelle Protokolldatei beim nächsten Datenbankstart abgeschlossen wird. Dies ist ähnlich der derzeitigen Verwendung von `NEWLOGPATH`. Wenn beim Schreiben im primären oder sekundären Protokollpfad ein Fehler auftritt, kennzeichnet die Datenbank den Pfad mit dem Fehler als fehlerhaft, schreibt eine Nachricht in die Datei `db2diag.log` und schreibt nachfolgende Protokollsätze nur noch in den verbleibenden fehlerfreien Protokollpfad. DB2 versucht erst wieder, den fehlerhaften Pfad zu verwenden, nachdem Höchstgröße der aktuellen Protokolldatei erreicht ist.

Wenn DB2 die nächste Protokolldatei öffnen muss, prüft es, ob dieser Pfad gültig ist, und benutzt diesen, wenn dies der Fall ist. Ist er nicht gültig, versucht DB2 erst wieder, den Pfad zu verwenden, nachdem zum ersten Mal auf die nächste Protokolldatei zugegriffen wurde. Es wird nicht versucht, die Protokollpfade zu synchronisieren. DB2 speichert jedoch Informationen zu Zugriffsfehlern, so dass beim Archivieren der Protokolldateien die korrekten Pfade verwendet werden. Wenn ein Fehler beim Schreiben im verbleibenden fehlerfreien Pfad auftritt, wird die Datenbank abnormal beendet.

Unterstützung von plattformübergreifendem Sichern und Wiederherstellen unter Sun Solaris und HP

Jetzt wird die plattformübergreifende Sicherung und Wiederherstellung zwischen Sun Solaris und HP unterstützt. Wenn Sie das Sicherungsimage zwischen Systemen übertragen, muss dies im binären Modus geschehen. Auf dem Zielsystem muss die Datenbank mit derselben Codepage und demselben Gebiet erstellt werden wie das System, auf dem die ursprüngliche Datenbank erstellt wurde.

Überlegungen zum DB2 Data Links Manager/Überlegungen zum Dienstprogramm BACKUP

Ersetzen Sie den zweiten Absatz in diesem Abschnitt durch folgenden Text:

Wenn Dateien verbunden werden, werden sie von den Data Links-Servern zum asynchronen Kopieren auf einen Archivierungs-Server wie ADSTAR (ADSTAR Distributed Storage Manager) oder auf Platte terminiert. Wenn das Sicherungsprogramm ausgeführt wird, stellt DB2 sicher, dass alle Dateien, die zum Kopieren terminiert sind, kopiert werden. Zu Beginn des Sicherungsvorgangs stellt DB2 einen Kontakt zu allen Data Links-Servern her, die in der DB2-Konfigurationsdatei angegeben sind. Wenn ein Data Links-Server eine oder mehrere verbundene Dateien hat und nicht aktiv ist bzw. während der Sicherung gestoppt wird, enthält die Sicherung nicht die vollständigen DATALINK-Informationen. Die Sicherung wird erfolgreich abgeschlossen. Bevor der Data Links-Server als erneut für die Datenbank verfügbar gekennzeichnet werden kann, muss das Sicherungsverfahren für alle anstehenden Sicherungen erfolgreich abgeschlossen sein. Wenn eine Sicherung eingeleitet wird, während bereits die doppelte Anzahl des Werts "num_db_backups" (siehe unten) an anstehenden Sicherungen auf den Abschluss auf dem Data Links-Server wartet, schlägt die Operation fehl. Dieser Data Links-Server muss erneut gestartet werden, und die anstehenden Sicherungen müssen abgeschlossen werden, bevor weitere Sicherungen zulässig sind.

Überlegungen zum DB2 Data Links Manager/Überlegungen zu den Dienstprogrammen RESTORE und ROLLFORWARD

Ersetzen Sie Absätze, die folgendermaßen beginnen:

Wenn Sie eine Datenbank bzw. einen Tabellenbereich wiederherstellen und die Option WITHOUT DATALINK nicht...
und

Wenn Sie eine Datenbank oder einen Tabellenbereich wiederherstellen und die Option WITHOUT DATALINK angeben...

durch folgenden Text:

Wenn Sie eine Datenbank bzw. einen Tabellenbereich wiederherstellen, müssen die folgenden Bedingungen erfüllt sein, damit die Wiederherstellungsoperation erfolgreich sein kann:

- o Wenn ein Data Links-Server, der in der Sicherungsdatei aufgezeichnet ist, nicht aktiv ist, ist die Wiederherstellung dennoch erfolgreich.

Tabellen mit DATALINK-Spalteninformationen, die durch den fehlenden Data Links-Server beeinflusst werden, werden in den Status DRP (DataLink Reconcile Pending) gesetzt, nachdem die Wiederherstellung (bzw. die aktualisierende Wiederherstellung) abgeschlossen ist. Bevor die Data Links-Server wieder als für die Datenbank verfügbar markiert werden können, muss diese Wiederherstellung erfolgreich abgeschlossen sein.

- o Wenn ein Data Links-Server, der in der Sicherungsdatei aufgezeichnet ist, während der Wiederherstellung inaktiv wird, schlägt die Wiederherstellung fehl. Sie kann erneut gestartet werden, während der Data Links-Server inaktiv ist (siehe oben).
- o Wenn eine vorherige Datenbankwiederherstellung auf einem Data Links-Server noch nicht abgeschlossen ist, schlagen nachfolgende Datenbank- oder Tabellenbereichswiederherstellungen fehl, bis diese Data Links-Server erneut gestartet wurden und die unvollständige Wiederherstellung abgeschlossen ist.
- o Informationen zu allen DATALINK-Spalten, die in der Sicherungsdatei aufgezeichnet werden, müssen in den Registriertabellen der entsprechenden Data Links-Server vorhanden sein.

Wenn nicht alle Informationen zu den DATALINK-Spalten in den Registriertabellen aufgezeichnet sind, wird die Tabelle mit den fehlenden DATALINK-Spalteninformationen in den Status DRNP (DataLink Reconcile Not Possible) gesetzt, nachdem die Wiederherstellung (bzw. die aktualisierende Wiederherstellung) abgeschlossen ist.

Wenn die Sicherung nicht in den Registriertabellen aufgezeichnet wird, kann dies bedeuten, dass die angegebene Sicherungsdatei älter ist als der Wert für "num_db_backups" und bereits eine Garbage Collection durchgeführt wurde. Dies bedeutet, dass die archivierten Dateien aus dieser früheren Sicherung entfernt wurden und nicht wiederhergestellt werden können. Alle Tabellen mit DATALINK-Spalten werden in den Status DRP (DataLink Reconcile Pending) gesetzt.

Wenn die Sicherung nicht in den Registriertabellen aufgezeichnet wird, kann dies bedeuten, dass das Sicherungsverfahren noch nicht abgeschlossen ist, da der Data Links-Server nicht aktiv ist. Alle Tabellen mit DATALINK-Spalten werden in den Status DRP (Datalink Reconcile Pending) gesetzt. Wenn der Data Links-Server erneut gestartet wird, wird die Sicherung vor der Wiederherstellung abgeschlossen.

Die Tabelle bleibt für Benutzer verfügbar. Allerdings verweisen die Werte in den DATALINK-Spalten möglicherweise nicht genau auf die Dateien (beispielsweise wird eine Datei, die einem Wert für die DATALINK-Spalte entspricht, nicht gefunden). Wenn dieses Verhalten unerwünscht ist, können Sie die Tabelle in den Status "Überprüfung anstehend" setzen. Setzen Sie dazu die Anweisung "SET CONSTRAINTS for tabellenname TO DATALINK RECONCILE PENDING" ab.

Wenn sich nach einer Wiederherstellungsoperation eine Tabelle im Status „Abstimmung nicht möglich“ befindet, können Sie die DATALINK-Spaltendaten mit einer der Methoden unter „Herausnehmen einer Tabelle aus dem Status "Datalink_Reconcile_Not_Possible" (Abstimmung nicht möglich)" korrigieren.

Die Anmerkung am Ende des ersten Absatzes bleibt unverändert.

Fügen Sie folgenden Text am Ende dieses Abschnitts hinzu:

Es ist sehr empfehlenswert, die Datei datalink.cfg zu archivieren, um bestimmte ungewöhnliche Wiederherstellungsfälle abzudecken, da die Datei datalink.cfg im Datenbanksicherungsbild nur die Datalink-Konfiguration zum Zeitpunkt der Sicherung widerspiegelt. Das Vorhandensein der neuesten Datei datalink.cfg ist erforderlich, um alle Wiederherstellungsfälle abzudecken. Daher muss diese Datei nach jedem Aufruf des Befehls ADD DATALINKS MANAGER oder DROP DATALINKS MANAGER gesichert werden. Dies hilft, die neueste Datei datalink.cfg abzurufen, wenn diese nicht auf Platte verfügbar ist.

Wenn die neueste Datei datalink.cfg nicht auf Platte verfügbar ist, ersetzen Sie die vorhandene Datei datalink.cfg (aus dem Sicherungsbild) durch die neueste Dateiversion, die vor der Ausführung einer aktualisierenden Wiederherstellung archiviert wurde. Dies muss nach der Wiederherstellung der Datenbank durchgeführt werden.

Wiederherstellen von Datenbanken von einer Offlinesicherung ohne aktualisierendes Wiederherstellen

Sie können nur auf Datenbankebene, nicht jedoch auf Tabellenbereichsebene eine Wiederherstellung ohne aktualisierendes Wiederherstellen durchführen. Wenn Sie eine Datenbank ohne aktualisierende Wiederherstellung wiederherstellen wollen, können Sie entweder eine nicht wiederherstellbare Datenbank (d. h. eine Datenbank mit Umlaufprotokollierung) wiederherstellen oder den Parameter WITHOUT ROLLING FORWARD im Befehl RESTORE DATABASE angeben.

Wenn Sie das Wiederherstellungsprogramm mit der Option WITHOUT DATALINK verwenden, werden alle Tabellen mit DATALINK-Spalten in den Status DRP (Data-Link Reconcile Pending) gesetzt. Dann wird bei der Wiederherstellung keine Abstimmung mit den Data Links-Servern durchgeführt.

Wenn Sie die Option WITHOUT DATALINK nicht verwenden und ein Data Links-Server in der Sicherungsdatei nicht mehr in der Datenbank definiert ist (d. h. mit dem Befehl DROP DATALINKS MANAGER gelöscht wurde), werden Tabellen mit DATALINK-Daten, die auf den gelöschten Data Links-Server verweisen, vom Wiederherstellungsprogramm in den Status DRP gesetzt.

Wenn Sie nicht die Option WITHOUT DATALINK verwenden, alle Data Links-Server verfügbar sind und alle Informationen zu den DATALINK-Spalten vollständig in den Registriertabellen aufgezeichnet sind, tritt folgender Fall für jeden in der Sicherungsdatei aufgeführten der Data Links-Server ein:

- Alle Dateien, die nach dem Sicherungsbild für die Datenbankwiederherstellung verknüpft wurden, werden als unverknüpft gekennzeichnet (da sie im Sicherungsbild nicht als verknüpft aufgezeichnet sind).
- Alle Dateien, deren Verbindung nach dem Sicherungsbild aufgehoben wurde, die jedoch vor der Erstellung des Sicherungsbilds verknüpft wurden, werden als verknüpft gekennzeichnet (da sie im Sicherungsbild als verknüpft aufgezeichnet sind). Wenn die Datei später mit einer anderen Tabelle in einer anderen Datenbank verknüpft wurde, wird die wiederhergestellte Tabelle in den Status DRP (DataLink Reconcile Pending) gesetzt.

Anmerkung: Die Ausführungen oben gelten nicht, wenn das Sicherungsbild für die Datenbankwiederherstellung erstellt wurde, als mindestens einer der Data Links-Server nicht aktiv war, da die DATALINK-Informationen in der Sicherung dann unvollständig sind. Die Ausführungen oben gelten ebenfalls nicht, wenn das für die Datenbankwiederherstellung verwendete Sicherungsbild nach einer Datenbankwiederherstellung mit oder ohne aktualisierende Wiederherstellung erstellt wurde. In beiden Fällen werden alle Tabellen mit DATALINK-Spalten in den Status DRP (DataLink Reconcile Pending) gesetzt, und während der Wiederherstellung erfolgt keine Abstimmung mit den Data Links-Servern.

Wiederherstellen von Datenbanken und Tabellenbereichen und aktualisierendes Wiederherstellen bis zum Ende der Protokolle

Wenn Sie die Datenbank oder den Tabellenbereich wiederherstellen und dann bis zum Ende der Protokolle aktualisierend wiederherstellen (d. h. dass alle Protokolle angegeben werden), ist eine Abstimmungsprüfung nur dann erforderlich, wenn mindestens einer der Data Links-Server in der Sicherungsdatei während der Wiederherstellungsoperation nicht aktiv ist. Wenn Sie nicht sicher sind, ob alle Protokolle für die aktualisierende Wiederherstellung angegeben wurden, oder glauben, dass Sie DATALINK-Werte abstimmen müssen, gehen Sie wie folgt vor:

1. Setzen Sie folgende SQL-Anweisung für die betroffenen Tabellen ab:

```
SET CONSTRAINTS FOR tabellenname TO DATALINK RECONCILE PENDING
```

Dadurch wird die Tabelle in den Status DRP (DataLink Reconcile Pending) und „Überprüfung anstehend“ gesetzt.

2. Wenn eine Tabelle nicht den Status „Überprüfung anstehend“ haben soll, setzen Sie die folgende SQL-Anweisung ab:

```
SET CONSTRAINTS FOR tabellenname IMMEDIATE CHECKED
```

Dadurch wird die Tabelle aus dem Status „Überprüfung anstehend“ genommen, behält jedoch den Status DRP (DataLink Reconcile Pending). Sie müssen das Dienstprogramm zum Abstimmen (Reconcile) verwenden, um die Tabelle aus diesem Status herauszunehmen.

Es kann vorkommen, dass die Sicherungsdatei DATALINK-Daten enthält, die auf einen DB2 Data Links Manager verweisen (d. h. ein DB2 Data Links Manager wurde zum Zeitpunkt der Sicherung in der Datenbank registriert), der aus der Datenbank gelöscht wurde. Für jede aktualisierend wiedergestellte Tabelle, die mindestens eine Tabelle mit DATALINK-Daten enthält, die auf den gelöschten DB2 Data Links Manager verweisen, werden alle Tabellen vom Wiederherstellungsprogramm in den Status DRP gesetzt.

DB2 Data Links Manager und Interaktionen bei Wiederherstellung

Die folgende Tabelle zeigt die verschiedenen Arten von Wiederherstellung, die Sie durchführen können, die DB2 Data Links Manager-Verarbeitung, die bei RESTORE- und ROLLFORWARD-Operationen auftritt, und ob Sie nach Beendigung der Wiederherstellung das Dienstprogramm Reconcile zum Abstimmen ausführen müssen:

Art der Wiederherstellung	DB2 Data Links Manager-Verarbeitung bei RESTORE	DB2 Data Links Manager-Verarbeitung bei ROLLFORWARD	Abstimmen mit Reconcile
Nicht wiederherstellbare Datenbank (<i>logretain=NO</i>)			
Datenbank-wiederherstellung einer Gesamtsicherung, alle Data Links-Server aktiv	Schnelle Abstimmung wird durchgeführt	N/V	Kann optional ausgeführt werden, wenn ein Problem mit Dateiverbindungen (Links) angenommen wird.
RESTORE für Datenbank mit Option WITHOUT DATALINK	Tabellen werden in den Status DRP (DataLink Reconcile Pending) gesetzt	N/V	Erforderlich

Art der Wiederherstellung	DB2 Data Links Manager-Verarbeitung bei RES-TORE	DB2 Data Links Manager-Verarbeitung bei ROLL-FORWARD	Abstimmen mit Reconcile
Datenbank-wiederherstellung einer vollständigen Sicherung, mindestens ein Data Links-Server inaktiv	Schnelle Abstimmung wird nur auf den Tabellen in Tabellenbereichen durchgeführt, die keine Verbindungen zu einem inaktiven Data Links-Server haben. Andere Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	N/V	Erforderlich für Tabellen in Tabellenbereichen mit Verbindungen zum inaktiven Data Links-Server
Datenbank-wiederherstellung einer unvollständigen Sicherung, alle Data Links-Server aktiv	Schnelle Abstimmung wird nicht durchgeführt, alle Tabellen mit DATALINK-Spalten werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	N/V	Erforderlich
Wiederherstellbare Datenbank (<i>logretain=YES</i>)			
Datenbank-wiederherstellung mit Hilfe von WITHOUT ROLLING FORWARD und einer vollständigen Sicherung, alle Data Links-Server aktiv	Schnelle Abstimmung wird durchgeführt	N/V	Optional
Datenbank-wiederherstellung mit Hilfe von WITHOUT ROLLING FORWARD und WITHOUT DATALINK und einer vollständigen oder unvollständigen Sicherung, alle Data Links-Server aktiv oder inaktiv	Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt	N/V	Erforderlich
Datenbank-wiederherstellung mit Hilfe von WITHOUT ROLLING FORWARD und einer vollständigen Sicherung, mindestens ein Data Links-Server inaktiv	Schnelle Abstimmung wird nur für Tabellen in Tabellenbereichen durchgeführt, die keine Verbindungen zu den inaktiven Data Links-Servern haben. Andere Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	N/V	Erforderlich für Tabellen in Tabellenbereichen mit Verbindungen zu den inaktiven Data Links-Servern
Datenbank-wiederherstellung mit Hilfe von WITHOUT ROLLING FORWARD und einer unvollständigen Sicherung, Data Links-Server aktiv oder inaktiv	Schnelle Abstimmung wird nicht durchgeführt, alle Tabellen mit DATALINK-Spalten werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	N/V	Erforderlich

Art der Wiederherstellung	DB2 Data Links Manager-Verarbeitung bei RES-TORE	DB2 Data Links Manager-Verarbeitung bei ROLL-FORWARD	Abstimmen mit Reconcile
Datenbank-wiederherstellung und aktualisierende Wiederherstellung zum Ende der Protokolle mit Hilfe einer vollständigen Sicherung, alle Data Links-Server aktiv	Keine Aktion	Keine Aktion	Optional
Datenbank-wiederherstellung und aktualisierende Wiederherstellung bis zum Ende der Protokolle mit Hilfe einer vollständigen Sicherung, mindestens ein Data Links-Server inaktiv während der aktualisierenden Wiederherstellung	Keine Aktion	Keine Aktion	Optional
Datenbank-wiederherstellung und aktualisierende Wiederherstellung bis zum Ende der Protokolle mit Hilfe einer vollständigen oder unvollständigen Sicherung, beliebige Data Links-Server inaktiv während der Wiederherstellung	Keine Aktion	Alle Tabellen mit DATALINK-Spalten werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	Erforderlich für alle Tabellen mit DATALINK-Spalten
Datenbank-wiederherstellung und aktualisierende Wiederherstellung zum Ende der Protokolle mit Hilfe einer unvollständigen Sicherung, alle Data Links-Server aktiv während der Wiederherstellung	Keine Aktion	Keine Aktion	Optional
Datenbank-wiederherstellung und aktualisierende Wiederherstellung bis zum Ende der Protokolle mit Hilfe einer vollständigen oder unvollständigen Sicherung, alle Data Links-Server aktiv, Sicherung auf keinem Data Links-Server bekannt	Keine Aktion	Alle Tabellen in Tabellenbereichen mit Verbindungen zu einem Data Links-Server, auf dem die Sicherung unbekannt ist, werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	Erforderlich

Art der Wiederherstellung	DB2 Data Links Manager-Verarbeitung bei RES-TORE	DB2 Data Links Manager-Verarbeitung bei ROLL-FORWARD	Abstimmen mit Reconcile
Tabellenbereichswiederherstellung und aktualisierende Wiederherstellung bis zum Ende der Protokolle mit Hilfe einer vollständigen Sicherung, alle Data Links-Server aktiv	Keine Aktion	Keine Aktion	Optional
Tabellenbereichswiederherstellung und aktualisierende Wiederherstellung bis zum Ende der Protokolle mit Hilfe einer vollständigen Sicherung, mindestens ein Data Links-Server inaktiv während der aktualisierenden Wiederherstellung	Keine Aktion	Keine Aktion	Optional
Tabellenbereichswiederherstellung und aktualisierende Wiederherstellung bis zum Ende der Protokolle mit Hilfe einer vollständigen oder unvollständigen Sicherung, beliebige Data Links-Server inaktiv während der Wiederherstellung	Keine Aktion	Alle Tabellen in Tabellenbereichen mit Verbindungen zu einem inaktiven Data Links-Server werden in den Status DRP (Datalink Reconcile Pending) gesetzt.	Erforderlich für Tabellen in Tabellenbereichen mit Verbindungen zu einem inaktiven Data Links-Server
Tabellenbereichswiederherstellung und aktualisierende Wiederherstellung zum Ende der Protokolle mit Hilfe einer unvollständigen Sicherung, alle Data Links-Server aktiv	Keine Aktion	Keine Aktion	Optional
Datenbankwiederherstellung und aktualisierende Wiederherstellung bis zu einem bestimmten Zeitpunkt mit Hilfe einer vollständigen oder unvollständigen Sicherung, beliebige Data Links-Server aktiv oder inaktiv während der Wiederherstellung und/oder aktualisierenden Wiederherstellung	Keine Aktion	Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt	Erforderlich

Art der Wiederherstellung	DB2 Data Links Manager-Verarbeitung bei RES-TORE	DB2 Data Links Manager-Verarbeitung bei ROLL-FORWARD	Abstimmen mit Reconcile
Tabellenbereichswiederherstellung und aktualisierende Wiederherstellung bis zu einem bestimmten Zeitpunkt mit Hilfe einer vollständigen oder unvollständigen Sicherung, Data Links-Server aktiv oder inaktiv während der Wiederherstellung und/oder aktualisierenden Wiederherstellung	Keine Aktion	Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt	Erforderlich
Datenbankwiederherstellung auf einen anderen Datenbanknamen, Aliasnamen, Host-Namen oder ein anderes Exemplar ohne aktualisierende Wiederherstellung 1 auf Seite 100	Tabellen werden in den Status DRNP (Datalink Reconcile Not Possible) gesetzt	N/V	Optional, aber Tabellen im Status DRNP (Datalink Reconcile Not Possible) müssen manuell korrigiert werden.
Datenbankwiederherstellung auf einen anderen Datenbanknamen, Aliasnamen, Host-Namen oder ein anderes Exemplar und aktualisierende Wiederherstellung	Keine Aktion	Tabellen werden in den Status DRNP (Datalink Reconcile Not Possible) gesetzt	Optional, aber Tabellen im Status DRNP (Datalink Reconcile Not Possible) müssen manuell korrigiert werden.
Datenbankwiederherstellung von einer nicht nutzbaren Sicherung (Image mit Garbage Collection auf dem Data Links-Server) ohne aktualisierende Wiederherstellung 1 auf Seite 100, mit oder ohne Option WITHOUT DATALINK	Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt	Keine Aktion	Erforderlich
Datenbankwiederherstellung von einer nicht nutzbaren Sicherung (Image mit Garbage Collection auf dem Data Links-Server) und aktualisierende Wiederherstellung, mit oder ohne Option WITHOUT DATALINK	Keine Aktion	Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt	Erforderlich

Art der Wiederherstellung	DB2 Data Links Manager-Verarbeitung bei RES-TORE	DB2 Data Links Manager-Verarbeitung bei ROLL-FORWARD	Abstimmen mit Reconcile
Tabellenbereichswiederherstellung von einer nicht nutzbaren Sicherung (Image mit Garbage Collection auf dem Data Links-Server) und aktualisierende Wiederherstellung	Keine Aktion	Tabellen werden in den Status DRP (Datalink Reconcile Pending) gesetzt	Erforderlich

Anmerkungen:

1. Wiederherstellung mit Hilfe einer Offlinesicherung und der Option WITHOUT ROLLING FORWARD (*logretain* ist aktiviert) oder Wiederherstellung mit Hilfe einer Offlinesicherung (*logretain* ist inaktiviert)
2. Eine *vollständige* Sicherung ist eine Sicherung, die ausgeführt wurde, während alle erforderlichen Data Links-Server aktiv waren. Eine *unvollständige* Sicherung ist eine Sicherung, die ausgeführt wurde, während mindestens ein Data Links-Server inaktiv war.
3. Die schnelle Abstimmung wird nicht durchgeführt, wenn das Sicherungsimago, das für die Datenbankwiederherstellung verwendet wurde, nach einer Datenbankwiederherstellung mit oder ohne aktualisierender Wiederherstellung erstellt wurde. In diesem Fall werden alle Tabellen mit DATALINK-Spalten in den Status DRP (Datalink Reconcile Pending) gesetzt.

Erkennen von Situationen, die Abstimmung erfordern

Im Folgenden werden einige Situationen beschrieben, in denen Sie eventuell das Dienstprogramm Reconcile ausführen müssen:

- Die gesamte Datenbank wird mit RESTORE wiederhergestellt und bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt. Da die gesamte Datenbank aktualisierend bis zu einer festgeschriebenen Transaktion wiederhergestellt wird, hat keine Tabelle den Status „Überprüfung anstehend“ (aufgrund von referenziellen Integritätsbedingungen oder Prüfungen auf Integritätsbedingung). Sämtliche Daten in der Datenbank werden in einen konsistenten Status gebracht. Die DATALINK-Spalten können jedoch nicht mit den Metadaten im DB2 Data Links Manager synchronisiert werden, und eine Abstimmung ist erforderlich. In dieser Situation haben Tabellen mit DATALINK-Daten bereits den Status DRP. Sie sollten das Dienstprogramm **reconcile** für jede dieser Tabellen aufrufen.
- Ein bestimmter Data Links-Server, der den DB2 Data Links Manager ausführt, kann seine Metadaten nicht mehr verfolgen. Dies kann aus verschiedenen Gründen eintreten. Beispiel:
 - Mit dem Data Links-Server wurde ein Kaltstart ausgeführt.
 - Die Metadaten des Data Links-Servers wurden in einem früheren Status wiederhergestellt.

In einigen Situationen, z. B. bei den SQL-Anweisungen UPDATE und DELETE, ist DB2 eventuell nicht in der Lage, ein Problem mit den Metadaten in einem Data Links-Server zu erkennen. In diesen Fällen schlägt die SQL-Anweisung fehl. Sie würden die Tabelle mit der Anweisung SET CONSTRAINTS in den Status DRP setzen und dann das Dienstprogramm **reconcile** für diese Tabelle ausführen.

- Ein Dateisystem ist nicht verfügbar (zum Beispiel aufgrund eines Plattenausfalls) und wird nicht auf den aktuellen Stand wiederhergestellt. In diesem Fall fehlen möglicherweise Dateien.
- Ein DB2 Data Links Manager wird aus einer Datenbank gelöscht, und es gibt DATALINK FILE LINK CONTROL-Werte, die auf diesen DB2 Data Links Manager verweisen. Sie sollten das Dienstprogramm **reconcile** für solche Tabellen ausführen.

Anhang C. Benutzer-Exit zur Datenbankwiederherstellung

Im Abschnitt „Hinweise zum Archivieren und Abrufen“ ist folgender Absatz nicht mehr zutreffend und sollte aus der Liste entfernt werden:

Ein Benutzer-Exit-Programm kann unterbrochen werden, wenn die Verbindung eines fernen Clients zum DB2-Server verloren geht. Das heißt, wenn während der Archivierung von Protokollen über einen Benutzer-Exit einer der anderen über SNA verbundenen Clients die Verbindung verliert oder ausgeschaltet wird, wird ein Signal (SIGUSR1) an den Server gesendet. Der Server leitet dieses Signal an den Benutzer-Exit weiter und verursacht eine Unterbrechung. Das Benutzer-Exit-Programm kann so geändert werden, dass es auf eine Unterbrechung prüft und anschließend die Verarbeitung fortsetzt.

Der Abschnitt „Fehlerbehandlung“ enthält eine Liste mit Anmerkungen, in welcher der Inhalt von Anmerkung 3 durch folgende Informationen ersetzt werden muss:

- Aufrufe für Benutzer-Exit-Programme werden fünf Minuten lang ausgesetzt. Während dieser Zeit werden alle Anforderungen ignoriert, einschließlich der Protokolldateianforderung, die den Rückkehrcode verursachte.

Nach Ablauf der fünfminütigen Aussetzung, in der keine Anforderungen verarbeitet wurden, wird die nächste Anforderung verarbeitet. Wenn bei Verarbeitung dieser Anforderung kein Fehler auftritt, wird die Verarbeitung neuer Anforderungen des Benutzer-Exit-Programms fortgesetzt und DB2 setzt die Archivierungsanforderung für die Protokolldateien erneut ab, deren Archivierung zuvor fehlgeschlagen oder ausgesetzt worden war. Wenn ein Rückkehrcode von größer als 8 während der Wiederholung generiert wird, werden Anforderungen für weitere fünf Minuten ausgesetzt. Die fünfminütigen Aussetzungen werden solange fortgesetzt, bis das Problem behoben wurde oder die Datenbank gestoppt und erneut gestartet wurde.

Sobald alle Verbindungen von Anwendungen zur Datenbank getrennt wurden und die Datenbank wieder geöffnet wurde, setzt DB2 die Archivierungsanforderung für diejenigen Protokolldateien ab, die bei der vorangegangenen Verwendung der Datenbank möglicherweise nicht erfolgreich archiviert wurden.

Wenn das Benutzer-Exit-Programm die Protokolldateien nicht archivieren konnte, befinden sich möglicherweise viele Protokolldateien auf Ihrer Platte und die Leistung des Systems kann sich durch den zusätzlichen Aufwand zur Formatierung dieser Protokolldateien verschlechtern. Wenn die Platte voll ist, nimmt der Datenbankmanager keine weiteren Anwendungsanforderungen für Datenbankänderungen mehr entgegen.

Wenn das Benutzer-Exit-Programm zum Abrufen von Protokolldateien aufgerufen wurde, wird die aktualisierende Wiederherstellung ausgesetzt, aber nicht gestoppt, außer wenn ein Stopp im Dienstprogramm ROLLFORWARD DATABASE angegeben wurde. Wurde kein Stopp angegeben, können Sie den Fehler beheben und die Wiederherstellung wiederaufnehmen.

Anhang D. Absetzen von Befehlen an mehrere Datenbankpartitions-server

Fügen Sie am Ende des Abschnitts zum Angeben des auszuführenden Befehls Folgendes hinzu:

Wenn Sie ein Shell-Script der Korn-Shell ausführen, das Logiken enthält, um im Hintergrund von stdin zu lesen, sollten Sie stdin ausdrücklich auf eine Quelle umleiten, von der der Prozess lesen kann, ohne dass er auf dem Terminal gestoppt wird (Nachricht SIGTTIN). Sie können ein Script mit der folgenden Form ausführen, um stdin umzuleiten:

```
shell_script </dev/null &
```

Dies gilt, wenn keine Eingabe gemacht werden soll.

Auf ähnliche Weise können Sie immer `</dev/null` angeben, wenn der Befehl `db2_all` im Hintergrund ausgeführt wird. Beispiel:

```
db2_all ";diesen_befehl_ausführen" </dev/null &
```

Indem Sie dies tun, können Sie stdin umleiten und verhindern, dass das Script auf dem Terminal gestoppt wird.

Wenn Sie die Ausgabe eines fernen Befehls nicht interessiert, gibt es eine Alternative zu dieser Methode. Mit dem folgenden Befehl können Sie die Option „daemonize“ im Präfix `db2_all` verwenden:

```
db2_all ";diesen_befehl_dämonisieren" &
```

Anhang I. Knotenübergreifende Hochgeschwindigkeitskommunikation

Der folgende Abschnitt wurde aktualisiert:

Aktivieren von DB2 zur Ausführung mit VI

Ausführliche Installationsanweisungen finden Sie im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg*.

Nach dem Abschluss der Installation von DB2 auf der Basis der Informationen im Handbuch *DB2 Enterprise - Extended Edition für Windows Einstieg* müssen Sie die folgenden DB2-Variablen der Profilregistrierdatenbank definieren und die folgenden Aufgaben auf jedem Datenbankpartitions-Server des Exemplars ausführen:

- Setzen Sie `DB2_VI_ENABLE=ON`.

Verwenden Sie den Befehl `db2set`, um den Wert der Registrierungsvariable zu ändern. Verwenden Sie den Befehl `db2_all`, um den Befehl `db2set` auf allen Datenbankpartitions-Servern des Exemplars auszuführen. Sie müssen mit einem Benutzerkonto angemeldet sein, das zur Administratorengruppe gehört, um den Befehl `db2_all` ausführen zu können.

Im folgenden Beispiel wird das Zeichen `;` innerhalb der doppelten Anführungszeichen geschrieben, damit die Anforderung gleichzeitig auf allen Datenbankpartitions-Servern des Exemplars ausgeführt werden kann:

```
db2_all ";db2set DB2_VI_ENABLE=ON"
```

Weitere Informationen zum Befehl `db2_all` finden Sie im Abschnitt "Absetzen von Befehlen an mehrere Datenbankpartitions-Server" in *Systemverwaltung: Implementierung*.

- Setzen Sie DB2_VI_DEVICE=nic0.

Beispiel:

```
db2_a11 ";db2set DB2_VI_DEVICE=nic0"
```

Anmerkung: Bei Synfinity Interconnect sollte für diese Variable der Wert DB2_VI_DEVICE=VINIC angegeben sein. Der Einheitenname (VINIC) muss in Großbuchstaben eingegeben werden.

- Setzen Sie DB2_VI_VIPL=vipl.dll.

Beispiel:

```
db2_a11 ";db2set DB2_VI_VIPL=vipl.dll"
```

Anmerkung: Der im Beispiel verwendete Wert ist der Standardwert für die Registrierungsvariable. Weitere Informationen zu Registrierungsvariablen finden Sie in *Systemverwaltung: Optimierung*.

- Geben Sie db2start auf dem MPP-Exemplar ein.
- Sehen Sie sich die Datei „db2diag.log“ an. Es sollte für jede Partition eine Nachricht vorhanden sein, die angibt, dass VI aktiviert ist.
- Möglicherweise müssen die FCM-Konfigurationsparameter (Fast Communications Manager) aktualisiert werden. Wenn aufgrund der geltenden Ressourceneinschränkungen unter FCM Probleme auftreten, müssen Sie die Werte für die FCM-Konfigurationsparameter anheben. Wenn Sie von einer anderen Umgebung für Hochgeschwindigkeitsverbindungen wechseln, in der die Werte für die FCM-Konfigurationsparameter angehoben wurden, müssen diese Werte möglicherweise wieder gesenkt werden. Unter Windows NT kann es außerdem erforderlich sein, die Variable DB2NTMEMSIZE der Profilregistrierdatenbank so zu definieren, dass die DB2-Standardwerte überschrieben werden. Weitere Informationen zu Registrierungsvariablen finden Sie in *Systemverwaltung: Optimierung*.

Systemverwaltung: Optimierung

Kapitel 3. Überlegungen zu Anwendungen

Angeben der Isolationsstufe

Fügen Sie im Abschnitt „Angeben der Isolationsstufe“ folgende Absätze hinzu:

Zusätzlich zur Einstellung der Isolationsstufe auf Paketebene bei der Vorbereitung oder Bindung einer Anwendung können Sie eine Isolationsstufe auf Anweisungsebene definieren. Eine Isolationsstufe auf Anweisungsebene wird mit Hilfe der Klausel WITH angegeben.

Die folgenden SQL-Anweisungen unterstützen Isolation auf Anweisungsebene:

- Anweisung SELECT
- SELECT INTO
- Gezielter Löschvorgang (Searched DELETE)
- INSERT
- Aktualisierung mit Suche (Searched UPDATE)
- DECLARE CURSOR

Für die Verwendung der Isolation auf Anweisungsebene gelten einige Bedingungen:

- Die Klausel WITH kann nicht in Unterabfragen verwendet werden.
- Die Option WITH UR gilt nur für Nur-Lese-Operationen. Wenn die Anweisung in anderen Fällen verwendet wird, wird sie automatisch von „UR“ in „CS“ geändert.
- Die Standardisolationsstufe für eine Anweisung ist die Isolationsstufe des Pakets, in dem die Anweisung gebunden ist.
- Die Isolationsstufe auf Anweisungsebene setzt die Isolationsstufe für das Paket außer Kraft, in dem die Anweisung enthalten ist.

Anpassen der Optimierungsklasse

Ersetzen Sie im Abschnitt „Anpassen der Optimierungsklasse“ die Informationen zu Ebene 2 durch folgende Informationen:

- 2 - Diese Klasse steuert das Optimierungsprogramm, so dass es eine Optimierung verwendet, die die Optimierung der Klasse 1 bedeutend verbessert, während gleichzeitig die Kompilierungskosten für komplexe Abfragen beträchtlich geringer gehalten werden als bei Klasse 3 und höher. Das bedeutet:
- Alle verfügbaren Statistikdaten, einschließlich Frequenz und nicht einheitliche Quantilverteilungsstatistik, werden verwendet.
 - Alle Regeln für das Umschreiben der Abfrage werden angewendet, einschließlich der Weiterleitung von Abfragen an Übersichtstabellen, mit Ausnahme von rechenintensiven Regeln, die nur in sehr seltenen Fällen gültig sind.
 - Es wird die schnelle Verknüpfungszählung (Greedy Join Enumeration) verwendet.

- Eine Vielzahl von Zugriffsmethoden wird berücksichtigt, darunter Vorab-
lesezugriff über Listen und Weiterleitung an Übersichtstabellen.
- Die Star-Join-Strategie wird, sofern zutreffend, berücksichtigt.

Die Optimierungsklasse 2 ist der Klasse 5 sehr ähnlich; allerdings verwendet sie die schnelle Verknüpfungsaufzählung (Greedy Join Enumeration) anstelle von dynamischer Programmierung. Diese Klasse bietet die beste Optimierung aller Optimierungsklassen, die den Algorithmus für die schnelle Verknüpfungsaufzählung (Greedy Join Enumeration) verwenden, der weniger Alternativen für komplexe Abfragen berücksichtigt und daher weniger Kompilierzeit als die Klassen 3 und höher benötigt. Sie wird daher für sehr komplexe Abfragen in Entscheidungshilfe- oder OLAP-Umgebungen (Online Analytic Processing) empfohlen. In solchen Fällen besteht eine sehr hohe Wahrscheinlichkeit, dass dieselbe Abfrage selten ausgeführt wird, so dass ihr Zugriffsplan wahrscheinlich nicht im Cache bleibt, bis die Abfrage das nächste Mal auftritt.

Dynamische Compound-Anweisungen

Dieser neue Abschnitt wird nach dem Abschnitt „Compound-SQL-Anweisungen“ hinzugefügt:

Eine dynamische Compound-Anweisung gruppiert andere SQL-Anweisungen zu einem ausführbaren Block. In der dynamischen Compound-Anweisung können Sie SQL-Variablen und Bedingungen für SQLSTATE-Werte deklarieren und eine oder mehrere prozedurale SQL-Anweisungen aufnehmen. Wenn ein Fehler in der dynamischen Compound-Anweisung auftritt, werden alle bisherigen SQL-Anweisungen rückgängig gemacht, und die verbleibenden SQL-Anweisungen in der dynamischen Compound-Anweisung werden nicht verarbeitet.

Die dynamische Compound-Anweisung kann in einen Auslöser, eine SQL-Funktion oder eine SQL-Methode eingebettet oder über dynamische SQL-Anweisungen abgesetzt werden. Diese ausführbare Anweisung kann dynamisch vorbereitet werden. Es sind keine Zugriffsrechte zum Aufrufen der Anweisung erforderlich. Die Berechtigungs-ID für die Anweisung muss jedoch die notwendigen Zugriffsrechte haben, um die eingebetteten SQL-Anweisungen in der Compound-Anweisung aufrufen zu können.

Variablen sind in den Unteranweisungen in der Variablendeklaration vorhanden. Bedingungen sind in den Unteranweisungen basierend auf den SQLSTATE-Werten der Bedingungsdeklaration vorhanden. Dynamische Compound-Anweisungen werden von DB2 wie eine einzige Anweisung kompiliert. Diese Anweisung kann effizient für kurze Scripts mit wenig Steuerungsflusslogik, jedoch mit beträchtlichem Datenfluss verwendet werden. Bei größeren Konstrukten mit verschachteltem komplexem Steuerungsfluss sollten Sie die Verwendung von SQL-Prozeduren erwägen.

Es gibt mehrere Steuerungsfluss-Logikanweisungen, die in der dynamischen Compound-Anweisung verwendet werden können. Dies sind: die Anweisung FOR, die Anweisung IF, die Anweisung ITERATE und die Anweisung WHILE. Ausführliche Informationen zu diesen Anweisungen und den anderen unterstützten Anweisungen finden Sie im Handbuch *SQL Reference*.

Kapitel 4. Überlegungen zur Umgebung

Fügen Sie nach dem Abschnitt „Verwenden von Index Advisor“ folgenden Abschnitt hinzu:

Größere Indexschlüssel

Es kann zugelassen werden, dass Spalten mit mehr als 255 Byte als Teil eines Indexschlüssels angegeben werden. Die Registrierungsvariable `DB2_INDEX_2BYTEVARLEN` erlaubt die Verwendung von 2 Byte statt 1 Byte zur Speicherung der Länge eines Indexschlüssels.

Von den Änderungen an der Registrierungsvariablen sind mehrere SQL-Anweisungen betroffen. Dies sind:

- `CREATE TABLE`. Primärschlüssel, Fremdschlüssel und eindeutige Schlüssel mit Variablenschlüsselkomponenten können größer als 255 Byte sein.
- `CREATE INDEX`. Alle Indizes, einschließlich eindeutiger Indizes und `INCLUDE`-Spalten, mit Variablenschlüsselkomponenten können größer als 255 Byte sein.
- `ALTER TABLE`. Primärschlüssel, Fremdschlüssel und eindeutige Schlüssel mit Variablenschlüsselkomponenten können größer als 255 Byte sein. Alle Indizes, einschließlich eindeutiger Indizes und `INCLUDE`-Spalten, mit Variablenschlüsselkomponenten können größer als 255 Byte sein. Die Fremdschlüsselbeschränkung auf 255 Byte gilt unabhängig vom Wert der Registrierungsvariablen nicht mehr. Die Wahrheitsbedingung für den Primärschlüssel entsprechend dem Fremdschlüssel erzwingt alle Einschränkungen und Begrenzungen.

Gehen Sie wie folgt vor, um vorhandene Indizes so zu konvertieren, dass sie größere Indexschlüssel nutzen können: Löschen Sie die Indizes, setzen Sie die Registrierungsvariable `DB2_INDEX_2BYTEVARLEN` auf `ON`, und erstellen Sie dann die Indizes erneut (mit den größeren Spalten). Weitere Informationen zu den SQL-Anweisungen und der Syntaxbeschreibung finden Sie im Handbuch *SQL Reference*.

Kapitel 5. Systemkatalogstatistiken

Erfassen und Verwenden von Verteilungsstatistiken

Im Abschnitt „Beispiel zur Auswirkung auf Gleichheitsvergleichselemente“ wird das Vergleichselement `C <= 10` behandelt. Der Fehler wird mit dem Wert -86% angegeben. Dies ist nicht korrekt. Der Satz am Absatzende muss wie folgt lauten:

Unter der Annahme einer gleichmäßigen Datenverteilung und unter Verwendung der Formel (1) wird die Anzahl der Zeilen, die das Vergleichselement erfüllen, auf 1 geschätzt, d. h., es entsteht ein Fehler von -87,5%.

Im Abschnitt „Beispiel zur Auswirkung auf Gleichheitsvergleichselemente“ wird das Vergleichselement `C > 8,5 AND C <= 10` behandelt. Die Schätzung des Werts `r_2` mit Hilfe linearer Interpolation muss wie folgt geändert werden:

$$r_2 *= \frac{10 - 8,5}{100 - 8,5} \times (\text{Anzahl der Zeilen mit dem Wert } > 8,5 \text{ und } \leq 100,0)$$
$$r_2 *= \frac{10 - 8,5}{100 - 8,5} \times (10 - 7)$$

$$r_2 = \frac{1,5}{91,5} \times (3)$$

$$r_2 = 0$$

Der Absatz im Anschluss an dieses neue Beispiel muss wie folgt geändert werden:

Der endgültige Schätzwert ist $r_1 + r_2 = 7$,
und der Fehler ist nur -12,5%.

Regeln zur Aktualisierung von Katalogstatistiken

Im Abschnitt „Regeln zur Aktualisierung von Spaltenstatistiken“ sollte der letzte Listenpunkt durch Folgenden ersetzt werden:

HIGH2KEY muss größer als LOW2KEY sein, wenn mehr als drei eindeutige Werte in der entsprechenden Spalte vorhanden sind. Im Fall von höchstens drei eindeutigen Werten in der Spalte kann HIGH2KEY LOW2KEY entsprechen.

Subelementstatistik

In FixPak 1 gibt es eine Option zum Erfassen und Verwenden von Subelementstatistikdaten. Dies sind Statistikdaten zum Inhalt von Daten in Spalten, wenn die Daten eine Struktur in der Form einer Reihe von durch Leerzeichen getrennten Teilfeldern oder Subelementen haben.

Angenommen, Sie haben eine Datenbank mit einer Tabelle DOCUMENTS, in der jede Zeile ein Dokument beschreibt, und diese Tabelle hat eine Spalte KEYWORDS mit einer Liste der relevanten Schlüsselwörter zu diesem Dokument für den Textabruf. Die Werte in KEYWORDS würden dann wie folgt aussehen:

```
'Datenbank Simulation analytisch Information Management'  
'Simulation Modell Fruchtfliege Reproduktion Temperatur'  
'Forstwirtschaft Fichte Boden Erosion Regen'  
'Wald Temperatur Boden Niederschlag Feuer'
```

In diesen Beispiel besteht jeder Spaltenwert aus 5 Subelementen, die jeweils aus einem Wort (dem Schlüsselwort) bestehen und durch ein Leerzeichen voneinander getrennt sind.

Bei allen Abfragen mit LIKE-Vergleichselementen für solche Spalten, die das Platzhalterzeichen % verwenden:

```
SELECT .... FROM DOCUMENTS WHERE KEYWORDS LIKE '%Simulation%'
```

ist es für das Optimierungsprogramm oft vorteilhaft, einige der Basisstatistiken zur Subelementstruktur zu kennen, insbesondere:

SUB_COUNT

Durchschnittliche Anzahl von Subelementen.

SUB_DELIM_LENGTH

Durchschnittliche Länge jedes Begrenzers zwischen den einzelnen Subelementen, wobei ein Begrenzer in diesem Kontext aus einem oder mehreren aufeinanderfolgenden Leerzeichen besteht.

Im Beispiel mit der Spalte KEYWORDS beträgt SUB_COUNT 5 und SUB_DELIM_LENGTH 1, da jeder Begrenzer ein einziges Leerzeichen ist.

In FixPak 1 steuert der Systemadministrator das Erfassen und Verwenden dieser Statistiken durch eine Erweiterung der Registrierungsvariablen

DB2_LIKE_VARCHAR. Diese Registrierungsvariable wirkt sich auf die Art und Weise aus, in der das DB2 UDB-Optimierungsprogramm mit einem Vergleichselement der folgenden Form umgeht:

```
COLUMN LIKE '%xxxxxx'
```

Dabei ist *xxxxxx* eine beliebige Zeichenfolge, d. h. jedes LIKE-Vergleichselement, dessen Suchwert mit einem Prozentzeichen (%) beginnt. (Es kann, muss aber nicht mit einem Prozentzeichen enden). Diese Vergleichselemente werden im Folgenden als "LIKE-Platzhalter-Vergleichselemente" bezeichnet. Bei allen Vergleichselementen muss das Optimierungsprogramm schätzen, wie viele Zeilen mit dem Vergleichselement übereinstimmen. Bei LIKE-Platzhalter-Vergleichselementen geht das Optimierungsprogramm davon aus, dass die abgeglichene Spalte eine Struktur mit einer Reihe von Elementen hat, die miteinander zur gesamten Spalte verkettet sind. Es schätzt auch die Länge jedes Elements auf der Grundlage der Länge der Zeichenfolge einschließlich führender und abschließender Prozentzeichen. Die neue Syntax lautet:

```
db2set DB2_LIKE_VARCHAR=[Y|N|S|num1][,Y|N|num2]
```

Dabei gilt Folgendes:

- Der erste Ausdruck (vor dem Komma) hat folgende Bedeutung, allerdings nur für Spalten, die keine positive Subelementstatistik haben:

S	Verwendet den Algorithmus wie in DB2 Version 2.
N	Verwendet einen Subelementalgorithmus mit fester Länge.
Y (Standard)	Verwendet einen Subelementalgorithmus variabler Länge mit einem Standardwert für den Algorithmusparameter.
num1	Verwendet einen Subelementalgorithmus variabler Länge und "num1" als Algorithmusparameter.

- Der zweite Ausdruck (nach dem Komma) hat folgende Bedeutung:

N (Standard)	Erfasst oder verwendet keine Subelementstatistiken.
Y	Erfasst Subelementstatistiken. Verwendet einen Subelementalgorithmus variabler Länge, der diese Statistiken verwendet, zusammen mit einem Standardwert für den Algorithmusparameter bei Spalten mit positiven Subelementstatistiken.
num2	Erfasst Subelementstatistiken. Verwendet einen Subelementalgorithmus variabler Länge, der diese Statistiken verwendet, zusammen mit "num2" als Algorithmusparameter bei Spalten mit positiven Subelementstatistiken.

Wenn der Wert von DB2_LIKE_VARCHAR nur den ersten Ausdruck enthält, werden keine Subelementstatistiken erfasst und bereits erfasste Subelementstatistiken werden ignoriert. Der angegebene Wert wirkt sich wie oben angegeben in der Art und Weise aus, in der das Optimierungsprogramm die Auswahlmöglichkeit von LIKE-Platzhalter-Vergleichselementen berechnet:

- Wenn der Wert "S" ist, verwendet das Optimierungsprogramm denselben Algorithmus wie in DB2 Version 2, der nicht von einem Subelementmodell ausgeht.
- Wenn der Wert "N" ist, verwendet das Optimierungsprogramm einen Algorithmus, der von einem Subelementmodell und einer festen Spaltenlänge ausgeht, selbst wenn sie als variable Länge definiert ist.
- Wenn der Wert "Y" (Standardwert) oder eine Gleitkommakonstante ist, verwendet das Optimierungsprogramm einen Algorithmus, der von einem Subelementmodell ausgeht und erkennt, dass die Spalte eine variable Länge hat, wenn sie so definiert ist. Es leitet Subelementstatistiken auch von der Abfrage selbst statt von den Daten ab. Dieser Algorithmus umfasst einen Parameter (den "Algorithmusparameter"), der angibt, um wie viel länger das Element als die von den Prozentzeichen eingeschlossene Zeichenfolge ist.

- Wenn der Wert "Y" ist, verwendet das Optimierungsprogramm einen Standardwert von 1,9 für den Algorithmusparameter.
- Wenn der Wert eine Gleitkommakonstante ist, verwendet das Optimierungsprogramm den angegebenen Wert als Algorithmusparameter. Diese Konstante muss zwischen 0 und 6,2 liegen.

Wenn der Wert von DB2_LIKE_VARCHAR zwei Ausdrücke enthält und der zweite "Y" oder eine Gleitkommakonstante ist, werden Subelementstatistiken in SBCS-Zeichenfolgespalten vom Typ CHAR, VARCHAR, GRAPHIC oder VARGRAPHIC bei einer RUNSTATS-Operation erfasst und bei der Kompilierung von Abfragen mit LIKE-Platzhalter-Vergleichselementen verwendet. Das Optimierungsprogramm verwendet einen Algorithmus, der vom Subelementmodell ausgeht und die Statistik SUB_COUNT und SUB_DELIM_LENGTH sowie einen Algorithmusparameter zum Berechnen der Auswahlmöglichkeit des Vergleichselements verwendet. Der Algorithmusparameter wird auf dieselbe Weise angegeben wie der inferenzielle Algorithmus, d. h.:

- Wenn der Wert "Y" ist, verwendet das Optimierungsprogramm einen Standardwert von 1,9 für den Algorithmusparameter.
- Wenn der Wert eine Gleitkommakonstante ist, verwendet das Optimierungsprogramm den angegebenen Wert als Algorithmusparameter. Diese Konstante muss zwischen 0 und 6,2 liegen.

Wenn das Optimierungsprogramm bei der Kompilierung feststellt, dass keine Subelementstatistiken für die Spalte in der Abfrage erfasst wurden, verwendet es den "inferenziellen" Subelementalgorithmus, d. h. den Algorithmus, der verwendet wird, wenn nur der erste Ausdruck von DB2_LIKE_VARCHAR angegeben wird. Damit die Subelementstatistiken vom Optimierungsprogramm verwendet werden, muss daher der zweite Ausdruck von DB2_LIKE_VARCHAR sowohl während RUNSTATS als auch während der Kompilierung gesetzt sein.

Die Werte der Subelementstatistiken können durch Abfragen von SYSIBM.SYSCOLUMNS angezeigt werden. Beispiel:

```
select substr(NAME,1,16), SUB_COUNT, SUB_DELIM_LENGTH
from sysibm.syscolumns where tname = 'DOCUMENTS'
```

Die Spalten SUB_COUNT und SUB_DELIM_LENGTH sind in der Statistiksicht SYSSTAT.COLUMNS nicht vorhanden und können daher nicht aktualisiert werden.

Anmerkung: RUNSTATS kann etwas länger dauern, wenn diese Option verwendet wird. RUNSTATS kann beispielsweise bei einer Tabelle mit fünf Zeichenspalten 15 bis 40% länger dauern, wenn die Optionen DETAILED und DISTRIBUTION nicht verwendet werden. Wenn entweder DETAILED oder DISTRIBUTION als Option angegeben wird, ist der prozentuale Zusatzaufwand geringer, obwohl der absolute Zusatzaufwand gleich ist. Wenn Sie diese Option verwenden wollen, sollten Sie den Zusatzaufwand gegenüber den Verbesserungen der Abfrageleistung abwägen.

Kapitel 6. Der SQL-Compiler

In folgenden Abschnitten sind Änderungen erforderlich:

Replizierte Übersichtstabellen

Die folgenden Informationen ersetzen oder ergänzen die bereits vorhandenen Informationen in diesem Abschnitt:

Mit replizierten Übersichtstabellen kann die Zusammenfassung von Verknüpfungen durch Kollokation unterstützt werden. In einem Sternschema mit einer großen Fakttable, die sich über zwanzig Knoten erstreckt, sind beispielsweise die Verknüpfungen zwischen der Fakttable und den Dimensionstabellen dann am effizientesten, wenn diese Tabellen durch Kollokation zusammengefasst sind.

Wenn alle Tabellen in dieselbe Knotengruppe versetzt würden, wäre höchstens eine Dimensionstabelle korrekt für eine zusammengefasste Verknüpfung partitioniert. Alle anderen Dimensionstabellen könnten nicht in einer zusammengefassten Verknüpfung verwendet werden, da die Verknüpfungsspalte(n) in der Fakttable nicht mit dem Partitionierungsschlüssel der Fakttable übereinstimmen würde(n).

Beispielsweise könnte eine Tabelle FACT (C1, C2, C3, ...) in C1 partitioniert sein, eine Tabelle DIM1 (C1, dim1a, dim1b, ...) in C1, eine Tabelle DIM2 (C2, dim2a, dim2b, ...) in C2, usw.

In diesem Beispiel könnten Sie erkennen, dass die Verknüpfung zwischen FACT und DIM1 korrekt ist, da das Vergleichselement DIM1.C1 = FACT.C1 zusammengefasst würde. Die beiden Tabellen werden in der Spalte C1 partitioniert.

Die Verknüpfung mit DIM2 mit dem Vergleichselement WHERE DIM2.C2 = FACT.C2 kann nicht zusammengefasst werden, da FACT in der Spalte C1 partitioniert wird, nicht in der Spalte C2.

In diesem Fall sollte DIM2 in der Knotengruppe der Fakttable repliziert werden. Dadurch kann die Verknüpfung lokal in jeder Partition erstellt werden.

Anmerkung: Die vorliegende Erörterung von replizierten Übersichtstabellen hängt mit der datenbankübergreifenden Replikation zusammen. Datenbankinterne Replikation umfasst Subskriptionen, Steuertabellen und Daten in unterschiedlichen Datenbanken und auf unterschiedlichen Betriebssystemen. Wenn Sie sich für datenbankinterne Replikation interessieren, finden Sie weitere Informationen in *Replikation Benutzer- und Referenzhandbuch*.

Die Quelltable beim Erstellen einer replizierten Übersichtstabelle kann aus einer Knotengruppe mit einem oder mit mehreren Knoten stammen. Meistens handelt es sich um eine kleine Tabelle, die in eine Knotengruppe mit einem Knoten versetzt werden kann. Sie können die Datenmenge für die Replikation einschränken, indem Sie nur einen Teil der Spalten der Tabelle angeben, die Zeilenzahl mit Hilfe der verwendeten Vergleichselemente begrenzen oder beide Methoden anwenden, wenn Sie die replizierte Übersichtstabelle erstellen.

Anmerkung: Für das Funktionieren replizierter Übersichtstabellen ist die Option zur Datenerfassung nicht erforderlich.

Die replizierte Übersichtstabelle könnte auch in einer Knotengruppe mit mehreren Knoten erstellt werden. Dies ist die Knotengruppe, in die Sie Ihre großen Tabellen versetzt haben. In diesem Fall werden in allen Partitionen der Knotengruppe Kopien der Quellentabelle erstellt. Es ist besser, Verknüpfungen zwischen einer großen Fakttable und den Dimensionstabellen in dieser Umgebung lokal zu erstellen, als die Quellentabelle in alle Partitionen verteilen zu müssen. Indizes für replizierte Tabellen werden nicht automatisch erstellt. Indizes werden erstellt und können sich von den in der Quellentabelle erstellten Indizes unterscheiden.

Anmerkung: Sie können keine eindeutigen Indizes für replizierte Tabellen erstellen (und keine Integritätsbedingungen festlegen). Dadurch werden ungültige Integritätsbedingungen vermieden, die in den Quellentabellen nicht vorhanden sind. Diese Integritätsbedingungen werden unterbunden, selbst wenn die Quellentabelle eine identische Integritätsbedingung enthält.

Nach der Verwendung der Anweisung REFRESH sollten Sie in der replizierten Tabelle RUNSTATS wie in anderen Tabellen ausführen.

Sie können auf die replizierten Tabellen direkt in einer Abfrage verweisen. Sie können jedoch bei einer replizierten Tabelle nicht das Vergleichselement NODENUMBER() verwenden, um die Tabellendaten in einer bestimmten Partition anzuzeigen. Verwenden Sie die Funktion EXPLAIN, um zu prüfen, ob eine erstellte replizierte Übersichtstabelle benutzt wurde (vorausgesetzt, eine Abfrage hat auf die Quellentabelle verwiesen). Stellen Sie zunächst sicher, dass die EXPLAIN-Tabellen vorhanden sind. Erstellen Sie anschließend einen EXPLAIN-Plan für die Anweisung SELECT, für die Sie sich interessieren. Formatieren Sie schließlich die EXPLAIN-Ausgabe mit dem Dienstprogramm db2exfmt.

Ob der vom Optimierungsprogramm ausgewählte Zugriffsplan die replizierte Übersichtstabelle tatsächlich verwendet, hängt von den zu verknüpfenden Informationen ab. Die replizierte Übersichtstabelle könnte nicht verwendet werden, wenn das Optimierungsprogramm feststellt, dass es weniger aufwendig wäre, die ursprüngliche Quellentabelle in die anderen Partitionen der Knotengruppe zu kopieren.

Datenzugriffskonzepte und Optimierung

Der Abschnitt „Zugriff über mehrere Indizes“ unter „Konzepte der Indexsuche“ wurde geändert.

Fügen Sie vor der Anmerkung am Abschnittsende folgende Informationen ein:

Zur Umsetzung der Leistungsvorteile dynamischer Bitzuordnungen beim Durchsuchen mehrerer Indizes müssen Sie möglicherweise den Datenbankkonfigurationsparameter „Zwischenspeicher für Sortierlisten“ (*sortheap*) und den Konfigurationsparameter des Datenbankmanagers für den Schwellenwert des Sortierspeichers (*sheapthres*) ändern. Bei der Verwendung von dynamischen Bitzuordnungen in Zugriffsplänen ist zusätzlicher Sortierspeicherbereich erforderlich. Wenn *sheapthres* auf einen relativ ähnlichen Wert wie *sortheap* gesetzt wird (d. h. weniger als Faktor 2 oder 3 pro gleichzeitiger Abfrage), müssen dynamische Bitzuordnungen mit Zugriff über mehrere Indizes mit erheblich weniger Speicherkapazität arbeiten als vom Optimierungsprogramm geplant.

Sie können dieses Problem lösen, indem Sie den Wert von *sheapthres* im Verhältnis zu *sortheap* erhöhen.

Der Abschnitt „Suchstrategien für eine Sternverknüpfung (Star Join)“ unter „Vergleichselementterminologie“ wurde geändert.

Fügen Sie am Abschnittsende folgende Informationen hinzu:

Die im Rahmen der Sternverknüpfung erstellten und verwendeten dynamischen Bitzuordnungen verwenden Sortierspeicher. Weitere Informationen zum Datenbankkonfigurationsparameter „Zwischenspeicher für Sortierlisten“ (*sortheap*) finden Sie in Kapitel 13, „Konfigurieren von DB2“, des Handbuchs *Systemverwaltung: Optimierung*.

Kapitel 8. Leistung bei der Ausführung

Verwalten des Datenbankpufferpools

Fügen Sie im Abschnitt „Verwalten des Datenbankpufferpools“ folgende Informationen nach dem Absatz ein, der wie folgt beginnt: „Beim Erstellen eines Pufferpools wird standardmäßig eine Seitengröße von 4 KB verwendet.“:

Beim Arbeiten mit Windows 2000 werden Pufferpoolgrößen bis zu 64 GB abzüglich der Größe von DB2 und des Betriebssystems unterstützt. (Hier wird davon ausgegangen, dass DB2 das Hauptprodukt auf dem System ist.) Diese Unterstützung ist verfügbar über Microsoft Address Windowing Extensions (AWE).

AWE kann zwar mit Pufferpools jeder Größe verwendet werden, doch gibt es auch andere empfohlene Windows-Produkte, wenn Sie AWE-Nutzung bei größeren Pufferpools benötigen. Windows 2000 Advanced Server bietet Unterstützung für bis zu 8 GB Hauptspeicher. Windows 2000 Data Center Server bietet Unterstützung für bis zu 64 GB Hauptspeicher.

DB2 und Windows 2000 müssen korrekt konfiguriert sein, um AWE-Pufferpools unterstützen zu können. Der Pufferpool, der AWE nutzt, muss in der Datenbank vorhanden sein.

Wenn Sie einen 3 GB großen Benutzeradressbereich zuordnen wollen, verwenden Sie die Boot-Option „/3GB“ von Windows 2000. Dadurch kann eine größere AWE-Fenstergröße verwendet werden. Wenn Sie Zugriff auf mehr als 4 GB Hauptspeicher über die AWE-Speicherschnittstelle ermöglichen wollen, verwenden Sie die Boot-Option „/PAE“ von Windows 2000. Wählen Sie in der Systemsteuerung „System“ und „Starten und Wiederherstellen“ aus. In der Dropdown-Liste können Sie die verfügbaren Boot-Optionen sehen. Wenn die gewünschte Boot-Option (/3GB oder /PAE) ausgewählt ist, können Sie mit dem nächsten Schritt für die Einrichtung der AWE-Unterstützung fortfahren. Wenn die gewünschte Option nicht auswählbar ist, müssen Sie die Option der Datei `boot.ini` im Systemlaufwerk hinzufügen. Die Datei `boot.ini` enthält eine Liste von Aktionen, die beim Start des Betriebssystems ausgeführt werden. Fügen Sie „/3GB“ und/oder „/PAE“ (getrennt durch Leerzeichen) am Ende der Liste der bereits vorhandenen Parameter hinzu. Sobald Sie diese geänderte Datei gespeichert haben, können Sie die korrekte Boot-Option wie oben erläutert prüfen und auswählen.

Windows 2000 muss auch geändert werden, um das Recht zum Sperren von Seiten im Speicher dem Benutzer zuzuordnen, unter dem DB2 installiert ist. Zur Einrichtung dieses Rechts melden Sie sich in Windows 2000 als der Benutzer an, der DB2 installiert hat, und wählen Sie im Menü „Start“ den Ordner „Verwaltung“ und dort das Programm „Lokale Sicherheitsrichtlinie“ aus. Dort können Sie die Zuordnung der Benutzerrechte für das Recht zum Sperren von Seiten im Speicher auswählen.

DB2 setzt voraus, dass die Registrierungsvariable `DB2_AWE` gesetzt ist. Zur korrekten Einstellung dieser Variablen müssen Sie die ID des Pufferpools kennen, für den Sie die AWE-Unterstützung zulassen wollen. Sie müssen auch die Anzahl der physischen Seiten und der Adressfensterseiten kennen, die Sie zuordnen wollen. Die Anzahl physischer Seiten, die zugeordnet werden sollen, sollte unter der Anzahl der insgesamt verfügbaren physischen Seiten liegen. Die tatsächlich ausgewählte Anzahl hängt von Ihrer Arbeitsumgebung ab. Wenn Sie z. B. eine Umgebung haben, bei der nur DB2 und Datenbankanwendungen auf Ihrem System verwendet werden, können Sie ein halbes bis ein GB weniger als die Gesamtgröße des

physischen Seiten als Wert für die Variable DB2_AWE verwenden. Bei einer Umgebung, in der auch andere Anwendungen das System nutzen, müssen Sie einen höheren Wert von der Gesamtanzahl abziehen, um mehr physische Seiten für diese anderen Anwendungen zu reservieren. Die in der Registrierungsvariablen DB2_AWE verwendete Anzahl ist die Anzahl der physischen Seiten, die zur Unterstützung von AWE und zur Verwendung durch DB2 verwendet werden. Die Obergrenze für Adressfensterseiten beträgt 1,5 GB oder 2,5 GB, wenn die Boot-Option /3GB von Windows 2000 aktiv ist.

Informationen zum Setzen der DB2-Registrierungsvariablen DB2_AWE finden Sie in der Tabelle der neuen und geänderten Registrierungsvariablen in „Anhang A. DB2-Registrierungsvariablen und DB2-Umgebungsvariablen“ weiter unten in diesem Abschnitt.

Verwalten mehrerer Datenbankpufferpools

Fügen Sie im Abschnitt „Verwalten mehrerer Datenbankpufferpools“ den folgenden Absatz nach dem Absatz hinzu, der wie folgt beginnt: „Bei der Arbeit an Ihrem Datenbankentwurf haben Sie möglicherweise festgestellt, dass Tabellen mit 8-KB-Seiten am besten geeignet waren.“:

Beim Arbeiten mit Windows 2000 kann die Registrierungsvariable DB2_AWE verwendet werden, um die Einstellungen für die Pufferpoolgröße in den Katalog- und Konfigurationsdateien außer Kraft zu setzen. Die Verwendung dieser Registrierungsvariablen ermöglicht Pufferpoolgrößen bis ca. 64 GB.

Ersetzen Sie im selben Abschnitt den Absatz direkt vor der Anmerkung durch folgenden Text:

Der Datenbankmanager startet mit Minimalwerten, um es Ihnen zu ermöglichen, eine Verbindung zu Datenbank herzustellen. Sie können dann die Pufferpoolgrößen rekonfigurieren oder andere kritische Aufgaben durchführen, um die Datenbank mit den korrekten Pufferpoolgrößen erneut zu starten. Sie sollten die Datenbank nicht längere Zeit in diesem Zustand betreiben.

Im Abschnitt zur Reorganisation von Katalogen und Benutzertabellen muss der letzte Satz (mit einer kurzen Liste) des Absatzes, der mit „Das Dienstprogramm ermöglicht die Angabe eines temporären Tabellenbereichs,...“ beginnt, durch Folgendes ersetzt werden :

Bei der Verwendung desselben Tabellenbereichs läuft die Reorganisation von Tabellen schneller ab. Es tritt jedoch mehr Protokollierung auf, und für die reorganisierte Tabelle muss ausreichend Speicherplatz zur Verfügung stehen. Wenn Sie einen temporären Tabellenbereich angeben, empfiehlt es sich in der Regel, einen temporären SMS-Tabellenbereich anzugeben. Ein temporärer DMS-Tabellenbereich ist nicht zu empfehlen, da Sie bei dieser Art von Tabellenbereich jeweils nur eine REORG-Organisation ablaufen lassen können.

Fügen Sie im Abschnitt „Erweitern von Speicher“ den folgenden Absatz nach dem dritten Absatz des Abschnitts hinzu:

Bei der Zuordnung des AWE-Pufferpools (Address Windowing Extensions) von Windows 2000 mit der Registrierungsvariablen DB2_AWE kann der erweiterte Speichercache nicht verwendet werden.

Kapitel 9. Verwenden von Governor

Im Abschnitt zum Erstellen der Governor-Konfigurationsdatei muss der erste Satz im ersten Absatz nach der Besprechung der Zeitplanungsaktion durch folgenden Satz ersetzt werden:

Wenn mehr als eine Regel für eine Anwendung gilt, werden alle Regeln angewendet. Je nach Regel und den eingestellten Begrenzungen wird die Aktion zu der zuerst gefundenen Regelbegrenzung als erste angewendet.

Kapitel 13. Konfigurieren von DB2

Folgende Parameter müssen geändert werden:

Zwischenspeicher für Sortierlisten (sortheap)

Der Abschnitt „Empfehlung“ wurde geändert. Die dortigen Informationen sollten wie folgt lauten:

Beachten Sie Folgendes beim Arbeiten mit dem Sortierspeicher:

- Geeignete Indizes können die Verwendung des Sortierspeichers minimieren.
- Hash-Verknüpfungspuffer und dynamische Bitzuordnungen (zum logischen Verknüpfen von Indizes über AND und für Sternverknüpfungen verwendet) verwenden Sortierspeicher. Erhöhen Sie den Wert dieser Parameter, wenn diese Methoden verwendet werden.
- Erhöhen Sie den Wert dieses Parameters, wenn häufig umfangreiche Sortiervorgänge ausgeführt werden müssen.
- ... (die übrigen Punkte bleiben unverändert)

Schwellenwert für Sortierspeicher (sheapthres)

Der vorletzte Absatz in der Beschreibung dieses Parameters wurde geändert. Der Absatz sollte nun wie folgt lauten:

Bei folgenden Operationen wird der Sortierspeicher beispielsweise verwendet: Sortiervorgänge, dynamische Bitzuordnungen (zum logischen Verknüpfen von Indizes über AND und für Sternverknüpfungen verwendet) und Operationen, bei denen sich die Tabelle im Speicher befindet.

Folgende Informationen müssen der Beschreibung dieses Parameters hinzugefügt werden:

Es gibt keinen Grund, den Wert dieses Parameter beim Versetzen von einer Umgebung mit einem Knoten in eine Umgebung mit mehreren Knoten zu erhöhen. Wenn Sie die Konfigurationsparameter der Datenbank und des Datenbankmanagers in einer Umgebung mit einem Knoten (DB2 EE) optimiert haben, funktionieren diese Werte mit einer Umgebung mit mehreren Knoten (DB2 EEE) zumeist ebenfalls einwandfrei.

Der Parameter **Schwellenwert für Sortierspeicher**, ein Konfigurationsparameter des Datenbankmanagers, gilt für das gesamte DB2-Exemplar. Sie können diesen Parameter in anderen Knoten oder Partitionen nur durch die Erstellung mehrerer DB2-Exemplare auf unterschiedliche Werte setzen. Dies erfordert die Verwaltung verschiedener DB2-Datenbanken in verschiedenen Knotengruppen. Ein solches Vorgehen macht viele Vorteile einer Umgebung mit partitionierten Datenbanken zunichte.

Maximale Anzahl Sperren pro Anwendung (maxlocks)

Die folgende Änderung bezieht sich auf den Abschnitt „Empfehlung“ für den Datenbankkonfigurationsparameter „Maximale Anzahl Sperren pro Anwendung“ (maxlocks).

Empfehlung: Mit Hilfe der folgenden Formel können Sie *maxlocks* so definieren, dass eine Anwendung die doppelte durchschnittliche Anzahl Sperren anhalten kann:

$$\text{maxlocks} = 2 * 100 / \text{maxappls}$$

2 legt die doppelte durchschnittliche Anzahl fest, und 100 stellt den höchsten zulässigen Prozentsatz dar. Wenn nur wenige Anwendungen gleichzeitig ausgeführt werden, können Sie die folgende Formel als Alternative zur ersten verwenden:

$$\text{maxlocks} = 2 * 100 / (\text{durchschnittliche Anzahl gleichzeitig aktiver Anwendungen})$$

Bei der Definition von *maxlocks* muss u. a. die Verwendung in Verbindung mit der Größe der Sperrenliste (*locklist*) berücksichtigt werden. Der tatsächliche Grenzwert für die Anzahl Sperren, die eine Anwendung anhält, bevor eine Sperreneskalation auftritt, lautet:

$$\text{maxlocks} * \text{locklist} * 4096 / (100 * 36)$$

Dabei gilt Folgendes: 4096 ist die Anzahl Byte auf einer Seite, 100 ist der höchste zulässige Prozentsatz für *maxlocks* und 36 ist die Anzahl Byte pro Sperre. Wenn Sie wissen, dass eine Ihrer Anwendungen 1000 Sperren benötigt, und wenn keine Sperreneskalation auftreten soll, müssen Sie die Werte für *maxlocks* und *locklist* in dieser Formel so wählen, dass das Ergebnis größer ist als 1000. (Wird in dieser Formel 10 für *maxlocks* und 100 für *locklist* angegeben, ist das Ergebnis größer als die 1000 benötigten Sperren.)

Ist der Wert für *maxlocks* zu niedrig, tritt eine Sperreneskalation ein, wenn noch genügend Sperrbereich für andere gleichzeitig aktive Anwendungen vorhanden ist. Ist der Wert für *maxlocks* zu hoch, können wenige Anwendungen den meisten Sperrbereich verbrauchen, und andere Anwendungen müssen Sperreneskalation ausführen. Die erforderliche Sperreneskalation führt in diesem Fall zu schlechter gleichzeitiger Ausführung.

Diesen Konfigurationsparameter können Sie mit Hilfe des Datenbanksystemmonitors verfolgen und optimieren.

Konfigurieren von DB2/DB2 Data Links Manager/Data Links-Token-Verfallsintervall (dl_expint)

Im Gegensatz zu den Angaben in der Dokumentation verfällt das Zugriffssteuerungs-Token, wenn für *dl_expint* „-1“ definiert wird. Eine Lösungsmöglichkeit ist, für *dl_expint* den Maximalwert 31536000 (Sekunden) festzulegen. Dies entspricht einer Verfallszeit von einem Jahr, was für alle Anwendungen adäquat sein sollte.

Datenbankkonfigurationsparameter MIN_DEC_DIV_3

Der Datenbankkonfigurationsparameter MIN_DEC_DIV_3 dient dazu, eine Berechnungsänderung des Maßstabsfaktors für Dezimaldivision in SQL schnell zu ermöglichen. Für MIN_DEC_DIV_3 kann YES oder NO angegeben werden. Der Standardwert für MIN_DEC_DIV_3 ist NO.

Der Datenbankkonfigurationsparameter MIN_DEC_DIV_3 ändert den Ergebnismaßstabsfaktor einer dezimalen Divisionsrechenoperation. Lautet der Wert NO, wird der Maßstabsfaktor wie folgt berechnet: $31-p+s-s'$. Weitere Informationen finden Sie im Handbuch *SQL Reference*, Kapitel 3, „Decimal Arithmetic in SQL“. Lautet der Wert YES, wird der Maßstabsfaktor wie folgt berechnet: $\text{MAX}(3, 31-p+s-s')$. Hierdurch hat das Ergebnis einer Dezimaldivision immer einen Maßstabsfaktor von mindestens 3. Die Genauigkeit ist immer 31.

Eine Änderung des Datenbankkonfigurationsparameters kann Änderungen in Anwendungen für vorhandene Datenbanken verursachen. Dies kann der Fall sein, wenn der Ergebnismaßstabsfaktor für Dezimaldivisionen durch eine Änderung dieses Datenbankkonfigurationsparameters beeinflusst werden würde. Unten sind einige Szenarios aufgeführt, die sich auf Anwendungen auswirken können. Diese Szenarios sollten Sie berücksichtigen, bevor MIN_DEC_DIV_3 auf einem Datenbank-Server mit vorhandenen Datenbanken geändert wird.

- Wird der Ergebnismaßstabsfaktor einer der Anzeigenspalten geändert, könnte eine Anzeige, die in einer Umgebung mit einer Einstellung definiert ist, mit SQLCODE-Wert -344 fehlschlagen, wenn nach Änderung des Datenbankkonfigurationsparameters auf sie verwiesen wird. Die Nachricht SQL0344N bezieht sich auf rekursive allgemeine Tabellenausdrücke. Ist der Objektname (erstes Token) jedoch eine Sicht, müssen Sie die Anzeige löschen und erneut erstellen, um diesen Fehler zu vermeiden.
- Ein statisches Paket ändert sein Verhalten erst, wenn es erneut gebunden wird (implizit oder explizit). Wird der Wert NO zum Beispiel durch YES ersetzt, werden die zusätzlichen Maßstabsfaktorziffern erst in den Ergebnissen berücksichtigt, wenn eine erneute Bindeoperation erfolgt. Für alle geänderten statischen Pakete kann eine erneute Bindeoperation durch einen expliziten Befehl für erneutes Binden erzwungen werden.
- Eine Prüfung auf Integritätsbedingung mit Dezimaldivision kann einige Werte einschränken, die zuvor akzeptiert wurden. Diese Zeilen machen die Integritätsbedingung jetzt ungültig, werden jedoch erst erkannt, wenn die eine Spalte, die zur Zeile für die Prüfung auf Integritätsbedingung gehört, aktualisiert wird oder wenn der Befehl **SET INTEGRITY** mit der Option **IMMEDIATE CHECKED** verarbeitet wird. Damit die Prüfung auf eine solche Integritätsbedingung erzwungen wird, müssen Sie einen Befehl **ALTER TABLE** ausführen, um die Prüfung auf Integritätsbedingung zu löschen, und dann einen Befehl **ALTER TABLE** ausführen, um die Integritätsbedingung wieder hinzuzufügen.

Anmerkung: DB2 Version 7 hat außerdem folgende Einschränkungen:

1. Der Befehl **GET DB CFG FOR DBNAME** zeigt die Einstellung für MIN_DEC_DIV_3 nicht an. Die aktuelle Einstellung können Sie am besten ermitteln, indem Sie die Nebenwirkung eines Dezimaldivisionsergebnisses beobachten. Nehmen wir die folgende Anweisung:

```
VALUES (DEC(1,31,0)/DEC(1,31,5))
```

Gibt diese Anweisung den SQLCODE-Wert SQL0419N aus, verfügt die Datenbank nicht über MIN_DEC_DIV_3-Unterstützung

oder MIN_DEC_DIV_3 wurde auf OFF gesetzt. Lautet der Rückgabewert der Anweisung 1.000, ist ON für MIN_DEC_DIV_3 definiert.

2. MIN_DEC_DIV_3 erscheint nicht in der Liste der Konfigurationsschlüsselwörter, wenn Sie den folgenden Befehl ausführen: ?
`UPDATE DB CFG`

Maximaler Zwischenspeicher für Anwendungssteuerung (app_ctl_heap_sz)

Der Text für diesen Parameter sollte jetzt folgendermaßen lauten:

Bei partitionierten Datenbanken und nicht partitionierten Datenbanken mit aktivierter Intraparallelität (intra_parallel=ON) ist dies die Größe des gemeinsam benutzten Speicherbereichs, der für den Zwischenspeicher für die Anwendungssteuerung zugeordnet ist. Bei nicht partitionierten Datenbanken, bei denen Intraparallelität inaktiviert ist (intra_parallel=OFF), ist dies der maximale private Speicher, der für den Zwischenspeicher zugeordnet wird. Pro Verbindung und Partition gibt es einen Zwischenspeicher für die Anwendungssteuerung.

Der Zwischenspeicher für die Anwendungssteuerung ist hauptsächlich erforderlich für die gemeinsame Benutzung von Informationen zwischen Agenten, die dieselbe Anforderung bearbeiten, sowie in einer Umgebung mit partitionierten Datenbanken zum Speichern von ausführbaren Abschnitten, die SQL-Anweisungen darstellen. Die Nutzung dieses Zwischenspeichers ist minimal bei nicht partitionierten Datenbanken, wenn Abfragen mit einem Grad der Parallelität von maximal 1 ausgeführt werden.

Dieser Zwischenspeicher wird auch zum Speichern von Deskriptorinformationen für deklarierte temporäre Tabellen verwendet. Die Deskriptorinformationen für alle deklarierten temporären Tabellen, die nicht explizit gelöscht wurden, werden in diesem Zwischenspeicher gespeichert und können erst gelöscht werden, nachdem die deklarierte temporäre Tabelle gelöscht wurde.

Der Abschnitt „Empfehlung<“ wurde nicht geändert.

Zwischenspeicher für den Datenbanksystemmonitor (mon_heap_sz)

Die Standardeinstellung für den OS/2- und Windows NT-Datenbank-Server mit lokalen und fernen Clients und den Satellitendatenbank-Server mit lokalen Clients wurde von 24 auf 32 geändert. Der Bereich bleibt unverändert.

Maximale Anzahl aktiver Anwendungen (maxapps)

Die obere Bereichsgrenze für alle Plattformen wurde von 64 000 in 60 000 geändert. Der Standardwert bleibt unverändert.

Vor dem bedingten Prüfpunkt zu schreibende Protokollsätze (softmax)

Die Maßeinheit wird in den Prozentsatz der Größe einer primären Protokolldatei geändert.

Geänderte Seiten protokollieren (trackmod)

Konfigurationsart: Datenbank

Parameterart: Konfigurierbar

Standardwert [Bereich]: Off [On; Off]

Wenn dieser Parameter auf 0N gesetzt wird, verfolgt der Datenbankmanager, welche Seiten in der Datenbank seit der letzten Gesamtsicherung geändert wurden. Dadurch kann das Sicherungsprogramm feststellen, welche Seiten in einer Teilsicherung enthalten sein sollen, ohne jede Seite einzeln zu untersuchen. Bei SMS-Tabellenbereichen erfolgt diese Verfolgung auf Tabellenbereichsebene. Bei DMS-Tabellenbereichen erfolgt die Verfolgung auf Speicherbereichsebene für Daten und Indexseiten und auf Tabellenbereichsebene für andere Seitentypen. Nach der Einstellung dieses Parameters auf 0N müssen Sie eine Gesamtsicherung durchführen, damit Sie eine Ausgangsbasis haben, auf der Teilsicherungen aufsetzen können.

Datenbankprotokollpfad ändern (newlogpath)

Konfigurationsart: Datenbank

Parameterart: Konfigurierbar

Standardwert [Bereich]: Null [gültiger Pfad oder gültige Einheit]

Zugehörige Parameter: Pfad zu Protokolldateien (logpath); Datenbank ist konsistent (database_consistent)

Mit diesem Parameter können Sie eine Zeichenfolge mit maximal 242 Byte angeben, um die Speicherposition der Protokolldateien zu ändern. Die Zeichenfolge kann auf einen Pfadnamen oder eine unformatierte Einheit zeigen. Wenn sie auf einen Pfadnamen zeigt, darf dies kein relativer Pfadname, sondern es muss sich um einen vollständig qualifizierten Pfadnamen handeln.

Anmerkung: In einer Umgebung mit partitionierten Datenbanken wird die Knotennummer automatisch an den Pfad angehängt, um die Eindeutigkeit des Pfads in Konfigurationen mit mehreren logischen Knoten aufrechtzuerhalten.

Zur Angabe einer Einheit geben Sie eine Zeichenfolge an, die das Betriebssystem als Einheit identifiziert. Unter Windows NT z. B.:

\\.\d: oder \\.\PhysicalDisk5

Anmerkung: Windows NT Version 4.0 muss mit Service Pack 3 installiert sein, damit Sie Protokolle auf eine Einheit schreiben können.

Auf UNIX-basierten Plattformen:

/dev/rdblog8

Anmerkung: Sie können eine Einheit nur auf AIX-, Windows 2000-, Windows NT-, Solaris-, HP-UX-, NUMA-Q- und Linux-Plattformen angeben.

Die neue Einstellung wird erst zum LOGPATH-Wert, wenn beide der folgenden Bedingungen erfüllt sind:

- Die Datenbank ist konsistent (siehe Parameter „database_consistent“).

- Alle Benutzer sind von der Datenbank getrennt.

Wenn die erste neue Verbindung zur Datenbank hergestellt wird, verschiebt der Datenbankmanager die Protokolle an die von „logpath“ bezeichnete Position.

Im alten Pfad können noch Protokolldateien vorhanden sein, die möglicherweise noch nicht archiviert sind. Sie müssen diese Protokolldateien dann manuell archivieren. Wenn Sie in dieser Datenbank eine Replikation ausführen, kann es auch möglich sein, dass die Replikation die Protokolldateien mit dem Stand vor der Protokollpfadänderung benötigt. Wenn die Datenbank mit dem Datenbank-Konfigurationsparameter „userexit=Yes“ für aktivierten Benutzer-Exit konfiguriert ist und alle Protokolldateien entweder automatisch von DB2 oder manuell von Ihnen archiviert wurden, kann DB2 die Protokolldateien abrufen, um die Replikation abzuschließen. Oder Sie können Sie Dateien aus dem alten Protokollpfad in den neuen Protokollpfad kopieren.

Empfehlung: Idealerweise befinden sich die Protokolldateien auf einer physischen Platte, auf der keine hohe Ein-/Ausgabe auftritt. Vermeiden Sie beispielsweise, die Protokolle auf derselben Platte zu speichern wie das Betriebssystem oder große Datenbanken. Dies erlaubt eine effiziente Protokollierung mit minimalem Systemaufwand, z. B. Warten auf E/A.

Sie können mit Hilfe des Datenbanksystemmonitors die Anzahl der E/A-Operationen für die Datenbankprotokollierung verfolgen.

Weitere Informationen finden Sie in den folgenden Monitorelementbeschreibungen im Handbuch „System Monitor Guide and Reference“:

- log_reads (Anzahl gelesener Protokollseiten)
- log_writes (Anzahl geschriebener Protokollseiten)

Diese Datenelemente liefern den Umfang der E/A-Aktivität bezüglich der Datenbankprotokollierung. Sie können ein Überwachungsprogramm des Betriebssystems verwenden, um Informationen zu anderen E/A-Plattenaktivitäten zu sammeln und dann die beiden Arten von E/A-Aktivität vergleichen.

Pfad zu Protokolldateien (logpath)

Konfigurationsart: Datenbank

Parameterart: Informativ

Zugehörige Parameter: Datenbankprotokollpfad ändern (newlogpath)

Dieser Parameter enthält den aktuellen Pfad, der zur Protokollierung verwendet wird. Sie können diesen Parameter nicht direkt ändern, da er vom Datenbankmanager gesetzt wird, nachdem eine Änderung am Parameter „newlogpath“ wirksam wurde. Wenn eine Datenbank erstellt wird, wird die Wiederherstellungsprotokolldatei für sie in einem Unterverzeichnis des Verzeichnisses mit der Datenbank erstellt. Die Standardeinstellung ist ein Unterverzeichnis namens SQLOGDIR unter dem Verzeichnis, das für die Datenbank erstellt wurde.

Maximaler Speicher für Sperrenliste (locklist)

Der Maximalwert wurde von 60 000 auf 524 288 erhöht.

Anhang A. DB2-Registrierungsvariablen und DB2-Umgebungsvariablen

Folgende Registrierungsvariablen sind neu oder müssen geändert werden:

Tabelle der neuen und geänderten Registrierungsvariablen

Tabelle 6. Registrierungsvariablen

Variablenname	Betriebssystem	Werte
Beschreibung		
DB2MAXFSCRSEARCH	Alle	Standardwert=5 Werte: -1, 1 bis 33554
Gibt an, wie viele Steuersätze für freien Speicherbereich beim Hinzufügen eines Eintrags zu einer Tabelle durchsucht werden. Standardmäßig werden fünf Steuersätze für freien Speicherbereich durchsucht. Mit diesem Wert können Sie die Gewichtung zwischen Einfügeschwindigkeit und Speicherwiederverwendung verändern. Mit hohen Werten optimieren Sie die Speicherwiederverwendung. Mit niedrigen Werten optimieren Sie die Einfügeschwindigkeit. Der Wert -1 zwingt den Datenbankmanager, alle Steuersätze für freien Speicherbereich zu durchsuchen.		
DLFM_TSM_MGMTCLASS	AIX, Windows NT, Solaris	Standardwert: die TSM-Standardverwaltungsklasse Werte: alle gültigen TSM-Verwaltungsklassen
Gibt an, welche TSM-Verwaltungsklasse zum Archivieren und Abrufen von verknüpften Dateien verwendet wird. Wird für diese Variable kein Wert festgelegt, wird die TSM-Standardverwaltungsklasse verwendet.		
DB2_CORRELATED_PREDICATES	Alle	Standardwert=YES Werte: YES oder NO
Der Standardwert für diese Variable ist YES. Wenn in einer Verknüpfung eindeutige Indizes in korrelierten Spalten vorhanden sind und diese Registrierungsvariable auf YES gesetzt ist, versucht das Optimierungsprogramm, die Korrelation der Verknüpfungsvergleichselemente zu erkennen und auszugleichen. Ist diese Registrierungsvariable auf YES gesetzt, ermittelt das Optimierungsprogramm mit Hilfe der KEYCARD-Informationen aus den Statistiken des eindeutigen Index die Korrelationsfälle und passt die kombinierten Auswahlmöglichkeiten der korrelierten Vergleichselemente an, wodurch eine genauere Schätzung der Größe und des Aufwands für die Verknüpfung erzielt wird.		
DB2_VI_DEVICE	Windows NT	Standardwert= Null Werte: nic0 oder VINIC
Gibt den symbolischen Namen der Einheit oder des VIP-Exemplars (VIP - Virtual Interface Provider) an, die bzw. das der NIC (Network Interface Card) zugeordnet ist. Unabhängige Hardware-Lieferanten produzieren jeweils eigene NICs. Pro Windows NT-Maschine ist nur jeweils eine einzige NIC zulässig. Mehrere logische Knoten auf derselben physischen Maschine benutzen die gleiche NIC gemeinsam. Der symbolische Name „VINIC“ der Einheit muss in Großbuchstaben eingegeben werden und kann nur mit Synfinity Interconnect verwendet werden. Alle anderen momentan unterstützten Implementierungen verwenden den symbolischen Namen „nic0“ der Einheit.		
DB2_SELECTIVITY	ALL	Standardwert=NO Werte: YES oder NO
Diese Registrierungsvariable legt fest, wo die Klausel SELECTIVITY verwendet werden kann. Ausführliche Informationen zur Klausel SELECTIVITY finden Sie im Abschnitt zu Suchbedingungen und Sprachelementen des Handbuchs <i>SQL Reference</i> .		
Wird für diese Registrierungsvariable YES festgelegt, kann die Klausel SELECTIVITY angegeben werden, wenn das Vergleichselement ein Basisvergleichselement ist, in dem mindestens ein Ausdruck Host-Variablen enthält.		

Table 6. Registrierungsvariablen (Forts.)

Variablenname	Betriebssystem	Werte
Beschreibung		
DB2_UPDATE_PART_KEY	ALL	Standardwert=YES Werte: YES oder NO
Ab FixPak 3 lautet der Standardwert YES. Diese Registrierungsvariable gibt an, ob eine Aktualisierung des Partitionierungsschlüssels zulässig ist.		
DB2_BLOCK_ON_LOG_DISK_FULL	ALL	Standardwert=NO Werte: YES oder NO
Diese DB2-Registrierungsvariable kann so gesetzt werden, dass sie das Generieren von Fehlern wegen voller Platte ("disk full") unterbindet, wenn DB2 keine neue Protokolldatei im aktiven Protokollpfad erstellen kann.		
Stattdessen versucht DB2 in Intervallen von 5 Minuten, die Protokolldatei zu erstellen, bis es erfolgreich ist. Nach jedem Versuch schreibt DB2 eine Nachricht in die Datei db2diag.log. Sie können sich nur dann sicher sein, dass Ihre Anwendung aufgrund einer vollen Protokollplatte blockiert, wenn Sie die Datei db2diag.log prüfen.		
Bis die Protokolldatei erfolgreich erstellt ist, kann keine Benutzeranwendung, die versucht, Tabellendaten zu aktualisieren, Transaktionen festschreiben. Nur-Lese-Abfragen sind möglicherweise nicht direkt betroffen. Wenn jedoch eine Abfrage auf Daten zugreifen muss, die von einer Aktualisierungsanforderung gesperrt sind, oder auf eine Datenseite, die von der aktualisierenden Anwendung im Pufferpool fixiert ist, scheinen auch Nur-Lese-Abfragen zu blockieren.		
DB2_INDEX_2BYTEVARLEN	Alle	Standardwert=NO Werte: YES oder NO
Mit Hilfe dieser Registrierungsvariablen können Spalten mit mehr als 255 Byte als Teil eines Indexschlüssels angegeben werden. Für Indizes, die vor der Aktivierung dieser Registrierungsvariablen erstellt wurden, gilt die Schlüsselgrößenbeschränkung von 255 Byte. Indizes, die nach der Aktivierung dieser Registrierungsvariablen erstellt wurden, verhalten sich wie ein 2-Byte-Index, selbst wenn die Registrierungsvariable wieder inaktiviert wird.		
Einige SQL-Anweisungen sind von Änderungen an dieser Registrierungsdatenbankvariable betroffen, darunter CREATE TABLE, CREATE INDEX und ALTER TABLE. Weitere Informationen zu diesen Anweisungen finden Sie unter den Änderungen, die für das Handbuch <i>SQL Reference</i> dokumentiert sind.		
DB2_FORCE_FCM_BP	AIX	Standardwert=NO Werte: YES oder NO
Diese Registrierungsvariable gibt an, von wo die FCM-Ressourcen (Fast Communications Manager) zugeordnet werden. Die Ressourcen können entweder vom gemeinsam benutzten Speichersegment des Datenbankmanagers oder von einem separaten Segment aus zugeordnet werden. Bei mehreren logischen Knoten auf einer Maschine sollte diese Registrierungsvariable verwendet werden. Auf einem partitionierten Datenbanksystem mit symmetrischem Mehrprozessorbetrieb (SMP) hat die Einstellung dieser Registrierungsvariablen keine Auswirkung auf die Art der Kommunikation. In diesem Fall erfolgt die Kommunikation immer über den gemeinsam benutzten Speicher. Die Variable wirkt sich jedoch auf die Anzahl der gemeinsam benutzten Speichersegmente aus, die DB2 nutzt.		
DB2_AWE	Windows 2000	Standardwert=NULL Werte: <eintrag>[; <eintrag>;...]; dabei ist <eintrag>=<pufferpool-ID>, <anzahl physischer seiten>, <anzahl adressfenster>

Tabella 6. RegistrierungsvARIABLEN (Forts.)

Variablenname	Betriebssystem	Werte
Beschreibung		
<p>Diese Registrierungsvariable erlaubt es DB2 UDB unter Windows 2000, Pufferpools zuzuordnen, die bis zu 64 GB Hauptspeicher verwenden. Windows 2000 muss korrekt konfiguriert sein, um AWE-Pufferpools (Address Windows Extensions) unterstützen zu können. Dies umfasst die Erteilung des Rechts zum Sperren von Seiten im Speicher an den Benutzer unter Windows 2000 und das Setzen dieser Registrierungsvariablen in DB2. Für das Setzen dieser Variablen müssen Sie die Pufferpool-ID kennen, die für die AWE-Unterstützung verwendet wird. Sie müssen auch die Anzahl physischer Seiten, die zugeordnet werden sollen, und der Adressfensterseiten ermitteln.</p> <p>Informationen zur Ermittlung der zuzuordnenden Anzahl physischer Seiten und der Anzahl der Adressfenster finden Sie im Abschnitt „Verwalten des Datenbankpufferpools“ unter „Kapitel 8. Leistung bei der Ausführung“ in diesem Abschnitt.</p> <p>Anmerkung: Wenn AWE-Unterstützung aktiviert ist, kann der erweiterte Hauptspeicher (ESTORE) für keinen der Pufferpools in der Datenbank verwendet werden. Die Pufferpools, auf die diese Variable verweist, müssen bereits in SYSIBM.SYSBUFFERPOOLS vorhanden sein.</p>		
DB2_STPROC_LOCKUP_FIRST	Alle	Standardwert=NO Werte: YES oder NO
<p>Diese Registrierungsvariable wurde umbenannt. Sie hieß bisher DB2_DARI_LOOKUP_ALL.</p>		
DB2MEMDISCLAIM	AIX	Standardwert=YES Werte: YES oder NO
<p>Bei AIX können dem von DB2-Prozessen verwendeten Speicher Paging-Bereiche zugeordnet sein. Dieser Paging-Bereich kann reserviert bleiben, selbst wenn der zugehörige Speicher freigegeben wurde. Die Reservierung des Paging-Bereichs hängt von den optimierbaren Zuordnungsmaßnahmen für die Verwaltung des virtuellen Speichers des AIX-Systems ab. Diese Registrierungsvariable steuert, ob DB2-Agenten explizit anfordern, dass AIX den reservierten Paging-Bereich vom freigegebenen Hauptspeicher trennt.</p> <p>Die Einstellung „YES“ führt zu geringeren Anforderungen für den Paging-Bereich und möglicherweise zu weniger Plattenaktivität durch Seitenwechsel. Die Einstellung „NO“ führt zu höheren Anforderungen für den Paging-Bereich und möglicherweise zu mehr Plattenaktivität durch Seitenwechsel. In einigen Fällen, z. B. wenn ausreichend Paging-Bereich zur Verfügung steht und wenn Realspeicher so groß ist, dass kein Seitenwechsel auftritt, bietet die Einstellung NO eine geringe Leistungsverbesserung.</p>		
DB2MEMMAXFREE	Alle	Standardwert=8 388 608 Byte Werte: 0 bis 2 ³² -1 Byte
<p>Diese Registrierungsvariable steuert den maximalen nicht verwendeten Speicher, der von DB2-Prozessen beibehalten wird.</p>		
DB2_ANTIJOIN	Alle	Standardwert=NO in einer EEE-Umgebung Standardwert=YES in einer Nicht-EEE-Umgebung Werte: YES oder NO
<p>Bei DB2 Universal Database EEE-Umgebungen: Wenn YES angegeben ist, sucht das Optimierungsprogramm nach Möglichkeiten, um NOT EXISTS-Unterabfragen in Antiverknüpfungen umzuwandeln, die effizienter von DB2 verarbeitet werden können. Bei Nicht-EEE-Umgebungen: Wenn NO angegeben ist, schränkt das Optimierungsprogramm die Möglichkeit ein, NOT EXISTS-Unterabfragen in Antiverknüpfungen umzuwandeln.</p>		
NEWLOGPATH2	UNIX	Standardwert=NO Werte: YES oder NO
<p>Dieser Parameter sollte es Ihnen erlauben, anzugeben, ob ein sekundärer Pfad zur Implementierung von doppelter Protokollierung verwendet werden soll. Der zu verwendende Pfad wird durch Anhängen des Zeichens '2' an den aktuellen Wert von 'LOGPATH' generiert.</p>		

DB2DOMAINLIST	Windows NT	Standardwert=Null Werte: mindestens eine gültige Windows NT-Domäne (jeweils durch Komma getrennt)
<p>Diese Variable definiert mindestens eine Windows NT-Domäne. Nur von Benutzern, die zu diesen Domänen gehören, werden Verbindungs- oder Anschlussanforderungen akzeptiert.</p> <p>Diese Registrierungsvariable sollte nur in einer reinen Windows NT-Domänenumgebung mit DB2-Servern und -Clients mit DB2 Universal Database Version 7.1 (oder höher) verwendet werden.</p>		
DB2_LIKE_VARCHAR	Alle	Standardwert=Y,N Werte: Y, N, S, Gleitkommakonstante zwischen 0 und 6,2
<p>Diese Variable steuert die Erfassung und Verwendung von Subelementstatistiken. Dies sind Statistikdaten zum Inhalt von Daten in Spalten, wenn die Daten eine Struktur in der Form einer Reihe von durch Leerzeichen getrennten Teilfeldern oder Subelementen haben.</p> <p>Diese Registrierungsvariable wirkt sich auf die Art und Weise aus, in der das Optimierungsprogramm mit einem Vergleichselement der folgenden Form umgeht:</p> <pre>COLUMN LIKE '%xxxxxx%'</pre> <p>Dabei ist xxxxxx eine beliebige Buchstabenfolge.</p> <p>Die Syntax für die Verwendung dieser Registrierungsvariablen lautet:</p> <pre>db2set DB2_LIKE_VARCHAR=[Y N S num1] [,Y N S num2]</pre> <p>Dabei gilt Folgendes:</p> <ul style="list-style-type: none"> • Der Ausdruck vor dem Komma bzw. der einzige Ausdruck rechts des Vergleichselements hat folgende Bedeutung, allerdings nur für Spalten, die keine positive Subelementstatistiken haben: <ul style="list-style-type: none"> – S – Das Optimierungsprogramm schätzt die Länge jedes Elements in einer Reihe von Elementen, die zu einer Spalte verkettet sind, basierend auf der Länge der Zeichenfolge, die in die Zeichen % eingeschlossen ist. – Y – Standardwert. Verwendet einen Standardwert von 1,9 für den Algorithmusparameter. Verwendet einen Subelementalgorithmus variabler Länge mit dem Algorithmusparameter. – N – Verwendet einen Subelementalgorithmus fester Länge. – num1 – Verwendet den Wert von „num1“ als Algorithmusparameter mit dem Subelementalgorithmus variabler Länge. • Der Ausdruck nach dem Komma hat folgende Bedeutung: <ul style="list-style-type: none"> – N – Standardwert. Erfasst keine Subelementstatistiken. – Y – Erfasst Subelementstatistiken. Verwendet einen Subelementalgorithmus variabler Länge, der bei Spalten mit positiven Subelementstatistiken die erfassten statistischen Daten zusammen mit dem Standardwert 1,9 für den Algorithmusparameter verwendet. – num2 – Erfasst Subelementstatistiken. Verwendet einen Subelementalgorithmus variabler Länge, der bei Spalten mit positiven Subelementstatistiken die gesammelten statistischen Daten zusammen mit dem Wert von „num2“ für den Algorithmusparameter verwendet. 		
DB2_PINNED_BP	AIX, HP-UX	Standardwert=NO Werte: YES oder NO

Diese Variable wird auf einigen AIX-Systemen für den globalen Datenbankspeicher (einschließlich Pufferpools) verwendet, der der Datenbank im Hauptspeicher zugeordnet ist. Durch die Erhaltung dieses globalen Datenbankspeichers im Hauptspeicher des Systems wird die Datenbankleistung gleichmäßiger.

Bei einer Auslagerung des Pufferpools aus dem Hauptspeicher des Systems beispielsweise würde sich die Datenbankleistung verschlechtern. Die Verringerung der Platten-E/A durch die Erhaltung der Pufferpools im Systemhauptspeicher verbessert die Datenbankleistung. Wenn Sie andere Anwendungen haben, die mehr Hauptspeicher benötigen, können Sie zulassen, dass der globale Datenbankspeicher je nach den Hauptspeicheranforderungen des Systems aus dem Hauptspeicher ausgelagert wird.

Beim Arbeiten mit HP-UX in einer 64-Bit-Umgebung muss nicht nur diese Registrierungsvariable geändert werden, sondern die DB2-Exemplargruppe muss außerdem das Zugriffsrecht MLOCK erhalten. Dies erreichen Sie, indem ein Benutzer mit Root-Zugriffsrechten folgende Schritte ausführt:

1. Hinzufügen der DB2-Exemplargruppe zur Datei /etc/privgroup. Wenn z. B. die DB2-Exemplargruppe zur Gruppe db2iadm1 gehört, muss der Datei /etc/privgroup die folgende Zeile hinzugefügt werden:
db2iadm1 MLOCK
2. Absetzen des folgenden Befehls:
setprivgrp -f /etc/privgroup

DB2_RR_TO_RS	Alle	Standardwert=NO
		Werte: YES oder NO

Das *Sperren des nächsten Schlüssels* garantiert die Isolationsstufe „Wiederholte Lesen“ (Repeatable Read, RR), indem der nächste Schlüssel für alle INSERT- und DELETE-Anweisungen gesperrt wird und der nächsthöhere Schlüsselwert nach dem Ergebnis für SELECT-Anweisungen festgelegt wird. Bei UPDATE-Anweisungen, die Schlüsselteile des Index ändern, wird der ursprüngliche Indexschlüssel gelöscht und der neue Schlüsselwert eingefügt. Die Sperre des nächsten Schlüssels wird sowohl für das Einfügen wie auch für das Löschen des Schlüssels ausgeführt. Die Sperre des nächsten Schlüssels ist erforderlich, um die Standardfunktion „Wiederholtes Lesen“ von ANSI und SQL92 zu garantieren und ist der Standardwert in DB2.

Wenn Ihre Anwendung anscheinend stoppt oder hängt, sollten Sie die Momentaufnahmeninformationen für Ihre Anwendung überprüfen. Wenn das Problem mit der Sperre des nächsten Schlüssels zusammenhängt, können Sie die Registrierdatenbankvariable DB2_RR_TO_RS basierend auf zwei Bedingungen einschalten. Sie können DB2_RR_TO_RS einschalten wenn keine Ihrer Anwendungen das Verhalten „Wiederholtes Lesen“ verwendet und wenn Suchen nicht-festgeschriebene gelöschte Elemente überspringen können. Das Überspringen hat Auswirkungen auf die Isolationsstufen „Wiederholtes Lesen“, „Lesestabilität“ (Read Stability, RS) und „Cursorstabilität“ (Cursor Stability, CS). (Es gibt keine Zeilensperre für die Isolationsstufe „nicht festgeschriebener Lesevorgang“ (Uncommitted Read, UR).)

Wenn DB2_RR_TO_RS eingeschaltet ist, kann das Verhalten „Wiederholtes Lesen“ für Suchen in den Benutzertabellen nicht garantiert werden, weil die Sperre des nächsten Schlüssels während des Einfügens und Löschens von Indexschlüsseln nicht ausgeführt wird. Katalogtabellen sind von dieser Option nicht betroffen.

Die andere Änderung des Verhaltens ist, dass Suchen bei eingeschaltetem DB2_RR_TO_RS Zeilen überspringen, die gelöscht aber noch nicht festgeschrieben worden sind, obwohl die Zeile für die Suche in Frage gekommen wäre.

Anhang C. EXPLAIN-Programme (SQL)

Im Abschnitt „Ausführen von db2expln und dynexpln“ muss der letzte Absatz wie folgt ersetzt werden:

Zum Ausführen von db2expln müssen Sie über das Zugriffsrecht SELECT für die Systemkatalogsichten sowie über die Ausführungsberechtigung (EXECUTE) für das db2expln-Paket verfügen. Für die Ausführung von dynexpln benötigen Sie die Berechtigung BINDADD für die Datenbank, das zum Verbinden mit der Datenbank verwendete Schema muss vorhanden sein oder Sie müssen über die Berechtigung EXPLICIT_SCHEMA für die Datenbank und über alle Zugriffsrechte verfügen, die für die Bearbeitung der SQL-Anweisungen mit EXPLAIN erforderlich sind. (Wenn Sie über SYSADM- oder DBADM-Berechtigung verfügen, haben Sie automatisch alle erforderlichen Berechtigungsstufen.)

Administering Satellites Guide and Reference

Setting up Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition as Satellites

The sections that follow describe how to set up Windows-based Version 7.2 DB2 Personal Edition and DB2 Workgroup Edition systems so that they can be used as fully functional satellites in a satellite environment. For information about the terms and concepts used in the information that follows, refer to the *Administering Satellites Guide and Reference*. You can find this book at the following URL:

http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/v6pubs.d2w/en_main

For Technotes that supplement the information in the *Administering Satellites Guide and Reference*, refer to the following URL:

<http://www-4.ibm.com/cgi-bin/db2www/data/db2/udb/winos2unix/support/browse.d2w/report?type=tech5udb&tech5udb=Y>

Prerequisites

To set up either DB2 Personal Edition or DB2 Workgroup Edition as satellites, you require the following:

1. A DB2 control server

The DB2 control server is a DB2 Enterprise Edition system that runs on Windows NT or AIX, and has the Control Server component installed. The DB2 Enterprise Edition system that you use must be at Version 6 with FixPak 2 or higher, or Version 7 at any FixPak level.

- If you have a Version 6 Enterprise Edition system that you want to use as the DB2 control server, see „Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System“ auf Seite 132.
- If you are using Version 7 and do not have the Control Server component installed, install this component, re-install any FixPaks that you have already installed, then create the DB2 control server instance and satellite control database. Refer to the *Administering Satellites Guide and Reference* for instructions on creating these objects.

Anmerkung: If you are installing a Version 7.2 Enterprise Edition system on Windows NT for use as the DB2 control server, and you want to perform a response file installation, see the Technote entitled *DB2 Control Server Response File Keywords* for information about the keywords to specify in the response file.

2. The DB2 control server instance and the satellite control database

The DB2 control server instance is typically called DB2CTLSV, and the satellite control database is called SATCTLDB. The DB2 control server instance and the satellite control database are on the Enterprise Edition system, and, on Windows NT, are automatically created when you install DB2 with the Control Server component. If you install DB2 on AIX, see the *Administering Satellites Guide and Reference* for information about creating the DB2 control server instance and the satellite control database.

3. The Satellitenverwaltungszentrale

The Satellitenverwaltungszentrale is the set of GUI tools that you use to set up and administer the satellite environment. You access this set of tools from the

Steuerzentrale. For more information about the Satellitenverwaltungszentrale and the satellite environment, see the *Administering Satellites Guide and Reference*, and the online help that is available from the Satellitenverwaltungszentrale. If you are running a Version 6 Steuerzentrale, see „Upgrading a Version 6 Steuerzentrale and Satellitenverwaltungszentrale“ auf Seite 139.

If you have not already used the Satellitenverwaltungszentrale to set up the satellite environment and to create the object that represents the new satellite in the Satellitenverwaltungszentrale, you should do so before installing the satellite. For more information, see the description of how to set up and test a satellite environment in the *Administering Satellites Guide and Reference*.

4. A Version 7.2 Personal Edition or Workgroup Edition system that you want to use as a satellite.

Installation Considerations

When you install either DB2 Personal Edition or DB2 Workgroup Edition, you do not have to select any special component to enable either system to synchronize. If you intend to perform a response file installation, see „Performing a Response File Installation“ for the keywords that you should specify when installing the Version 7.2 system. If you are performing an interactive installation of your Version 7.2 system, see „Configuring the Version 7.2 System for Synchronization“ auf Seite 131 after you finish installing DB2 for values that you must set at the Version 7.2 system to enable it to synchronize.

Performing a Response File Installation: If you are performing a response file installation of Version 7.2 DB2 Personal Edition or DB2 Workgroup Edition, you can set the following keywords in the response file.

If you decide to not specify one or more of these keywords during the response file installation, see „Configuring the Version 7.2 System for Synchronization“ auf Seite 131 for additional steps that you must perform after installing DB2 to enable the Version 7.2 system to synchronize. You can also use the instructions in this section if you want to change any values that were specified during the response file installation.

db2.db2satelliteid

Sets the satellite ID on the system.

Anmerkung: If you do not specify this keyword, the satellite ID is automatically set to the user ID that was used to install DB2. If you want to use this user ID as the satellite ID, you do not have to specify a value for this keyword.

db2.db2satelliteappver

Sets the application version on the system.

Anmerkung: If you do not specify this keyword, the application version on the satellite is automatically set to V1R0M00. If you want to use this value as the application version, you do not have to specify a value for this keyword.

db2.satctldb_username

Sets the user name to be used for the system to connect to the satellite control database.

db2.satctldb_password

Sets the password that the user name passes to the DB2 control server when the user name connects to the satellite control database.

After you complete the response file installation, the Version 7.2 system is ready to synchronize. You should issue the **db2sync -t** command on the satellite to verify that the values specified on the satellite are correct, and that the satellite can connect to the satellite control database.

For additional information about performing a response file installation, refer to the *Administering Satellites Guide and Reference*.

Anmerkungen:

1. In Version 7, user IDs and passwords are required for the creation of all services on Windows NT and Windows 2000. These user IDs and passwords are specified in the response file by keyword pairs. The first keyword pair found in the response file becomes the default user ID and password for all services, unless you provide an override for a service by specifying the specific keyword pair for that service.

In Version 6, the **admin.userid** and the **admin.password** keywords could be specified during a response file installation of DB2 Satellite Edition to specify the user ID and password that would be used by the Remote Command Service. For Version 7.2 Personal Edition and Workgroup Edition, if you specify these keywords, they are used for the DB2DAS00 instance on the Version 7.2 system. For a DB2 Version 7.2 system, the Remote Command Service will use the user ID and password that is used by the DB2 instance on the system. If you do not specify values for **db2.userid** and **db2.password**, the defaulting rule described above applies.

2. In Version 6, you could create a database when installing DB2 Satellite Edition using a response file installation. You cannot create a database during a response file installation on the Version 7.2 Personal Edition or Workgroup Edition system that you intend to use as a satellite. The following keywords (which are described in the *Administering Satellites Guide and Reference*), are not supported:

- **db2.userdb_name**
- **db2.userdb_recoverable**
- **db2.userdb_rep_src**

Configuring the Version 7.2 System for Synchronization

If you install the Version 7.2 system interactively, several values must be set on the DB2 Personal Edition or DB2 Workgroup Edition system after installing DB2 before the system can synchronize.

Anmerkung: You can use execute an operating system script on the system to set all values at the satellite except for the user ID and password that the satellite uses to connect to the satellite control database (see step 4 auf Seite 132).

1. Set the satellite ID by using the **db2set** command.

If you install DB2 Personal Edition or DB2 Workgroup Edition interactively, the satellite ID is automatically set to the user ID that was used to install DB2. If you want to use this user ID as the satellite ID, you do not have to perform this step. For information about setting the satellite ID, see the *Administering Satellites Guide and Reference*.

2. Set the application version on the satellite by using the **db2sync -s** command.

If you install DB2 Personal Edition or DB2 Workgroup Edition interactively, the application version on the satellite is automatically set to V1R0M00. If you want to use this value as the application version, you do not have to perform this step.

You can use the **db2sync -g** command on the satellite to view the current setting of the application version. If you want to change this value, issue the **db2sync -s** command. You are prompted to provide a new value for the application version. For more information about setting the application version, see the *Administering Satellites Guide and Reference*.

3. Issue the **catalog node** and **catalog database** commands on the satellite to catalog the DB2 control server instance and the satellite control database, SATCTLDB, at the satellite.

You can also use the **db2sync -t** command on the satellite to open the DB2 Synchronizer application in test mode. If the SATCTLDB database is not cataloged at the satellite when you issue the command, the Catalog Control Database window opens. You can either use the DB2 discovery feature that is available from the Catalog Control Database window to catalog the DB2 control server and the SATCTLDB database, or you can type the hostname and server name in this window. You will also be prompted to specify the user ID and password that the satellite will use to connect to the satellite control database, as described in step 4.

Anmerkung: After you install Version 7.2 DB2 Personal Edition or DB2 Workgroup Edition interactively, the DB2 Synchronizer does not start automatically in test mode (as was the case for Version 6 DB2 Satellite Edition).

4. Issue the **db2sync -t** command on the satellite to:
 - Specify the user ID and the password that the satellite will use to connect to the satellite control databaseIf synchronization credentials are not already stored at the satellite, the Connect to Control Database window opens. You must use this window to specify the user ID and password the satellite will use to connect to the satellite control database.
 - Verify the values that are set on the satellite are correct
 - Verify that the satellite can connect to the satellite control database

After you complete these configuration tasks, the Version 7.2 system is ready to synchronize.

Installing FixPak 2 or Higher on a Version 6 Enterprise Edition System

The sections that follow describe the tasks that you must perform to upgrade a Version 6 Enterprise Edition system on Windows NT or AIX for use as a DB2 control server. If you are using a Version 6 Steuerzentrale, also perform the steps in „Upgrading a Version 6 Steuerzentrale and Satellitenverwaltungszentrale“ auf Seite 139 to verify that you have the correct level of the Steuerzentrale and the Satellitenverwaltungszentrale to administer the satellite environment.

Upgrading Version 6 DB2 Enterprise Edition for Use as the DB2 Control Server

For a Version 6 DB2 Enterprise Edition system to be used as the DB2 control server, it must be installed with the Control Server component, and DB2 Enterprise Edition should be at the FixPak 2 service level, or higher. Depending on whether

the DB2 control server component is installed, and the service level of DB2 Enterprise Edition, you will have to perform one of the following tasks:

- Install the DB2 control server component to an existing DB2 Enterprise Edition V6.1 system and install FixPak 2 or higher. Then update the satellite control database (SATCTLDB) on the system.
- Upgrade an already installed DB2 control server to the FixPak 2 level or higher.

Use the information that follows to identify which of the two preceding tasks you need to perform, and the steps that apply to your situation. The following is a summary of the steps that you will perform.

1. First, assess the current state of your DB2 Enterprise Edition installation. You will determine whether the Control Server component is installed, and the service level of DB2.
2. Second, based on the state information that you obtain, you will determine what needs to be done.
3. Third, you will perform the necessary steps to upgrade DB2 Enterprise Edition.

The DB2 control server can only run on DB2 Enterprise Edition for Windows NT and AIX. Continue with the instructions that are appropriate for your platform:

- „Upgrading DB2 Enterprise Edition on Windows NT“
- „Upgrading DB2 Enterprise Edition on AIX“ auf Seite 136

Upgrading DB2 Enterprise Edition on Windows NT: Use the information in the sections that follow to determine the current service level of your Version 6 DB2 Enterprise Edition system, and the steps that you need to perform to update the system to the FixPak 2 service level or higher. You will need to perform the steps of one or more of the following sections:

- „Assessing DB2 Enterprise Edition on Windows NT“
- „Determining What Needs to Be Done“ auf Seite 134
- „Installing the Control Server Component on Windows NT“ auf Seite 134
- „Installing FixPak 2 or Higher on Windows NT“ auf Seite 135
- „Upgrading the SATCTLDB on Windows NT“ auf Seite 135

Assessing DB2 Enterprise Edition on Windows NT: If you have DB2 Enterprise Edition installed on Windows NT, perform the following steps:

1. Check whether the Control Server component is installed. Use the Registry Editor to display the list of installed components:
 - a. Enter `regedit` at a command prompt.
 - b. Under the `HKEY_LOCAL_MACHINE\SOFTWARE\IBM\DB2\Components` registry key, check whether the Control Server is listed. If it is not listed, the control server is not installed.
2. Determine the service level of DB2 Enterprise Edition. Issue the `db2level` command from a command prompt. Use the table that follows to interpret the output:

Values of Key Fields in the db2level output			Your DB2 system is at:
Release	Level	Informational Tokens	
SQL06010	01010104	db2_v6, n990616	Version 6.1 base
SQL06010	01020104	DB2 V6.1.0.1, n990824, WR21136	Version 6.1 plus FixPak 1

SQL06010	01030104	DB2 V6.1.0.6, s991030, WR21163 or DB2 V6.1.0.9, s000101, WR21173	Version 6.1 plus FixPak 2
----------	----------	--	---------------------------

Anmerkung: If the level is greater than 01030104, your system is at a higher FixPak than FixPak 2.

- Record the information that you find, and continue at „Determining What Needs to Be Done“.

Determining What Needs to Be Done: Using the information that you have gathered, find the row in the following table that applies to your situation, and follow the steps that are required to prepare your DB2 Enterprise Edition system to support the DB2 control server at the FixPak 2 level or higher.

Sections that follow the table provide instructions for performing the required steps. Consider checking off each step as you perform it. Only perform the steps that apply to your situation.

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
No	Version 6.1 base, or Version 6.1 plus FixPak 1, or Version 6.1 plus FixPak 2 or higher	Perform the following steps: 1. „Installing the Control Server Component on Windows NT“ 2. „Installing FixPak 2 or Higher on Windows NT“ auf Seite 135 3. „Upgrading the SATCTLDB on Windows NT“ auf Seite 135
Yes	Version 6.1 base, or Version 6.1 plus FixPak 1	Perform the following steps: 1. „Installing FixPak 2 or Higher on Windows NT“ auf Seite 135 2. „Upgrading the SATCTLDB on Windows NT“ auf Seite 135
Yes	Version 6.1, plus FixPak 2 or higher	Perform the following step: 1. „Upgrading the SATCTLDB on Windows NT“ auf Seite 135

Installing the Control Server Component on Windows NT: To install the Control Server component on Windows NT:

- Ensure that all database activity on the system is complete before proceeding.
- Insert the DB2 Universal Database Enterprise Edition Version 6.1 CD in the CD drive.
 If the installation program does not start automatically, run the setup command in the root of the CD to start the installation process.
- When prompted, shut down all the processes that are using DB2.
- On the Welcome window, select **Next**.
- On the Select Products window, ensure that DB2 Enterprise Edition is selected.
- On the Select Installation Type panel, click **Custom**.
- On the Select Components panel, ensure that the Control Server component is selected, and click **Next**.

Anmerkung: If you select other components that are not already installed on your system, these components will be installed too. You cannot alter the drive or directory in which DB2 is installed.

8. On the Configure DB2 Services panels, you can modify the protocol values and the start-up options for the Control Server instance, or take the default values. Either modify the defaults and click **Next**, or click **Next** to use the defaults.
9. Click **Next** on the Start Copy files window to begin the installation process.
10. When the file copying process is complete, you have the option of rebooting your system. You should reboot now. The changes made to the system for the Control Server do not take effect until the system is rebooted.

When the installation process is complete and you have rebooted the system, the satellite control database (SATCTLDB) that was created as part of the Control Server installation must be cataloged in the DB2 instance if you want to use the Steuerzentrale and Satellitenverwaltungszentrale locally on the system. To catalog the SATCTLDB database:

1. Open a DB2 Command Window by selecting **Start>Programs>DB2 for Windows NT>Command Window**
2. Ensure that you are in the db2 instance.
Issue the set command and check the value of db2instance. If the value is not db2, issue the following command:

```
set db2instance=db2
```
3. Catalog the db2ctlsv instance by entering the following command:

```
db2 catalog local node db2ctlsv instance db2ctlsv
```
4. Catalog the SATCTLDB database by entering the following command

```
db2 catalog database satctldb at node db2ctlsv
```
5. Commit the cataloging actions by entering the following command:

```
db2 terminate
```
6. Close the DB2 Command Window.

Installing FixPak 2 or Higher on Windows NT: To upgrade an existing Version 6 DB2 Enterprise Edition system on Windows NT to FixPak 2 or higher, either:

- Download the latest FixPak for DB2 Enterprise Edition for Windows NT V6.1 from the Web, along with its accompanying readme. The FixPak can be downloaded by following the instructions at URL:
<http://www-4.ibm.com/software/data/db2/db2tech/version61.html>

Install the FixPak following the instructions in the readme.txt file.

- Use a DB2 Universal Database, Version 6.1 FixPak for Windows NT CD that is at FixPak 2 level or higher, and follow the instructions in the readme.txt file in the WINNT95 directory on the CD to complete the installation.

Upgrading the SATCTLDB on Windows NT: To upgrade the SATCTLDB database on Windows NT

1. Determine the level of the SATCTLDB database:
 - a. Log on with a user ID that has local administrative authority on the Windows NT system.
 - b. Open a DB2 Command Window by selecting **Start>Programs>DB2 for Windows NT>Command Window**.
 - c. Connect to the SATCTLDB by entering the following command

```
db2 connect to satctl db
```

- d. Determine if the trigger I_BATCHSTEP_TRGSCR exists in the database by issuing the following query:

```
db2 select name from sysibm.sysstriggers where name='I_BATCHSTEP_TRGSCR'
```

Record the number of rows that are returned.

- e. Enter the following command to close the connection to the database:

```
db2 connect reset
```

If step 1d returned one row, the database is at the correct level. In this situation, skip step 2, and continue at step 3. If zero (0) rows are returned, the database is not at the correct level, and must be upgraded, as described in step 2, before you can perform step 3.

2. To upgrade the SATCTLDB database, perform the following steps. Enter all commands in the DB2 Command Window:
 - a. Switch to the directory <db2path>\misc, where <db2path> is the install drive and path, for example c:\sql11ib.
 - b. Ensure that you are in the db2ct1sv instance.
Issue the set command and check the value of db2instance. If the value is not db2ct1sv, issue the following command:

```
set db2instance=db2ct1sv
```
 - c. Drop the SATCTLDB database by entering the following command:

```
db2 drop database satctl db
```
 - d. Create the new SATCTLDB database by entering the following command:

```
db2 -tf satctl db.dd1 -z satctl db.log
```
 - e. Issue the following command:

```
db2 terminate
```
3. Bind the db2satcs.dll stored procedure to the SATCTLDB database. Perform the following steps:
 - a. Connect to the SATCTLDB database by entering the following command

```
db2 connect to satctl db
```
 - b. Switch to the directory <db2path>\bnd, where <db2path> is the install drive and path, for example c:\sql11ib.
 - c. Issue the bind command, as follows:

```
db2 bind db2satcs.bnd
```
4. Enter the following command to close the connection to the database:

```
db2 connect reset
```
5. Close the DB2 Command Window.

Upgrading DB2 Enterprise Edition on AIX: Use the information in the sections that follow to determine the current service level of your Version 6 DB2 Enterprise Edition system, and the steps that you need to perform to update the system to the FixPak 2 service level, or higher. You will need to perform the steps of one or more of the following sections:

- „Assessing DB2 Enterprise Edition on AIX“ auf Seite 137
- „Determining What Needs to Be Done“ auf Seite 137
- „Installing the Control Server Component on AIX“ auf Seite 138
- „Installing FixPak 2 or Higher on AIX“ auf Seite 138
- „Upgrading the SATCTLDB Database on AIX“ auf Seite 138

Assessing DB2 Enterprise Edition on AIX: If you have Version 6 DB2 Enterprise Edition installed on AIX, perform the following steps:

1. Check whether the Control Server component is installed. Enter the following command:

```
lslpp -l | grep db2_06_01.ctsr
```

If no data is returned, the Control Server component is not installed.

2. Determine the service level of the DB2 Enterprise Edition. Log on as a DB2 instance owner, and issue the `db2level` command. Use the table that follows to interpret the output:

Values of Key Fields in the db2level output			Your DB2 system is at:
Release	Level	Informational Tokens	
SQL06010	01010104	db2_v6, n990616	Version 6.1 base
SQL06010	01020104	DB2 V6.1.0.1, n990824, U465423	Version 6.1 plus FixPak 1
SQL06010	01030104	DB2 V6.1.0.6, s991030, U468276 or DB2 V6.1.0.9, s000101, U469453	Version 6.1 plus FixPak 2

Anmerkung: If the level is greater than 01030104, your system is at a higher FixPak than FixPak 2.

3. Record the information that you find, and continue at „Determining What Needs to Be Done“.

Determining What Needs to Be Done: Using the information that you have gathered, find the row in the following table that applies to your situation, and follow the steps that are required to prepare your Version 6 DB2 Enterprise Edition system to support the DB2 control server at the FixPak 2 level.

Sections that follow the table provide instructions for performing the required steps. Consider checking off each step as you perform it. Only perform the steps that apply to your situation.

Control Server Component Installed	Service Level of DB2 Enterprise Edition System	Steps required to prepare your DB2 Enterprise Edition system
No	Version 6.1 base, or Version 6.1 plus FixPak 1, or Version 6.1 plus FixPak 2 or higher	Perform the following steps: 1. „Installing the Control Server Component on AIX“ auf Seite 138 2. „Installing FixPak 2 or Higher on AIX“ auf Seite 138 3. „Upgrading the SATCTLDB Database on AIX“ auf Seite 138
Yes	Version 6.1 base, or Version 6.1 plus FixPak 1	Perform the following steps: 1. „Installing FixPak 2 or Higher on AIX“ auf Seite 138 2. „Upgrading the SATCTLDB Database on AIX“ auf Seite 138
Yes	Version 6.1, plus FixPak 2 or higher	Perform the following step: 1. „Upgrading the SATCTLDB Database on AIX“ auf Seite 138

Installing the Control Server Component on AIX: To install the Control Server component on AIX

1. Log on as a user with root authority.
2. Insert the DB2 Universal Database Enterprise Edition Version 6.1 CD in the CD drive.
3. Change to the directory where the CD is mounted, for example, `cd /cdrom`.
4. Type the following command to start the DB2 installer:
`./db2setup`
5. When the DB2 Installer window opens, use the tab key to select the **Install** option, and press Enter.
6. Locate the Enterprise Edition line and use the tab key to select the **Customize** option beside it. Press Enter.
7. Select the DB2 Control Server component, tab to OK, and press Enter.
8. Follow the instructions on the remaining windows to complete the installation of the DB2 Control Server component.

When the installation process is complete, create the DB2CTLSV instance and the SATCTLDB database. To perform these tasks, follow the detailed instructions in "Setting up the DB2 Control Server on AIX" in Chapter 13 of the *Administering Satellites Guide and Reference*.

Installing FixPak 2 or Higher on AIX: To upgrade an existing DB2 Enterprise Edition system AIX to FixPak 2 or higher, either:

- Download the latest FixPak for DB2 Enterprise Edition for AIX V6.1 from the Web, along with its accompanying Fixpak.Readme. The FixPak can be downloaded by following the instructions at URL:

`http://www-4.ibm.com/software/data/db2/db2tech/version61.html`

Install the FixPak following the instructions in the Fixpak.Readme file.

- Use a DB2 Universal Database, Version 6.1 FixPak for AIX CD that is at FixPak 2 level or higher, and follow the instructions in the readme directory on the CD to complete the installation.

Ensure that you have updated the DB2CTLSV instance by running the `db2iupdt` command as instructed in the FixPak.Readme file.

Upgrading the SATCTLDB Database on AIX: To upgrade the SATCTLDB database on AIX:

1. Determine the level of the SATCTLDB database:
 - a. Log in as `db2ctlsv`.
 - b. Ensure that the database server has been started. If the server is not started, issue the `db2start` command.
 - c. Connect to the SATCTLDB database by entering the following command:
`db2 connect to satctldb`
 - d. Determine if the trigger `I_BATCHSTEP_TRGSCR` exists in the database by issuing the following query:
`db2 "select name from sysibm.systriggers where name='I_BATCHSTEP_TRGSCR'"`

Record the number of rows that are returned.

- e. Enter the following command to close the connection to the database:
`db2 connect reset`

If step 1d auf Seite 138 returned one row, the database is at the correct level. In this situation, skip step 2, and continue at step 3. If zero (0) rows are returned, the database is not at the correct level, and must be upgraded, as described in step 2, before you can perform step 3.

2. To upgrade the SATCTLDB database to the FixPak 2 level, perform the following steps. Enter all commands in the DB2 Command Window:
 - a. Switch to the \$HOME/sql1lib/misc directory.
 - b. Drop the SATCTLDB database by entering the following command:

```
db2 drop database satctl db
```
 - c. Create the new SATCTLDB database by entering the following command:

```
db2 -tf satctl db.dd1 -z $HOME/satctl db.log
```
 - d. Issue the following command:

```
db2 terminate
```
3. Bind the db2satcs.dll stored procedure to the SATCTLDB database. Perform the following steps:
 - a. Connect to the SATCTLDB database by entering the following command

```
db2 connect to satctl db
```
 - b. Switch to the directory \$HOME/sql1lib/bnd.
 - c. Issue the bind command, as follows:

```
db2 bind db2satcs.bnd
```
4. Enter the following command to close the connection to the database:

```
db2 connect reset
```

Upgrading a Version 6 Steuerzentrale and Satellitenverwaltungszentrale

To use a Version 6 Steuerzentrale and Satellitenverwaltungszentrale with a Version 6 DB2 control server and satellite control database (SATCTLDB) that have been upgraded to FixPak 2 or higher, the tools must also be upgraded to FixPak 2 or higher.

If the Steuerzentrale and the Satellitenverwaltungszentrale are running on the same system as the DB2 control server, they were upgraded when the DB2 Enterprise Edition system was upgraded to FixPak 2. However, if you run these tools on another system, you must upgrade this system to the FixPak 2 level or higher.

To upgrade this system to FixPak 2 or higher:

- Download the latest FixPak for your product at the V6.1 level from the Web, along with its accompanying readme. FixPaks can be downloaded by following the instructions at URL:

<http://www-4.ibm.com/software/data/db2/db2tech/version61.html>

Install the FixPak following the instructions in the readme file.

- Use a DB2 Universal Database, Version 6.1 FixPak CD for the operating system that you are running that is at FixPak 2 level or higher, and follow the instructions in the readme to complete the installation.

Command Reference

db2batch - Benchmark Tool

The last sentence in the description of the PERF_DETAIL parameter should read:

A value greater than 1 is only valid on DB2 Version 2 and DB2 UDB servers, and is not currently supported on host machines.

db2cap (new command)

db2cap - CLI/ODBC Static Package Binding Tool

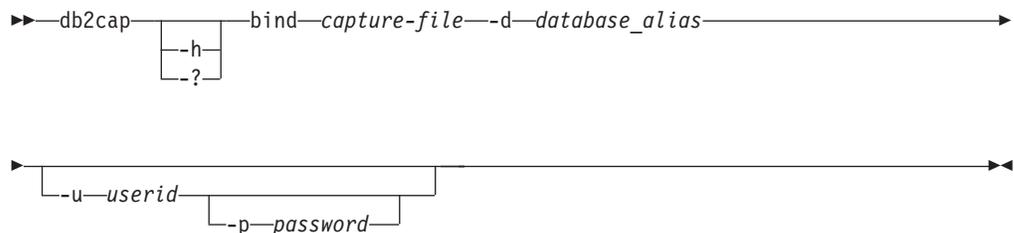
Binds a capture file to generate one or more static packages. A capture file is generated during a static profiling session of a CLI/ODBC/JDBC application, and contains SQL statements that were captured during the application run. This utility processes the capture file so that it can be used by the CLI/ODBC/JDBC driver to execute static SQL for the application.

For more information on how to use static SQL in CLI/ODBC/JDBC applications, see the Static Profiling feature in the *CLI Guide and Reference*.

Berechtigung

- Access privileges to any database objects referenced by SQL statements recorded in the capture file.
- Sufficient authority to set bind options such as OWNER and QUALIFIER if they are different from the connect ID used to invoke the **db2cap** command.
- BINDADD authority if the package is being bound for the first time; otherwise, BIND authority is required.

Befehlssyntax



Befehlsparameter

-h/-? Displays help text for the command syntax.

bind *capture-file*
Binds the statements from the capture file and creates one or more packages.

-d *database_alias*
Specifies the database alias for the database that will contain one or more packages.

-u *userid*
Specifies the user ID to be used to connect to the data source.

Anmerkung: If a user ID is not specified, a trusted authorization ID is obtained from the system.

-p *password*

Specifies the password to be used to connect to the data source.

Hinweise

The command must be entered in lowercase on UNIX platforms, but can be entered in either lowercase or uppercase on Windows operating systems and OS/2.

This utility supports a number of user-specified bind options that can be found in the capture file. For performance and security reasons, the file can be examined and edited with a text editor to change these options.

The SQLERROR(CONTINUE) and the VALIDATE(RUN) bind options can be used to create a package.

When using this utility to create a package, static profiling must be disabled.

The number of packages created depends on the isolation levels used for the SQL statements that are recorded in the capture file. The package name consists of up to a maximum of the first seven characters of the package keyword from the capture file, and one of the following single-character suffixes:

- 0 - Uncommitted Read (UR)
- 1 - Cursor Stability (CS)
- 2 - Read Stability (RS)
- 3 - Repeatable Read (RR)
- 4 - No Commit (NC)

To obtain specific information about packages, the user can:

- Query the appropriate SYSIBM catalog tables using the COLLECTION and PACKAGE keywords found in the capture file.
- View the capture file.

db2ckrst (new command)

db2ckrst - Check Incremental Restore Image Sequence

Queries the database history and generates a list of timestamps for the backup images required for an incremental restore. A simplified restore syntax for a manual incremental restore is also generated.

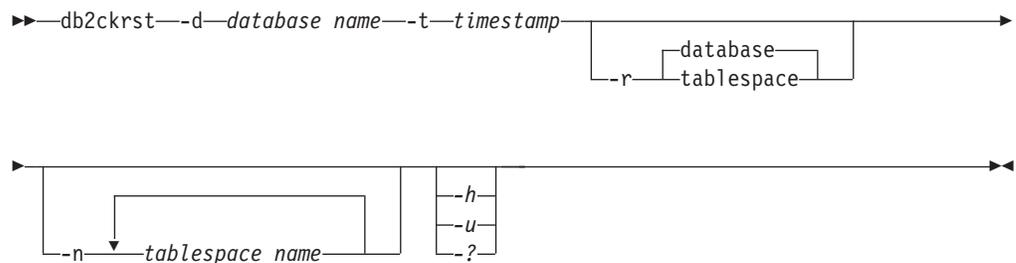
Berechtigung

None

Erforderliche Verbindung

None

Befehlssyntax



Befehlsparameter

-d database namefile-name

Specifies the alias name for the database that will be restored.

-t timestamp

Specifies the timestamp for a backup image that will be incrementally restored.

-r Specifies the type of restore that will be executed. The default is database.

Anmerkung: If tablespace is chosen and no table space names are given, the utility looks into the history entry of the specified image and uses the table space names listed to do the restore.

-n tablespace name

Specifies the name of one or more table spaces that will be restored.

Anmerkung: If a database restore type is selected and a list of table space names is specified, the utility will continue as a tablespace restore using the table space names given.

-h/-u/-?

Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Beispiele

```
db2ckrst -d mr -t 20001015193455 -r database
db2ckrst -d mr -t 20001015193455 -r tablespace
db2ckrst -d mr -t 20001015193455 -r tablespace -n tbsp1 tbsp2
```

```
> db2 backup db mr
```

```
Backup successful. The timestamp for this backup image is : 20001016001426
```

```
> db2 backup db mr incremental
```

```
Backup successful. The timestamp for this backup image is : 20001016001445
```

```
> db2ckrst -d mr -t 20001016001445
```

```
Suggested restore order of images using timestamp 20001016001445 for database mr.
```

```
=====
db2 restore db mr incremental taken at 20001016001445
db2 restore db mr incremental taken at 20001016001426
db2 restore db mr incremental taken at 20001016001445
=====
```

```
> db2ckrst -d mr -t 20001016001445 -r tablespace -n userspace1
```

```
Suggested restore order of images using timestamp 20001016001445 for database mr.
```

```
=====
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001445
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001426
db2 restore db mr tablespace ( USERSPACE1 ) incremental taken at 20001016001445
=====
```

Hinweise

The database history must exist in order for this utility to be used. If the database history does not exist, specify the HISTORY FILE option in the RESTORE command before using this utility.

If the FORCE option of the PRUNE HISTORY command is used, it will be possible to delete entries that are required for recovery from the most recent, full database backup image. The default operation of the PRUNE HISTORY command prevents required entries from being deleted. It is recommended that the FORCE option of the PRUNE HISTORY command not be used.

It is recommended that you keep good records of your backups and use this utility as a guide.

db2gncol (new command)

db2gncol - Update Generated Column Values

Updates generated columns in tables that are in check pending mode and have limited log space. This tool is used to prepare for a SET INTEGRITY statement on a table that has columns which are generated by expressions.

Berechtigung

One of the following

- *sysadm*
- *dbadm*

Befehlssyntax

```
►► db2gncol -d database -s schema_name -t table_name -c commit_count ►►  
  
┌──────────────────┬──────────┐  
└-u userid -p password┘ └-h┘
```

Befehlsparameter

-d *database*

Specifies an alias name for the database in which the table is located.

-s *schema_name*

Specifies the schema name for the table. The schema name is case sensitive.

-t *table_name*

Specifies the table for which new column values generated by expressions are to be computed. The table name is case sensitive.

-c *commit_count*

Specifies the number of rows updated between commits. This parameter influences the size of the log space required to generate the column values.

-u *userid*

Specifies a user ID with system administrator or database administrator privileges. If this option is omitted, the current user is assumed.

-p *password*

Specifies the password for the specified user ID.

-h

Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Hinweise

Using this tool instead of the FORCE GENERATED option on the SET INTEGRITY statement may be necessary if a table is large and the following conditions exist:

- All column values must be regenerated after altering the generation expression of a generated column.
- An external UDF used in a generated column was changed, causing many column values to change.
- A generated column was added to the table.
- A large load or load append was performed that did not provide values for the generated columns.

- The log space is too small due to long-running concurrent transactions or the size of the table.

This tool will regenerate all column values that were created based on expressions. While the table is being updated, intermittent commits are performed to avoid using up all of the log space. Once **db2gncol** has been run, the table can be taken out of check pending mode using the SET INTEGRITY statement.

db2inidb - Initialize a Mirrored Database

In a split mirror environment, this command is used to initialize a mirrored database for different purposes.

Authorization

Must be one of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*

Required Connection

None

Command Syntax



Command Parameters

database_alias

Specifies the alias of the database to be initialized.

SNAPSHOT

Specifies that the mirrored database will be initialized as a clone of the primary database. This database is read only.

STANDBY

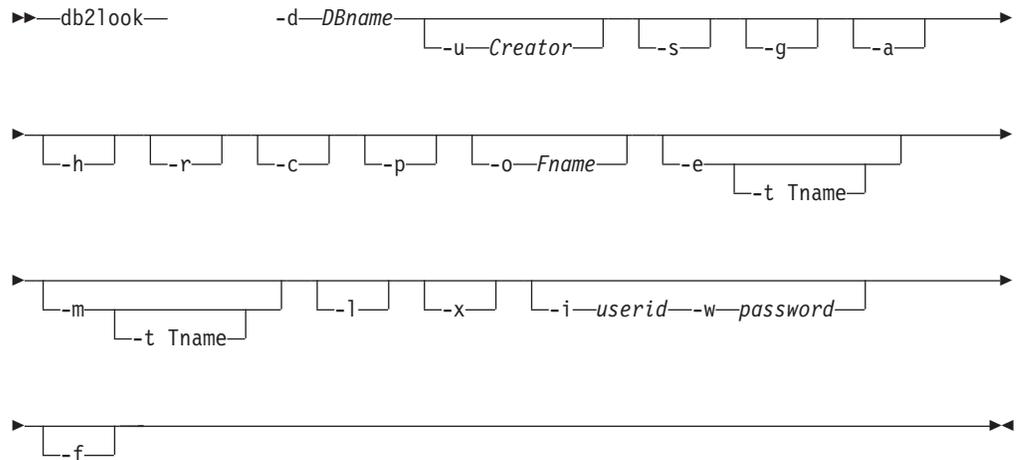
Specifies that the database will be placed in roll forward pending state. New logs from the primary database can be fetched and applied to the standby database. The standby database can then be used in place of the primary database if it goes down.

MIRROR

Specifies that the mirrored database is to be used as a backup image which can be used to restore the primary database.

db2look - DB2 Statistics Extraction Tool

The syntax diagram should appear as follows:



The `-td x` parameter has been added following the `-c` parameter. Its definition is as follows:

Specifies the statement delimiter for SQL statements generated by db2look. If this option is not specified, the default is the semicolon `';`. It is recommended that this option be used if the `-e` option is specified. In this case, the extracted objects may contain triggers or SQL routines.

The following example will also be added:

Generate the DDL statements for objects created by all users in the database DEPARTMENT. The db2look output is sent to file db2look.sql:

```
db2look -d department -a -e -td % -o db2look.sql  
db2 -td% -f db2look.sql
```

db2updv7 - Update Database to Version 7 Current Fix Level

This command updates the system catalogs in a database to support the current FixPak in the following ways:

- Enables the use of the new built-in functions (ABS, DECRYPT_BIN, DECRYPT_CHAR, ENCRYPT, GETHINT, MULTIPLY_ALT, and ROUND).
- Enables the use of the new built-in procedures (GET_ROUTINE_SAR and PUT_ROUTINE_SAR).
- Adds or applies corrections to WEEK_ISO and DAYOFWEEK_ISO functions on Windows and OS/2 databases.
- Applies a correction to table packed descriptors for tables migrated from Version 2 to Version 6.
- Creates the view SYSCAT.SEQUENCES.

Authorization

sysadm

Required Connection

Database. This command automatically establishes a connection to the specified database.

Command Syntax

```
db2upd7 -d database_name [-u userid -p password] [-h]
```

Command Parameters

-d database-name

Specifies the name of the database to be updated.

-u userid

Specifies the user ID.

-p password

Specifies the password for the user.

-h

Displays help information. When this option is specified, all other options are ignored, and only the help information is displayed.

Example

After installing the FixPak, update the system catalog in the sample database by issuing the following command:

```
db2upd7 -d sample
```

Usage Notes

This tool can only be used on a database running DB2 Version 7.1 or Version 7.2 with at least FixPak 2 installed. If the command is issued more than once, no errors are reported and each of the catalog updates is applied only once.

To enable the new built-in functions, all applications must disconnect from this database and the database must be deactivated if it has been activated.

New Command Line Processor Option (-x, Suppress printing of column headings)

A new option, -x, tells the command line processor to return data without any headers, including column names. The default setting for this command option is OFF.

True Type Font Requirement for DB2 CLP

To correctly display the national characters for single byte (SBCS) languages correctly from the DB2 command line processor (CLP) window, change the font to True Type.

ADD DATALINKS MANAGER

The required authorization level for this command is one of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*

The following usage note should be added:

This command is effective only after all applications have been disconnected from the database. The DB2 Data Links Manager being added must be completely set up and running for this command to be successful. The database must also be registered on the DB2 Data Links Manager using the `d1 fm add_db` command. The maximum number of DB2 Data Links Managers that can be added to a database is 16.

ARCHIVE LOG (new command)

Archive Log

Closes and truncates the active log file for a recoverable database. If user exit is enabled, issues an archive request.

Berechtigung

One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

Erforderliche Verbindung

This command automatically establishes a connection to the specified database. If a connection already exists, an error is returned.

Befehlssyntax

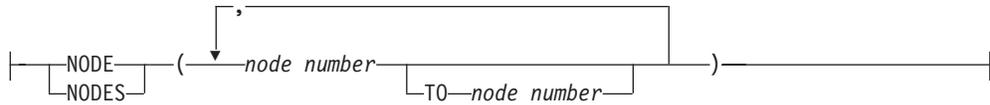
►► ARCHIVE LOG FOR DATABASE *database-alias* ►

► USER *username* USING *password* | On Node clause | ►►

On Node clause:

| ON Node List clause |
| ALL NODES EXCEPT Node List clause |

Node List clause:



Befehlsparameter

DATABASE database-alias

Specifies the alias of the database whose active log is to be archived.

USER username

Identifies the user name under which a connection will be attempted.

USING password

Specifies the password to authenticate the user name.

ON ALL NODES

Specifies that the command should be issued on all nodes in the `db2nodes.cfg` file. This is the default if a node clause is not specified.

EXCEPT

Specifies that the command should be issued on all nodes in the `db2nodes.cfg` file, except those specified in the node list.

ON NODE/ON NODES

Specifies that the logs should be archived for the specified database on a set of nodes.

node number

Specifies a node number in the node list.

TO node number

Used when specifying a range of nodes for which the logs should be archived. All nodes from the first node number specified up to and including the second node number specified are included in the node list.

Hinweise

This command can be used to collect a complete set of log files up to a known point. The log files can then be used to update a standby database.

This function can only be executed when there is no database connection to the specified database. This prevents a user from executing the command with uncommitted transactions. As such, the ARCHIVE LOG command will not forcibly commit the user's incomplete transactions. If a database connection to the specified database already exists and this command is executed, the command will terminate and return an error. If another application has transactions in progress with the specified database when this command is executed, there will be a slight performance degradation since the command flushes the log buffer to disk. Any other transactions attempting to write log records to the buffer will have to wait until the flush is complete.

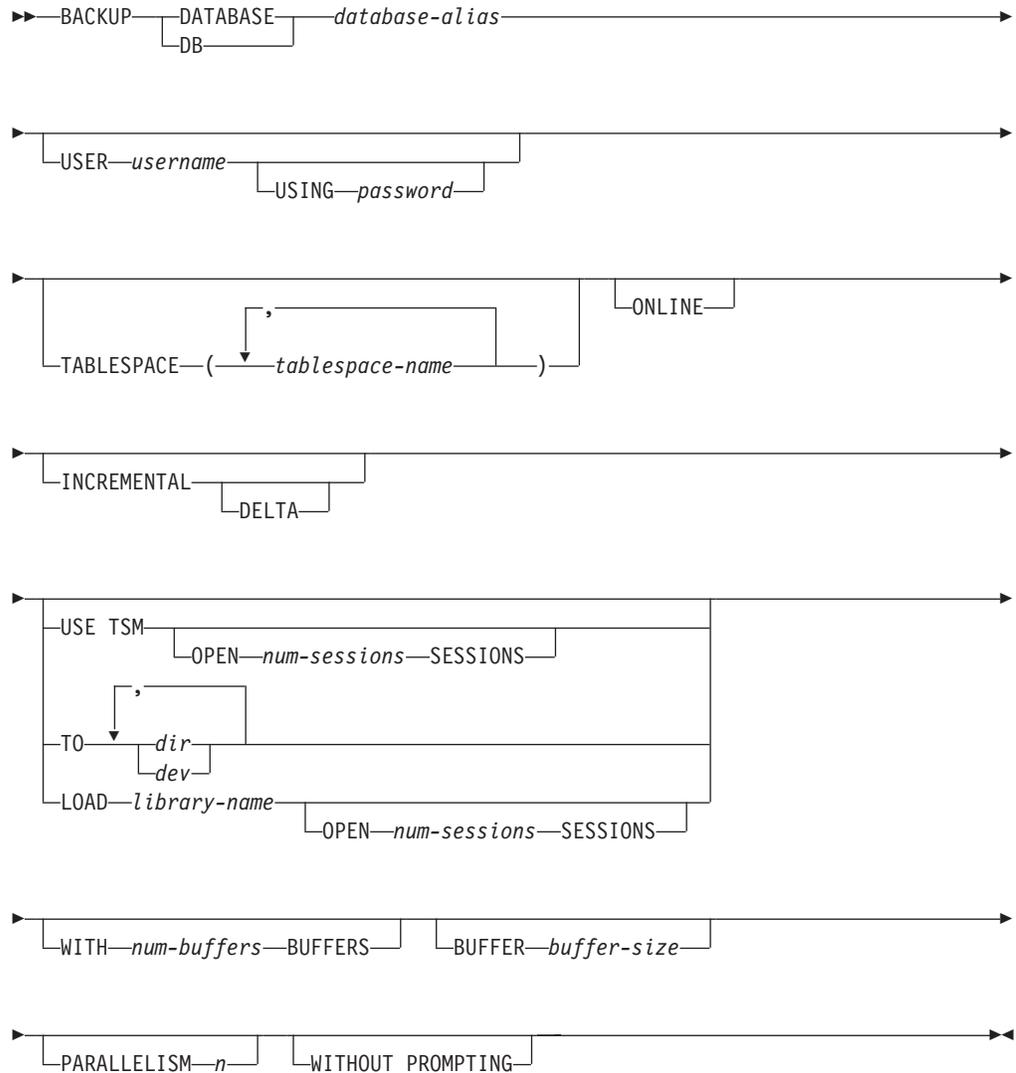
If used in an MPP environment, a subset of nodes may be specified by using a node clause. If the node clause is not specified, the default behaviour for this command is to close and archive the active log on all nodes.

Using this command will cause a database to lose a portion of its LSN space, and thereby hasten the exhaustion of valid LSNs.

BACKUP DATABASE

Syntax Diagram

The syntax diagram for BACKUP DATABASE will be updated to reflect the new INCREMENTAL parameter and the optional DELTA argument. Specifying the INCREMENTAL option alone will result in a cumulative backup image being produced. The optional DELTA argument can be used to specify the production of a non-cumulative backup image.



DB2 Data Links Manager Considerations

If one or more Data Links servers are configured for the database, the backup operation will succeed, even if a Data Links server is not available. When the Data Links server restarts, backup processing will be completed on that Data Links server before it becomes available to the database again.

Anmerkung: If there are twice as many backups still waiting for an unavailable Data Links server as are retained in the history file for the database (database configuration parameter `num_db_backups`), the backup operation will fail.

BIND

The command syntax for DB2 should be modified to show the federated parameter as follows:

```
FEDERATED---NO---  
          '-YES--'
```

FEDERATED

Specifies whether a static SQL statement in a package references a nickname or a federated view. If this option is not specified and a static SQL statement in the package references a nickname or a federated view, a warning is returned and the package is created.

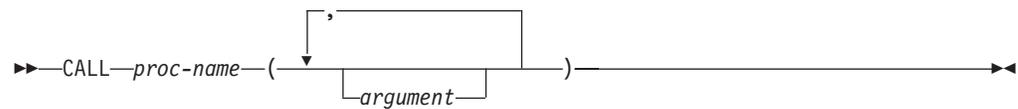
- NO** A nickname or federated view is not referenced in the static SQL statements of the package. If a nickname or federated view is encountered in a static SQL statement during the prepare or bind of this package, an error is returned and the package is *not* created.
- YES** A nickname or federated view can be referenced in the static SQL statements of the package. If no nicknames or federated views are encountered in static SQL statements during the prepare or bind of the package, no errors or warnings are returned and the package is created.

Anmerkung: In Version 7 FixPak 2, an SQL1179W warning message is generated by the server when precompiling a source file or binding a bind file without specifying a value for the FEDERATED option. The same message is generated when the source file or bind file includes static SQL references to a nickname. There are two exceptions:

- For clients that are at an earlier FixPak than Version 7 FixPak 2 or for downlevel clients, the `sqlprep()` API does not report this SQL1179W warning in the message file. The Command Line Processor PRECOMPILE command also does not output the warning in this case.
- For clients that are at an earlier FixPak than Version 7 FixPak 2 or for downlevel clients, the `sqlabndx` API does report this SQL1179W warning in the message file. However, the message file also incorrectly includes an SQL0092N message indicating that no package was created. This is not correct as the package is indeed created. The Command Line Processor BIND command returns the same erroneous warning.

CALL

The syntax for the CALL command should appear as follows:



The description of the *argument* parameter has been changed to:

Specifies one or more arguments for the stored procedure. All input and output arguments must be specified in the order defined by the procedure. Output arguments are specified using the "?" character. For example, a stored procedure foo with one integer input parameter and one output parameter would be invoked as "call foo (4, ?)".

Anmerkungen:

1. When invoking this utility from an operating system prompt, it may be necessary to delimit the command as follows:

```
"call DEPT_MEDIAN (51)"
```

A single quotation mark (') can also be used.

2. The stored procedure being called must be uniquely named in the database.
3. The stored procedure must be cataloged. If an uncataloged procedure is called, a DB21036 error message is returned.
4. A DB21101E message is returned if not enough parameters are specified on the command line, or the command line parameters are not in the correct order (input, output), according to the stored procedure definition.
5. There is a maximum of 1023 characters for a result column.
6. LOBS and binary data (FOR BIT DATA, VARBINARY, LONGVARBINARY, GRAPHIC, VARGAPHIC, or LONGVARGRAPHIC) are not supported.
7. CALL supports result sets.
8. If an SP with an OUTPUT variable of an unsupported type is called, the CALL fails, and message DB21036 is returned.
9. The maximum length for an INPUT parameter to CALL is 1024.

DROP DATALINKS MANAGER (new command)

DROP DATALINKS MANAGER

Drops a DB2 Data Links Manager from the list of registered DB2 Data Links Managers for a specified database.

Berechtigung

One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*

Befehlssyntax

```
►► DROP DATALINKS MANAGER FOR DATABASE  
DB dbname USING name ◀◀
```

Befehlsparameter

DATABASE *dbname*

Specifies a database name.

USING *name*

Specifies the name of the DB2 Data Links Manager server as shown by the LIST DATALINKS MANAGER command.

Beispiele

Example 1

Dropping a DB2 Data Links Manager *micky.almaden.ibm.com* from database TEST under instance *validate* residing on host *bramha.almaden.ibm.com* when some database tables have links to *micky.almaden.ibm.com*. *It is extremely important that the following steps be taken when dropping a DB2 Data Links Manager.*

1. Take a database backup for database TEST.
2. If there are any links to *micky.almaden.ibm.com*, unlink them:
 - a. Log on with a user ID belonging to SYSADM_GROUP and obtain an exclusive mode connection the the database TEST.

```
connect to test in exclusive mode
```

Ensure that this is the only connection to test using that user ID. This will prevent any new links from being created.

- b. Obtain a list of all FILE LINK CONTROL DATALINK columns and the tables containing them in the database.

```
select tabname, colname from syscat.columns where substr(dl_features,  
2, 1) = 'F'
```

- c. For each FILE LINK CONTROL DATALINK column in the list, issue SQL SELECT to determine if links to *micky.almaden.ibm.com* exist. For example, for a DATALINK column *c* in table *t*, the SELECT statement would be:

```
select count(*) from t where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

- d. For each FILE LINK CONTROL DATALINK column containing such links, issue SQL UPDATE to unlink values which are links to micky.almaden.ibm.com. For example, for a DATALINK column c in table t, the UPDATE statement would be:

```
update t set t.c = null where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

If t.c is not nullable, the following can be used instead:

```
update t set t.c = dlvalue('') where dlurlserver(t.c) = 'MICKY.ALMADEN.IBM.COM'
```

- e. Commit this SQL UPDATE:

```
commit
```

3. Issue the DROP DATALINKS MANAGER command:

```
drop datalinks manager for db test using node micky.almaden.ibm.com
```

4. Terminate the exclusive mode connection to make the changes effective and to allow other connections to the database:

```
terminate
```

5. Initiate unlink processing and garbage collection of backup information for TEST on micky.almaden.ibm.com. As DB2 Data Links Manager Administrator, issue the following command on micky.almaden.ibm.com:

```
dlfm drop_dlm test validate bramha.almaden.ibm.com
```

This will unlink any files that are still linked to database TEST, just in case the user has missed unlinking them before invoking step 3. If micky.almaden.ibm.com has backup information (for example, archive files, metadata) for files previously linked to database TEST, this command will initiate garbage collection of that information. The actual unlinking and garbage collection will be performed asynchronously.

Example 2

Deleting DATALINK values that are links to files on a DB2 Data Links Manager called micky.almaden.ibm.com, when the Manager has already been dropped from database TEST. This may be required if steps in Example 1 were not followed while dropping micky.almaden.ibm.com. SQL DELETE, SELECT, and UPDATE statements will not be successful for such DATALINK values (SQL0368). The user must run a reconcile operation for each table that contains such DATALINK values. Each DATALINK value that was a link to micky.almaden.ibm.com will be updated to NULL or a zero-length DATALINK value. Any row containing such a value will be inserted into the exception table (if one was specified). However, the DATALINK value will not include the prefix name. The prefix name in the original DATALINK value is no longer obtainable by the system, because micky.almaden.ibm.com has been dropped. For example, if the original DATALINK value was 'http://host.com/dlfs/x/y/a.b' and '/dlfs' is the prefix name, the DATALINK value in the exception table will contain 'http://host.com/x/y/a.b'. The files referenced by these DATALINK values will continue to remain in linked state on the DB2 Data Links Manager. The dlfm drop_dlm command can be issued on micky.almaden.ibm.com to initiate unlink processing for these files. If micky.almaden.ibm.com has backup information (for example, archive files, metadata) for files previously linked to database TEST, this command will initiate garbage collection of that information. The actual unlinking and garbage collection will be performed asynchronously.

Example 3

Multiple incarnations of a DB2 Data Links Manager micky.almaden.ibm.com for a database TEST. This scenario demonstrates that a DB2 Data Links Manager can be re-registered after being dropped, and that it is then treated as a completely new DB2 Data Links Manager. The following steps are only illustrative of a scenario that is possible. If, as recommended, the steps in Example 1 are followed for dropping micky.almaden.ibm.com, links to the older incarnation of micky.almaden.ibm.com will not exist; that is, one will not see error SQL0368 in step 7 below.

1. Register micky.almaden.ibm.com to database TEST:
add datalinks manager for db test using node micky.almaden.ibm.com port 14578

2. Create links to files on micky.almaden.ibm.com:

```
connect to test
create table t(c1 int, c2 datalink linktype url file link control mode
db2options)
insert into t values(1, dlvalue('file://micky.almaden.ibm.com/
pictures/yosemite.jpg'))
commit
terminate
```

3. Drop micky.almaden.ibm.com from database TEST:

```
drop datalinks manager for db test using micky.almaden.ibm.com
```

4. Select DATALINK values:

```
connect to test
select * from t
terminate
```

The user will see:

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMADEN.IBM.COM"
is not registered to the database. SQLSTATE=55022.
```

5. Register micky.almaden.ibm.com to database TEST again:

```
add datalinks manager for db test using node micky.almaden.ibm.com port 14578
```

6. Insert more DATALINK values:

```
connect to test
insert into t values(2, dlvalue('file://micky.almaden.ibm.com/pictures/tahoe.jpg'))
commit
```

7. Select DATALINK values:

```
select c2 from t where c1 = 2
```

is successful because the value being selected is a link to the currently registered incarnation of micky.almaden.ibm.com.

```
select c2 from t where c1 = 1
```

returns:

```
SQL0368 The DB2 Data Links Manager "MICKY.ALMADEN.IBM.COM"
is not registered to the database. SQLSTATE=55022.
```

because the value being selected is a link to the incarnation of micky.almaden.ibm.com which was dropped in step 3 above.

Hinweise

The effects of the DROP DATALINKS MANAGER command cannot be rolled back. It is extremely important to follow the steps outlined in Example 1 when using the DROP DATALINKS MANAGER command.

This command is effective only after all applications have been disconnected from the database.

Upon successful completion of the command, the user is informed (DB210201I) that no processing has been done on the DB2 Data Links Manager. Before dropping a DB2 Data Links Manager, the user must ensure that the database does not have any links to files on that DB2 Data Links Manager. If links do exist in the database after a DB2 Data Links Manager has been dropped, the user must run the reconcile utility to get rid of them. The reconcile utility will set these links to NULL (if the DATALINK column is nullable), or to a zero-length DATALINK value.

Files corresponding to links between a database and a dropped DB2 Data Links Manager remain in linked state. That is, they are inaccessible to operations like read, write, rename, delete, change of permissions, or change of ownership.

Archived copies of unlinked files on the DB2 Data Links Manager will not be garbage collected by this command. However, users can explicitly initiate unlink processing and garbage collection using the dlfm drop_dlm command on the DB2 Data Links Manager.

It is recommended that a database backup be taken before dropping a DB2 Data Links Manager. In addition, ensure that all replication subscriptions have replicated all changes involving this DB2 Data Links Manager.

If a backup was taken before the DB2 Data Links Manager was dropped from a database, and that backup image is used to restore after that DB2 Data Links Manager was dropped, restore or rollforward processing may put certain tables in datalink reconcile pending (DRP) state.

EXPORT

In the section "DB2 Data Links Manager Considerations", Step 3 of the procedure to ensure that a consistent copy of the table and the corresponding files referenced by DATALINK columns are copied for export should read:

3. Run the dlfm_export utility at each Data Links server. Input to the dlfm_export utility is the control file name, which is generated by the export utility. This produces a tar (or equivalent) archive of the files listed within the control file. For Distributed File Systems (DFS), the dlfm_export utility will get the DCE network root credentials before archiving the files listed in the control file. dlfm_export does not capture the ACLs information of the files that are archived.

In the same section, the bullets following "Successful execution of EXPORT results in the generation of the following files" should be modified as follows:

The second sentence in the first bullet should read:

- A DATALINK column value in this file has the same format as that used by the import and load utilities.

The first sentence in the second bullet should read:

Control files server_name, which are generated for each Data Links server. (On the Windows NT operating system, a single control file, ctrlfile.lst, is used by all Data Links servers. For DFS, there is one control file for each cell.)

The following sentence should be added to the paragraph before Table 5:

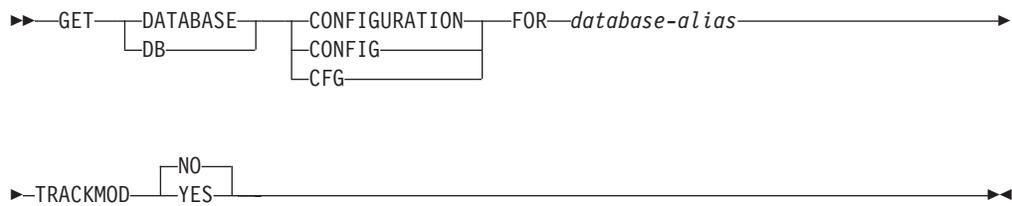
For more information about dlfm_export, refer to the "Data Movement Utilities Guide and Reference" under "Using Export to move DB2 Data Links Manager Data".

GET DATABASE CONFIGURATION

The description of the DL_TIME_DROP configuration parameter should be changed to the following:

Applies to DB2 Data Links Manager only. This parameter specifies the number of days files would be retained on an archive server (such as a TSM server) after a DROP DATABASE command is issued.

The new parameter TRACKMOD will be added to the GET DATABASE CONFIGURATION command. The syntax will appear as follows:



The parameter description will be added as follows:

TRACKMOD

Indicates whether DB2 should track modified pages in the database in order to allow incremental backups to be taken.

- OFF** Specifies that changed pages should not be tracked. This is the default for databases created prior to Version 7.1, FixPak 3.
- ON** Specifies that changed pages should be tracked. When this parameter is set, incremental backups of the database can be made. This is the default for databases created with Version 7.1, FixPak 3 or later.

GET ROUTINE (new command)

GET ROUTINE

Retrieves a routine SQL Archive (SAR) file for a specified SQL routine.

Berechtigung

dbadm

Erforderliche Verbindung

Database. If implicit connect is enabled, a connection to the default database is established.

Befehlssyntax

```
▶▶ GET ROUTINE INTO file_name FROM _____ PROCEDURE routine_name ▶▶  
                                          └ SPECIFIC ─┘
```

Befehlsparameter

INTO *file-name*

Names the file where routine SQL archive (SAR) is stored.

FROM

Indicates that start of the specification of the routine to be retrieved.

SPECIFIC

The specified *routine-name* is given as a specific name.

PROCEDURE

The routine is an SQL procedure.

routine-name

The name of the procedure. If SPECIFIC is specified then it is the specific name of the procedure. If the name is not qualified with a schema name, the CURRENT SCHEMA is used as the schema name of the routine. The *routine-name* must be an existing procedure that is defined as an SQL procedure.

Beispiele

```
GET ROUTINE INTO procs/procl.sar FROM PROCEDURE myappl.procl;
```

GET SNAPSHOT

The description for the FCM FOR ALL NODES parameter should appear as follows:

Provides Fast Communication Manager (FCM) statistics between the node against which the GET SNAPSHOT command was issued and the other nodes in the EEE instance.

IMPORT

In the section "DB2 Data Links Manager Considerations", the following sentence should be added to Step 3:

For Distributed File Systems (DFS), update the cell name information in the URLs (of the DATALINK columns) from the exported data for the SQL table, if required.

The following sentence should be added to Step 4:

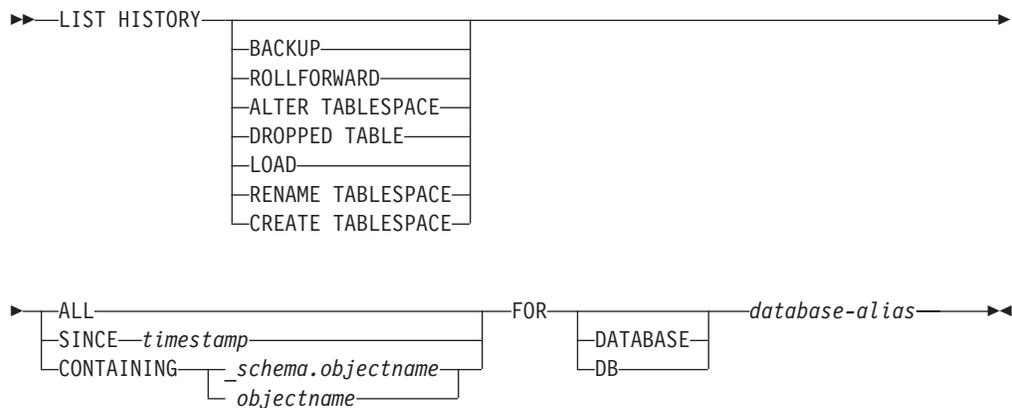
For DFS, define the cells at the target configuration in the DB2 Data Links Manager configuration file.

The paragraph following Step 4 should read:

When the import utility runs against the target database, files referred to by DATALINK column data are linked on the appropriate Data Links servers.

LIST HISTORY

The CREATE TABLESPACE parameter will be added to the LIST HISTORY command. The syntax diagram will appear as follows:



The parameter description will be added as follows:

CREATE TABLESPACE

Lists all CREATE TABLESPACE and DROP TABLESPACE operations.

The Usage Notes will be updated as follows:

The following symbols will be added to the Operation section of the report:

- A - Create tablespace
- O - Drop tablespace
- U - Unload

The symbols under the Type section of the report will be reorganized as follows:

- Backup Types
 - F - Offline
 - N - Online
 - I - Incremental Offline
 - O - Incremental Online
 - D - Delta Offline
 - E - Delta Online
- Rollforward Types
 - E - End of log
 - P - Point in time
- Load Types
 - I - Insert
 - R - Replace
- Alter tablespace Types
 - C - Add containers
 - R - Rebalance
- Quiesce Types
 - S - Quiesce Share
 - U - Quiesce Update
 - X - Quiesce Exclusive
 - Z - Quiesce Reset

LOAD

In the section "DB2 Data Links Manager Considerations", add the following sentence to Step 1 of the procedure that is to be performed before invoking the load utility, if data is being loaded into a table with a DATALINK column that is defined with FILE LINK CONTROL:

For Distributed File Systems (DFS), ensure that the DB2 Data Links Managers within the target cell are registered.

The following sentence should be added to Step 5:

For DFS, register the cells at the target configuration referred to by DATALINK data (to be loaded) in the DB2 Data Links Manager configuration file.

In the section "Representation of DATALINK Information in an Input File", the first note following the parameter description for `urlname` should read:

Currently "http", "file", "unc", and "dfs" are permitted as a schema name.

The first sentence of the second note should read:

The prefix (schema, host, and port) of the URL name is optional. For DFS, the prefix refers to the schema cellname filesystem-junction portion.

In the DATALINK data examples for both the delimited ASCII (DEL) file format and the non-delimited ASCII (ASC) file format, the third example should be removed.

The DATALINK data examples in which the load or import specification for the column is assumed to be `DL_URL_DEFAULT_PREFIX` should be removed and replaced with the following:

Following are DATALINK data examples in which the load or import specification for the column is assumed to be `DL_URL_REPLACE_PREFIX` ("http://qso"):

- `http://www.almaden.ibm.com/mrep/intro.mpeg`
This sample URL is stored with the following parts:
 - schema = http
 - server = qso
 - path = /mrep/intro.mpeg
 - comment = NULL string
- `/u/me/myfile.ps`
This is stored with the following parts:
 - schema = http
 - server = qso
 - path = /u/me/myfile.ps
 - comment = NULL string

PING (new command)

PING

Tests the network response time of the underlying connectivity between a client and a database server where DB2 Connect is used to establish the connection.

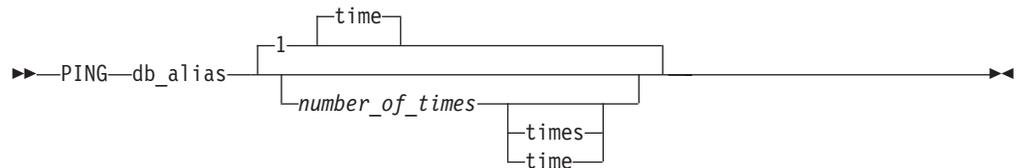
Berechtigung

None

Erforderliche Verbindung

Database

Befehlssyntax



Befehlsparameter

db_alias

Specifies the database alias for the database on a DRDA server that the ping is being sent to.

Anmerkung: This parameter, although mandatory, is not currently used. It is reserved for future use. Any valid database alias name can be specified.

number of times

Specifies the number of iterations for this test. The value must be between 1 and 32767 inclusive. The default is 1. One timing will be returned for each iteration.

Beispiele

To test the network response time for the connection to the host database server hostdb once:

```
db2 ping hostdb 1
```

or:

```
db2 ping hostdb
```

The command will display output that looks like this:

```
Elapsed time: 7221 microseconds
```

To test the network response time for the connection to the host database server hostdb 5 times:

```
db2 ping hostdb 5
```

or:

```
db2 ping hostdb 5 times
```

The command will display output that looks like this:

```
Elapsed time: 8412 microseconds  
Elapsed time: 11876 microseconds  
Elapsed time: 7789 microseconds  
Elapsed time: 10124 microseconds  
Elapsed time: 10988 microseconds
```

Hinweise

A database connection must exist before invoking this command, otherwise an error will result.

The elapsed time returned is for the connection between the client and a DRDA server database via DB2 Connect.

PUT ROUTINE (new command)

PUT ROUTINE

Uses the specified routine SQL Archive (SAR) file to define a routine in the database.

Berechtigung

dbadm

Erforderliche Verbindung

Database. If implicit connect is enabled, a connection to the default database is established.

Befehlssyntax

```
▶▶ PUT ROUTINE FROM file-name [OWNER new-owner] [USE REGISTERS] ▶▶
```

Befehlsparameter

FROM *file-name*

Names the file where routine SQL archive (SAR) is stored.

OWNER *new-owner*

Specifies a new authorization-name for the routine that will be used for authorization checking of the routine. The *new-owner* must have the necessary privileges for the routine to be defined. If the OWNER clause is not specified, the authorization-name that originally defined the routine is used.

USE REGISTERS

Indicates that the CURRENT SCHEMA and CURRENT PATH special registers are used to define the routine. If this clause is not specified, the settings for the default schema and SQL path are the settings used when the routine was originally defined. CURRENT SCHEMA is used as the schema name for unqualified object names in the routine definition (including the name of the routine) and CURRENT PATH is used to resolve unqualified routines and data types in the routine definition.

Beispiele

```
PUT ROUTINE FROM procs/proc1.sar;
```

Hinweise

No more than one procedure can be concurrently installed under a given schema.

RECONCILE

The following usage note should be added to the command description:

During reconciliation, attempts are made to link files which exist according to table data, but which do not exist according to Data Links Manager metadata, if no other conflict exists. A required DB2 Data Links Manager is one which has a referenced DATALINK value in the table. Reconcile tolerates unavailability of a required DB2 Data Links Manager as well as those that are configured to the database but are not part of the table data.

If an exception table is not specified, the exception report file (*filename.exp*) will have the host name, file name, column ID, and reason code for each of the DATALINK column values for which file references could not be re-established. If the file reference could not be re-established because the DB2 Data Links Manager itself was dropped from the database using the DROP DATALINKS MANAGER command, the file name reported in the exception report file is not the full file name; that is, the prefix part is missing. In the exception table for the DATALINK values whose DB2 Data Links Manager is dropped or is not available, the file name in the DATALINK value is not the full file name. The prefix part is missing. For example, if the original DATALINK value was 'http://host.com/dlfs/x/y/a.b', the value reported in the exception table will be 'http://host.com/x/y/a.b'; that is, the prefix name 'dlfs' will not be included. The exception report file in this case will have 'x/y/a.b'; that is, the prefix name 'dlfs' will not be included.

At the end of the reconciliation process, the table is taken out of datalink reconcile pending (DRP) state only if reconcile processing is complete on all the required DB2 Data Links Managers. If reconcile processing is pending on any of the required DB2 Data Links Managers (because they were unavailable), the table will remain, or be placed, in DRP state.

The following should be added to the list of possible violations:

```
00010-DB2 Data Links Manager referenced by the DATALINK value
has been dropped from the database using the DROP DATALINKS
MANAGER command. In this case, the corresponding DATALINK value
in the exception table will not contain the prefix name.
For example, if the original DATALINK value was
'http://host.com/dlfs/prfx/x/y/a.b', and '/dlfs/prfx' is
the prefix name, the exception table will contain
'http://host.com/x/y/a.b'.
```

REORGANIZE TABLE

The following sentence will be added to the Usage Notes:

REORGANIZE TABLE cannot use an index that is based on an index extension.

RESTORE DATABASE

Syntax

The following option will be added to the syntax of the RESTORE DATABASE command after the TABLESPACE/TABLESPACE ONLINE/HISTORY FILE options:



The parameter descriptions will be added as follows:

INCREMENTAL

Results in a manual cumulative restore of the database. The user will issue each of the restore commands.

ABORT

This parameter should be used to terminate an incremental restore before successful completion.

The following examples will also be added:

The following is a sample weekly incremental backup strategy with a recoverable database. A full backup is scheduled once per week, a delta every day, plus an incremental mid-week:

```
(Sun) backup db kdr use adsm  
(Mon) backup db kdr online incremental delta use adsm  
(Tue) backup db kdr online incremental delta use adsm  
(Wed) backup db kdr online incremental use adsm  
(Thu) backup db kdr online incremental delta use adsm  
(Fri) backup db kdr online incremental delta use adsm  
(Sat) backup db kdr online incremental use adsm
```

For a manual database restore of images created above on Friday morning, issue the following commands:

```
restore db kdr incremental taken at (Thu)  
restore db kdr incremental taken at (Sun)  
restore db kdr incremental taken at (Wed)  
restore db kdr incremental taken at (Thu)
```

Anmerkung: Any RESTORE command of the form db2 restore db <name> will perform a full database restore, regardless of whether the image being restored is a database image or a table space image. Any RESTORE command of the form db2 restore db <name> tablespace will perform a table space restore of the table spaces found in the image. Any RESTORE command in which a list of table spaces is provided will perform a restore of whatever table spaces were explicitly listed.

DB2 Data Links Manager Considerations

The second paragraph in the section entitled "DB2 Data Links Manager Considerations" should be replaced with:

If one or more Data Links servers are configured for the database, the restore operation will succeed, even if a Data Links server is not available. When the Data Links server restarts, restore processing will be completed on that Data Links server before it becomes available to the database again. NOTE: If a database restore operation is still waiting for an unavailable Data Links server, any subsequent database or table space restore operations will fail.

ROLLFORWARD DATABASE

The second paragraph in the section entitled "DB2 Data Links Manager Considerations" should be replaced with:

If one or more Data Links servers are configured for the database, the rollforward operation will succeed, even if a Data Links server is not available. When the Data Links server restarts, rollforward processing will be completed on that Data Links server before it becomes available to the database again.

Documentation Error in CLP Return Codes

In the Command Line Processor Return Codes section of Chapter 2, the second paragraph should appear as follows:

For example, the following Bourne shell script executes the GET DATABASE MANAGER CONFIGURATION command, then inspects the CLP return code:

```
db2 get database manager configuration
if [ "$?" = "0" ]
then echo "OK!"
fi
```

Versetzen von Daten Dienstprogramme und Referenz

Kapitel 2. Import

Verwenden von IMPORT mit gepufferten INSERT-Operationen

Die Anmerkung am Ende dieses Abschnitts muss wie folgt lauten:

Anmerkung: In allen Umgebungen mit Ausnahme von EEE werden gepufferte Einfügungen während Importen inaktiviert, bei denen der Parameter INSERT_UPDATE angegeben ist.

Kapitel 3. Load

Statusangaben für anstehende Aktionen nach einer Ladeoperation

Die beiden ersten Sätze im letzten Absatz dieses Abschnitts wurden wie folgt geändert:

Der vierte mögliche Status, der dem LOAD-Prozess zugeordnet ist (Status "Überprüfung anstehend"), bezieht sich auf referenzielle Integritätsbedingungen und Prüfungen auf Integritätsbedingungen, auf DATALINKS-Integritätsbedingungen, auf AST-Integritätsbedingungen oder auf Integritätsbedingungen für generierte Spalten. Wenn eine vorhandene Tabelle beispielsweise eine übergeordnete Tabelle mit einem Primärschlüssel ist, auf den ein Fremdschlüssel in einer abhängigen Tabelle verweist, werden beim Ersetzen von Daten in der übergeordneten Tabelle beide Tabellen (nicht der Tabellenbereich) in den Status "Überprüfung anstehend" versetzt.

LOAD-Rahmenbedingungen und -Einschränkungen

Für generierte Spalten und das Dienstprogramm LOAD gelten die folgenden Rahmenbedingungen:

- Eine Tabelle mit einer generierten Spalte in einem eindeutigen Index kann nur dann geladen werden, wenn die generierte Spalte eine INCLUDE-Spalte des Index ist oder der Dateitypwert generatedoverride verwendet wird. Bei Verwendung dieses Werts wird erwartet, dass alle Werte für die Spalte in der Eingabedatendatei bereitgestellt werden.
- Eine Tabelle mit einer generierten Spalte in einem Partitionierungsschlüssel kann nur dann geladen werden, wenn der Dateitypwert generatedoverride verwendet wird. Bei Verwendung dieses Werts wird erwartet, dass alle Werte für die Spalte in der Eingabedatendatei bereitgestellt werden.

Dateitypänderungswert "totalfreespace"

Der Dateitypänderungswert „totalfreespace“ (LOAD) wurde modifiziert, so dass er jetzt einen Wert zwischen 0 und 2 147 483 647 akzeptiert.

Kapitel 4. AutoLoader

"rexecd" erforderlich für Ausführung von Autoloader bei aktivierter Authentifizierung

Im Abschnitt „AutoLoader-Optionen“ wird den Beschreibungen der Parameter AUTHENTICATION und PASSWORD folgende Anmerkung hinzugefügt:

Wenn Sie in einer Linux-Umgebung den Autoloader mit aktivierter Authentifizierung (AUTHENTICATION=YES) ausführen, muss **rexecd** auf allen Maschinen aktiviert sein. Ist **rexecd** nicht aktiviert, wird folgende Fehlermeldung generiert:

```
openbreeze.torolab.ibm.com: Die Verbindung wurde zurückgewiesen.  
SQL6554N Beim Versuch, einen Prozess fern auszuführen, ist ein Fehler  
aufgetreten.
```

Die folgenden Fehlermeldungen werden in der Datei db2diag.log generiert:

```
2000-10-11-13.04.16.832852 Instance:svttdbm Node:000  
PID:19612(db2at1d) Appid:  
oper_system_services sqloRemoteExec Probe:31
```

Replikation Referenzhandbuch

Replikations-Server und Nicht-IBM Server

Sie müssen DataJoiner Version 2 oder höher verwenden, um Daten auf bzw. von Nicht-IBM Server(n) wie Informix, Microsoft SQL Server, Oracle, Sybase oder Sybase SQL Anywhere zu replizieren. Für diese Replikationsart können Sie die relationale Verbindungsfunktion nicht verwenden, da DB2 Relational Connect Version 7 keine Aktualisierungsfunktion besitzt. Außerdem benötigen Sie DJRA (DataJoiner Replication Administration) für die Verwaltung dieser heterogenen Replikation auf allen Plattformen (AS/400, OS/2, OS/390, UNIX und Windows) für alle Versionen von DB2 und DataJoiner.

Replikation unter Windows 2000

DB2 DataPropagator Version 7 ist mit dem Betriebssystem Windows 2000 kompatibel.

Bekannter Fehler beim Speichern von SQL-Dateien

Wenn Sie die Steuerzentrale in DB2 Connect Personal Edition verwenden, können Sie keine SQL-Dateien speichern. Wenn Sie versuchen, eine SQL-Datei zu speichern, bekommen Sie eine Fehlermeldung, dass der Datenbankverwaltungsserver (Database Administration Server, DAS) nicht aktiv ist, obwohl der Server eigentlich nicht verfügbar ist, weil er nicht mit DB2 Connect PE ausgeliefert wird.

DB2-Verwaltung

Es wird empfohlen, dass Sie die neuesten DB2-Pflegeprogramme für die verschiedenen DB2-Produkte installieren, die Sie in Ihrer Replikationsumgebung verwenden.

Dienstprogramm "Data Difference" im Internet

Sie können das Dienstprogramm „Data Difference“ aus dem Internet unter <ftp://ftp.software.ibm.com/ps/products/datapropagator/fixes/> herunterladen. Dieses Dienstprogramm ist ein Beispieldienstprogramm, das Sie verwenden können, um zwei Versionen derselben Datei zu vergleichen und eine Ausgabedatei zu erstellen, die die Unterschiede anzeigt. Einzelangaben finden Sie in der Readme-Datei, die dem Beispieldienstprogramm beiliegt.

Kapitel 3. Beispielszenario für die Datenreplikation

Replikationsszenarios

Ein neues Szenario für die Replikation heterogener Daten finden Sie auf der Bibliotheksseite der DataPropagator-Website (<http://www.ibm.com/software/data/dpropr/>). Führen Sie die Schritte in diesem Szenario aus, um Änderungen von einer Replikationsquellentabelle in einer Oracle-Datenbank unter AIX in eine Zieltabelle einer Datenbank in DB2 für Windows NT zu kopieren. Dieses Szenario verwendet DB2 DataJoiner Replication Administration (DJRA), Auslöser für Erfassungsfunktion, das Apply-Programm und DB2 DataJoiner.

In Kapitel 3, "Beispielszenario für die Datenreplikation", im Abschnitt "Einrichten der Replikationsumgebung für dieses Szenario" sollte Schritt 6, "Erstellen einer Kennwortdatei", wie folgt lauten:

Schritt 6: Erstellen einer Kennwortdatei

Da das Apply-Programm eine Verbindung zum Quellen-Server herstellen muss, müssen Sie eine Kennwortdatei für die Benutzerauthentifizierung erstellen. Stellen Sie sicher, dass die Benutzer-ID, mit der das Apply-Programm ausgeführt wird, die Kennwortdatei lesen kann.

Gehen Sie wie folgt vor, um eine Kennwortdatei zu erstellen:

1. Wechseln Sie in einem Eingabeaufforderungsfenster von Windows NT in das Verzeichnis C:\scripts.
2. Erstellen Sie in diesem Verzeichnis eine neue Datei namens DEPTQUAL.PWD. Sie können diese Datei mit einem beliebigen Texteditor, z. B. Editor, erstellen. Die Namenskonvention für die Kennwortdatei lautet: *applyqual.pwd*. Dabei ist *applyqual* eine Zeichenfolge, die der Groß-/Kleinschreibung und dem Wert des Apply-Qualifikationsmerkmals entsprechen *muss*, das beim Erstellen der Subskriptionsgruppe verwendet wurde. Für dieses Szenario lautet das Apply-Qualifikationsmerkmal DEPTQUAL.

Anmerkung: Die Namenskonvention von Version 5 von DB2 DataPropagator wird ebenfalls unterstützt.

3. Der Inhalt der Kennwortdatei hat das folgende Format:

```
SERVER=server USER=benutzer-id PWD=kennwort
```

Dabei gilt Folgendes:

server

Der Name des Quellen-, Ziel- oder Steuerungs-Servers, so wie er in der Subskriptionsgruppentabelle erscheint. Für dieses Szenario lauten diese Namen SAMPLE und COPYDB.

benutzer-id

Die Benutzer-ID, die Sie zum Verwalten dieser Datenbank verwenden wollen. Bei den Betriebssystemen Windows NT und UNIX wird bei diesem Wert zwischen Groß-/Kleinschreibung unterschieden.

kennwort

Das Kennwort zur Benutzer-ID. Bei den Betriebssystemen Windows NT und UNIX wird bei diesem Wert zwischen Groß-/Kleinschreibung unterschieden.

Nehmen Sie in diese Datei keine Leer- oder Kommentarzeilen auf. Fügen Sie nur Server-Namen, Benutzer-IDs und Kennwörter hinzu.

4. Der Inhalt der Kennwortdatei sollte ähnlich wie das folgende Beispiel aussehen:

```
SERVER=SAMPLE USER=subina PWD=subpw  
SERVER=COPYDB USER=subina PWD=subpw
```

Weitere Informationen zur DB2-Authentifizierung und -Sicherheit finden Sie im Handbuch *IBM DB2 Systemverwaltung*.

Kapitel 5. Planung der Replikationsumgebung

Tabellen- und Spaltennamen

Die Replikation unterstützt Leerzeichen in Tabellen- und Spaltennamen nicht.

DATALINK-Replikation

DATALINK-Replikation ist für Solaris im FixPak 1 für Version 7.1 verfügbar. Sie erfordert einen FTP-Dämon, der auf den Quellen-DATALINK- und den Ziel-DATALINK-Dateisystemen ausgeführt wird und den Befehl MDTM (modtime) unterstützt, der die Zeit der letzten Änderung für eine bestimmte Datei anzeigt. Wenn Sie Version 2.6 oder eine andere Version des Solaris-Betriebssystems verwenden, die keine FTP-Unterstützung für MDTM umfasst, benötigen Sie weitere Software wie WU-FTPD.

Sie können DATALINK-Spalten nicht zwischen DB2-Datenbanken unter AS/400 und DB2-Datenbanken auf anderen Plattformen replizieren.

Auf der AS/400-Plattform wird die Replikation des Kommentarattributs von DATALINK-Werten nicht unterstützt.

Wenn Sie AIX 4.2 ausführen, müssen Sie vor der Ausführung des Standard-Benutzer-Exit-Programms (ASNDLCOPY) die vorläufige Programmkorrektur für APAR IY03101 (AIX 4210-06 RECOMMENDED MAINTENANCE FOR AIX 4.2.1) installieren. Diese vorläufige Programmkorrektur enthält eine Y2K-Korrektur für den Befehl "modtime/MDTM" im FTP-Dämon. Sie können die Korrektur prüfen, indem Sie die letzte von dem Befehl "modtime <datei>" zurückgegebene Änderungszeit anzeigen. Dabei ist <datei> eine Datei, die nach dem 1. Januar 2000 geändert worden ist.

Wenn die Zieltabelle eine externe CCD-Tabelle ist, ruft DB2 DataPropagator die Routine ASNDLCOPY auf, um DATALINK-Dateien zu replizieren. Die neuesten Informationen zur Verwendung der Programme ASNDLCOPY und ASNDLCOPYD finden Sie im Prolog des Quellencodes der jeweiligen Programme. Es gelten folgende Einschränkungen:

- Interne CCD-Tabellen können DATALINK-Bezugswerte, aber keine DATALINK-Werte enthalten.
- Komprimierte externe CCD-Tabellen können DATALINK-Werte enthalten.
- Nicht komprimierte CCD-Zieltabellen können keine DATALINK-Spalten enthalten.
- Wenn der Quellen-Server mit dem Ziel-Server identisch ist, darf die Subskriptionsgruppe keine Elemente mit DATALINK-Spalten enthalten.

LOB-Einschränkungen

Komprimierte interne CCD-Tabellen können keine Verweise auf LOB-Spalten oder LOB-Bezugswerte haben.

Planen der Replikation

Der Abschnitt "Konnektivität" im Kapitel "Planung der Replikationsumgebung" sollte noch folgenden Text enthalten:

Wenn das Apply-Programm keine Verbindung zum Steuerungs-Server herstellen kann, wird das Apply-Programm beendet.

Wenn Sie Datenblockung für AS/400 verwenden, müssen Sie sicherstellen, dass die Gesamtmenge der Replikationsdaten während des Intervalls "4 Millionen Zeilen" nicht übersteigt, nicht "4 MB", wie im Abschnitt "Datenblockung bei hohem Änderungsaufkommen" angegeben.

Kapitel 6. Einrichten der Replikationsumgebung

Vorbedingung für beliebige Tabellenreplikation

Wenn Sie eine beliebige Tabellenreplikation (Update-Anywhere Replication) mit Konflikterkennung und mehr als 150 Elementen in einer Subskriptionsgruppe einrichten wollen, müssen Sie folgende DDL ausführen, um auf dem Steuerungs-Server die Tabelle ASN.IBMSNAP_COMPENSATE zu erstellen:

```
CREATE TABLE ASN.IBMSNAP_COMPENSATE (  
    APPLY_QUAL char(18) NOT NULL,  
    MEMBER SMALLINT,  
    INTENTSEQ CHAR(10) FOR BIT DATA,  
    OPERATION CHAR(1));
```

Einrichten Ihrer Replikationsumgebung

Im Abschnitt "Anpassen von CD-Tabellennamen, Indexnamen und Tabellenbereichsnamen" wird gesagt, dass sich die Datei DPREPL.DFT im Verzeichnis \sql1ib\bin oder \sql1ib\java befindet. Dies ist nicht korrekt. DPREPL.DFT befindet sich im Verzeichnis \sql1ib\cc.

Die Beschreibung der Sicherungsbegrenzung im Unterabschnitt "Angabe von Anpassungsparametern für das Capture-Programm" in Abschnitt "Einrichten des Capture-Programms" in Kapitel 6 "Einrichten der Replikationsumgebung" sollte darauf hinweisen, dass die Sicherungsbegrenzung nur verwendet wird, um Zeilen zu entfernen, wenn „Capture“ neu gestartet wird oder wenn Sie den Befehl „Capture prune“ verwenden. Wenn Sie „Capture“ mit der Option „auto-pruning“ gestartet haben, verwendet der Befehl die Sicherungsbegrenzung zum Entfernen von Reihen nicht.

Kapitel 8. Fehlerbestimmung

Replication Analyzer funktioniert auf Windows 32-Bit-Systemen und unter AIX. Bei der Verwendung von Analyzer unter AIX müssen Sie sicherstellen, dass das Verzeichnis sql1ib/bin in Ihrer Umgebungsvariablen PATH vor /usr/local/bin angezeigt wird, damit keine Konflikte mit /usr/local/bin/analyze auftreten.

Replication Analyzer hat optional zwei zusätzliche Schlüsselwörter: CT und AT.

CT=*n* Zeigt nur diejenigen Einträge der Capture-Trace-Tabelle an, die weniger als *n* Tage alt sind. Dieses Schlüsselwort ist optional. Wenn Sie es nicht angeben, wird der Standardwert (7 Tage) verwendet.

AT=*n* Zeigt nur diejenigen Einträge der Apply-Prüfprotokolltabelle an, die weniger als *n* Tage alt sind. Dieses Schlüsselwort ist optional. Wenn Sie es nicht angeben, wird der Standardwert (7 Tage) verwendet.

Beispiel:

```
analyze mydb1 mydb2 f=mydirectory ct=4 at=2 deepcheck q=applyqual1
```

Für Replication Analyzer werden folgende Schlüsselwortinformationen aktualisiert:

deepcheck

Weist Replication Analyzer an, eine ausführlichere Analyse vorzunehmen, einschließlich Bereinigungsinformationen für CD- und UOW-Tabelle, Tabellenbereichspartitionierung von DB2 für OS/390 und Zusatzinformationen für die Komprimierung sowie einer Analyse der Zielindizes im Hinblick auf die Subskriptionsschlüssel, Subskriptionsdauer und SQL-Anweisungsfehler in der Subskriptionsgruppe. Die Analyse umfasst alle Server. Dieses Schlüsselwort ist optional.

lightcheck

Gibt an, dass folgende Informationen aus dem Bericht ausgeklammert werden sollen: alle Spaltendetails aus der Tabelle ASN.IBMSNAP_SUBS_COLS, Fehler, Anomalien oder Auslassungen bei der Subskription sowie falsche oder ineffiziente Indizes. Diese Informationsverringering spart Ressourcen und verkleinert die HTML-Ausgabedatei. Dieses Schlüsselwort ist optional und gegenseitig ausschließend mit dem Schlüsselwort "deepcheck".

Analyzer-Tools stehen in vorläufigen Programmkorrekturen für die Replikation auf AS/400-Plattformen zur Verfügung. Diese Tools erfassen Informationen zu Ihrer Replikationsumgebung und erzeugen eine HTML-Datei, die Sie zur Unterstützung bei der Fehlerbestimmung an Ihren IBM Ansprechpartner senden können. Laden Sie zum Abrufen der AS/400-Tools die entsprechende vorläufige Programmkorrektur (PTF) herunter (für das Produkt 5769DP2 müssen Sie beispielsweise PTF SF61798 oder deren neueste Version herunterladen).

Fügen Sie dem Abschnitt "Fehlerbehebung" das folgende Problem und seine Lösung hinzu:

Problem: Das Apply-Programm führt Schleifen aus, ohne Änderungen zu replizieren; die Apply-Prüfprotokolltabelle zeigt STATUS=2 an.

Die Subskriptionsgruppe enthält mehrere Quellentabellen. Zur Verbesserung der Bearbeitung von Detailpunkten für eine Quelle in der Gruppe wird eine interne CCD-Tabelle für diese Quellentabelle definiert, jedoch in einer anderen Subskriptionsgruppe. Die Quellentabelle wird aktualisiert, aber der Apply-Prozess, der die interne CCD-Tabelle füllt, wird asynchron ausgeführt (z. B. wird das Apply-Programm möglicherweise nicht gestartet oder ein Ereignis nicht ausgelöst, usw.). Das Apply-Programm, das Aktualisierungen von der Quellen- in die Zieltabelle repliziert, führt Schleifen aus, weil es darauf wartet, dass die interne CCD-Tabelle aktualisiert wird.

Starten Sie zum Stoppen der Schleifen das Apply-Programm (oder lösen Sie das Ereignis aus, das die Replikation verursacht) für die interne CCD-Tabelle. Das Apply-Programm füllt die interne CCD-Tabelle und ermöglicht dem Apply-Programm in der Warteschleife, Änderungen von allen Quellentabellen zu verarbeiten.

Eine ähnliche Situation könnte bei einer Subskriptionsgruppe auftreten, die Quellentabellen mit internen CCD-Tabellen enthält, die von mehreren Apply-Programmen gefüllt werden.

Kapitel 9. Capture und Apply für AS/400

Der Text unter "Anmerkung zur Arbeitsverwaltung" muss wie folgt lauten:

Sie können die Standarddefinitionen ändern oder eigene Definitionen bereitstellen. Wenn Sie eine eigene Beschreibung des Subsystems erstellen, müssen Sie es QZSNDPR nennen und in einer anderen Bibliothek als QDPR erstellen. Weitere Informationen zum Ändern dieser Definitionen finden Sie in der Veröffentlichung "OS/400 Work Management V4R3", SC41-5306.

Fügen Sie dem Abschnitt "Prüfen und Anpassen der Installation von DB2 DataPropagator für AS/400" Folgendes hinzu:

Wenn Sie aufgrund einer hohen Anzahl an Transaktionen große Probleme mit Zugriffskonflikten haben, können Sie den Standardwert für das Wartezeitlimit von 30 auf 120 erhöhen. Sie können den Job bei jedem Start von Capture ändern oder das Standardzeitlimit mit der folgenden Prozedur für alle Jobs ändern, die in Ihrem Subsystem ausgeführt werden.

1. Setzen Sie folgenden Befehl ab, um durch Kopieren von QGPL/QBATCH ein Objekt einer neuen Klasse zu erstellen:

```
CRTDUPOBJ OBJ(QBATCH) FROMLIB(QGPL) OBJTYPE(*CLS) TOLIB(QDPR) NEWOBJ(QZSNDPR)
```

2. Ändern Sie den Zeitlimitwert für die neu erstellte Klasse (beispielsweise in 300 Sekunden):

```
CHGCLS CLS(QDPR/QZSNDPR) DFTWAIT(300)
```

3. Ändern Sie den Route-Eintrag in der Subsystembeschreibung QDPR/QZSNDPR, damit die neue Klasse verwendet wird:
CHGRTGE SBSO(QDPR/QZSNDPR) SEQNBR(9999) CLS(QDPR/QZSNDPR)

Der Unterabschnitt "Verwenden der Exit-Routine DLTJRNRCV (Delete Journal Receiver)" des Abschnitts "Das Journal" in Kapitel 9, "Capture und Apply für AS/400", sollte den folgenden Satz enthalten: Wenn Sie die Registrierung für die gelöschte Journalempfänger-Exit-Routine entfernen, stellen Sie sicher, dass alle Journale, die als Quellentabellen verwendet werden, den Eintrag DLTRCV(*NO) haben.

Die Parameter des Befehls ADDEXITPGM unter der Überschrift "Registrieren der Exit-Routine DLTJRNRCV (Delete Journal Receiver) auf erweiterten Systemen" müssen wie folgt lauten:

```
ADDEXITPGM EXITPNT(QIBM_QJO_DLT_JRNRCV)
            FORMAT(DRCV0100)
            PGM(QDPR/QZSNDREP)
            PGMNBR(*LOW)
            CRTEXITPNT(*NO)
            PGMFTA(65535 10 QSYS)
```

Kapitel 10. Capture und Apply für OS/390

In Kapitel 10 wurden die folgenden Absätze aktualisiert:

Vorbedingungen für DB2 DataPropagator für OS/390

Sie müssen DB2 für OS/390 Version 5, DB2 für OS/390 Version 6 oder DB2 für OS/390 Version 7 haben, um DB2 DataPropagator für OS/390 Version 7 (V7) auszuführen.

UNICODE- und ASCII-Schemen für Codeumsetzung unter OS/390

DB2 DataPropagator für OS/390 V7 unterstützt die UNICODE- und ASCII-Schemen für Codeumsetzung. Um die neuen Schemen für Codeumsetzung ausnutzen zu können, müssen Sie DB2 für OS/390 V7 haben, und Sie müssen Ihre Quelle, Ihr Ziel und Ihre Steuertabellen von DB2 DataPropagator manuell erstellen oder konvertieren, wie in den folgenden Abschnitten beschrieben wird. Ihre vorhandene Replikationsumgebung arbeitet allerdings mit DB2 DataPropagator für OS/390 V7, auch wenn Sie keine Schemen für Codeumsetzung ändern.

Auswählen eines Schemas für Codeumsetzung

Wenn Ihre Quellen-, CD- und Zieltabellen dasselbe Schema für Codeumsetzung verwenden, können Sie den Aufwand zur Datenkonvertierung in Ihrer Replikationsumgebung minimieren. Wenn Sie Schemen für Codeumsetzung für die Tabellen auswählen, befolgen Sie die Regel für eine einzelne ID für codierten Zeichensatz: Zeichendaten in einem Tabellenbereich können in ASCII, UNICODE oder EBCDIC codiert werden. Alle Tabellen in einem Tabellenbereich müssen dasselbe Schema für Codeumsetzung verwenden. Das Schema für Codeumsetzung für alle Tabellen in einer SQL-Anweisung muss dasselbe sein. Außerdem müssen alle Tabellen, die Sie in Sichten und Verknüpfungen verwenden, dasselbe Schema für Codeumsetzung verwenden.

Wenn Sie die Regel für eine einzelne ID für codierten Zeichensatz nicht befolgen, entdeckt DB2 die Verletzung und gibt SQLCODE -873 während des Bindens oder der Ausführung zurück. Welche Tabellen in ASCII oder UNICODE sein sollten, hängt von Ihrer Client-/Serverkonfiguration ab. Befolgen Sie besonders die folgenden Regeln, wenn Sie ein Schema für Codeumsetzung für die Tabellen auswählen:

- Quellen- oder Zieltabellen in DB2 für OS/390 können EBCDIC, ASCII oder UNICODE verwenden. Sie können aus oder in Tabellen kopiert werden, die dasselbe Schema für Codeumsetzung oder ein anderes Schema eines unterstützten Datenbankverwaltungssystems (DB2-Systemfamilie oder nicht DB2 mit DataJoiner) haben.
- Auf einem Quellenserver von DB2 für OS/390 müssen alle CD-, UOW-, Register- und Löschsteuertabellen auf demselben Server dasselbe Schema für Codeumsetzung verwenden. Geben Sie das Schema für Codeumsetzung immer ausdrücklich an, um die Konsistenz sicherzustellen.
- Alle Steuertabellen (ASN.IBMSNAP_SUBS_xxxx) auf demselben Steuerungsserver müssen dasselbe Schema für Codeumsetzung verwenden.
- Andere Steuertabellen können beliebige Schemen für Codeumsetzung verwenden. Es wird allerdings empfohlen, dass die Tabelle ASN.IBMSNAP_CRITSEC in EBCDIC bleibt.

Einstellen von Schemen für Codeumsetzung

Ändern Sie die SQL, die zum Generieren der Tabellen verwendet wird, wie folgt, um das passende Schema für Codeumsetzung für Tabellen anzugeben:

- Erstellen Sie neue Quellen- und Zieltabellen mit dem passenden Schema für Codeumsetzung, oder ändern Sie das Schema für Codeumsetzung der vorhandenen Ziel- und Quellentabellen. Es wird empfohlen, dass Sie die Programme „Capture“ und „Apply“ stoppen, bevor Sie das Schema für Codeumsetzung von vorhandenen Tabellen ändern. Danach sollten Sie einen Kaltstart für das Programm „Capture“ ausführen und das Programm „Apply“ erneut starten. Gehen Sie wie folgt vor, um das Schema für Codeumsetzung von vorhandenen Tabellen zu ändern:

1. Verwenden Sie das Dienstprogramm REORG, um die vorhandene Tabelle zu kopieren.
2. Löschen Sie die vorhandene Tabelle.
3. Erstellen Sie die Tabelle erneut, und geben Sie dabei das neue Schema für Codeumsetzung an.
4. Verwenden Sie das Dienstprogramm LOAD, um die alten Daten in die neue Tabelle zu laden.

Weitere Informationen zu den Dienstprogrammen LOAD und REORG finden Sie in *DB2 Universal Database for OS/390 Utility Guide and Reference*.

- Erstellen Sie neue Steuertabellen mit dem passenden Schema für Codeumsetzung, oder ändern Sie das Schema für Codeumsetzung für vorhandene Steuertabellen.

Die Datei DPCNTL.MVS wird mit DB2 für OS/390 im Verzeichnis `sql1lib\samples\repl` ausgeliefert und enthält einige Anweisungen CREATE TABLE, die die Steuertabellen erstellen. Fügen Sie für die Tabellen, die in ASCII oder UNICODE sein müssen (z. B. ASN.IBMSNAP_REGISTER und ASN.IBMSNAP_PRUNCNTL), das Schlüsselwort CCSID ASCII oder CCSID UNICODE hinzu, wie in dem folgenden Beispiel gezeigt wird:

```
CREATE TABLE ASN.IBMSNAP_PRUNCNTL (
  TARGET_SERVER      CHAR( 18)          NOT NULL,
  TARGET_OWNER      CHAR( 18)          NOT NULL,
  TARGET_TABLE      CHAR( 18)          NOT NULL,
  SYNCHTIME         TIMESTAMP,
  SYNCHPOINT        CHAR( 10)          FOR BIT DATA,
  SOURCE_OWNER      CHAR( 18)          NOT NULL,
  SOURCE_TABLE      CHAR( 18)          NOT NULL,
  SOURCE_VIEW_QUAL  SMALLINT           NOT NULL,
  APPLY_QUAL        CHAR( 18)          NOT NULL,
  SET_NAME          CHAR( 18)          NOT NULL,
  CNTL_SERVER       CHAR( 18)          NOT NULL,
  TARGET_STRUCTURE  SMALLINT           NOT NULL,
  CNTL_ALIAS        CHAR( 8)
) CCSID UNICODE
DATA CAPTURE CHANGES          IN TSSNAP02;
```

Verwenden Sie die Dienstprogramme REORG und LOAD, um vorhandene Steuer- und CD-Tabellen zu verändern.

- Wenn Sie neue Replikationsquellen oder Subskriptionsgruppen erstellen, ändern Sie die vom Verwaltungs-Tool generierte SQL-Datei, um das passende Schema für Codeumsetzung anzugeben. Die SQL hat einige Anweisungen CREATE TABLE, die verwendet werden, um die CD- und Zieltabellen für die Replikationsquelle und die Subskriptionsgruppe zu erstellen. Fügen Sie das Schlüsselwort CCSID ASCII oder CCSID UNICODE hinzu, wo es benötigt wird. Beispiel:

```
CREATE TABLE user1.cdtable1 (
  employee_name  varchar,
  employee_age   decimal
) CCSID UNICODE;
```

Weitere Informationen zu IDs für codierten Zeichensatz finden Sie im Handbuch *DB2 UDB for OS/390 SQL Reference*.

Kapitel 11. Capture und Apply für UNIX-Plattformen

Einstellen der Umgebungsvariablen für Capture und Apply unter UNIX und Windows

Wenn Sie eine Quelldatenbank mit einer anderen Codepage als der Standard-Codepage erstellt haben, setzen Sie die Umgebungsvariable DB2CODEPAGE auf diese Codepage. Schlagen Sie weitere Informationen zum Ermitteln von Codepage-Werten im Handbuch *DB2 Systemverwaltung* nach, bevor Sie DB2CODEPAGE festlegen. Der Befehl „Capture“ muss in derselben Codepage ausgeführt werden, wie die Datenbank, für die er Daten erfasst. DB2 ermittelt die Codepage für Capture aus der Umgebung, die während der Ausführung von Capture aktiv ist. Wenn DB2CODEPAGE nicht festgelegt ist, ermittelt DB2 den Codepage-Wert des Betriebssystems. Der Wert, der vom Betriebssystem ermittelt wird, ist für Capture korrekt, wenn Sie die Standard-Codepage bei der Erstellung der Datenbank verwendet haben.

Kapitel 14. Tabellenstrukturen

Fügen Sie der Beschreibung der Statusspalte für den Wert "2" folgenden Satz hinzu:

Wenn Sie beim Verwenden interner CCD-Tabellen mehrmals den Wert "2" in der Statusspalte der Apply-Prüfprotokolltabelle erhalten, lesen Sie in "Kapitel 8: Fehlerbestimmung," den Abschnitt "Problem: Das Apply-Programm führt Schleifen aus, ohne Änderungen zu replizieren; die Apply-Prüfprotokolltabelle zeigt STATUS=2 an".

Kapitel 15. Nachrichten der Programme Capture und Apply

Die Nachricht ASN0017E sollte wie folgt lauten:

ASN0017E

Das Capture-Programm hat einen schwerwiegenden internen Fehler festgestellt und konnte die korrekte Fehlernachricht nicht anzeigen. Der Routinename ist "routine". Der Rückkehrcode ist "rückkehrcode".

Nachricht ASN1027S muss hinzugefügt werden:

ASN1027S

Es wurden zu viele große Objekte (LOBs) angegeben. Fehlercode: "<fehlercode>".

Erläuterung: Es wurden zu viele Spalten für große Objekte (BLOB, CLOB oder DBCLOB) für ein Element einer Subskriptionsgruppe angegeben. Die maximal zulässige Spaltenanzahl ist 10.

Benutzeraktion: Entfernen Sie die überzähligen Spalten für große Objekte aus dem Element einer Subskriptionsgruppe.

Nachricht ASN1048E muss folgendermaßen lauten:

ASN1048E

Die Ausführung eines Apply-Zyklus ist fehlgeschlagen. Alle Einzelangaben sind in der Apply-Prüflistentabelle zu finden: "<text>"

Erläuterung: Ein Apply-Zyklus ist fehlgeschlagen. In der Nachricht bezeichnet "<text>" den "<zielsever>", "<zieleigner, zieltabelle, anwsg-nummer>" und "<strg-server>".

Benutzeraktion: Stellen Sie anhand der APPERRM-Felder in der Prüfprotokolltabelle fest, warum der Apply-Zyklus fehlgeschlagen ist.

Anhang A. Starten der Programme Capture und Apply aus einer Anwendung heraus

Die Kommentare zu der Beispielroutine, mit der die Programme Capture und Apply gestartet werden, enthalten einige Fehler. Der Code in dem Beispiel ist jedoch korrekt. Der letzte Teil des Beispiels bezieht sich auf die Apply-Parameter, obwohl die Kommentare angeben, dass er sich auf die Capture-Parameter bezieht.

Beispiele der Apply- und Capture-API und ihrer jeweiligen Makefiles finden Sie in folgenden Verzeichnissen:

Für NT - sql1lib\samples\repl
Für UNIX - sql1lib/samples/repl

System Monitor Guide and Reference

db2ConvMonStream

In the Usage Notes, the structure for the snapshot variable datastream type SQLM_ELM_SUBSECTION should be sqlm_subsection.

Troubleshooting Guide

Starting DB2 on Windows 95, Windows 98, and Windows ME When the User Is Not Logged On

For a `db2start` command to be successful in a Windows 95, Windows 98, or Windows Millennium Edition (ME) environment, you must either:

- Log on using the Windows logon window or the Microsoft Networking logon window
- Issue the **db2logon** command (see note 1 auf Seite 184 for information about the **db2logon** command).

In addition, the user ID that is specified either during the logon or for the **db2logon** command must meet DB2's requirements (see note 2 auf Seite 185).

When the **db2start** command starts, it first checks to see if a user is logged on. If a user is logged on, the **db2start** command uses that user's ID. If a user is not logged on, the **db2start** command checks whether a **db2logon** command has been run, and, if so, the **db2start** command uses the user ID that was specified for the **db2logon** command. If the **db2start** command cannot find a valid user ID, the command terminates.

During the installation of DB2 Universal Database Version 7 on Windows 95, Windows 98, and Windows ME, the installation software, by default, adds a shortcut to the Startup folder that runs the **db2start** command when the system is booted (see note 1 auf Seite 184 for more information). If the user of the system has neither logged on nor issued the **db2logon** command, the **db2start** command will terminate.

If you or your users do not normally log on to Windows or to a network, you can hide the requirement to issue the **db2logon** command before a **db2start** command by running commands from a batch file as follows:

1. Create a batch file that issues the **db2logon** command followed by the **db2start.exe** command. For example:

```
@echo off
db2logon db2local /p:password
db2start
cls
exit
```

2. Name the batch file `db2start.bat`, and store it in the `/bin` directory that is under the drive and path where you installed DB2. You store the batch file in this location to ensure that the operating system can find the path to the batch file.

The drive and path where DB2 is installed is stored in the DB2 registry variable `DB2PATH`. To find the drive and path where you installed DB2, issue the following command:

```
db2set -g db2path
```

Assume that the `db2set` command returns the value `c:\sql11b`. In this situation, you would store the batch file as follows:

```
c:\sql11b\bin\db2start.bat
```

3. To start DB2 when the system is booted, you should run the batch file from a shortcut in the Startup folder. You have two options:

- Modify the shortcut that is created by the DB2 installation program to run the batch file instead of **db2start.exe**. In the preceding example, the shortcut would now run the `db2start.bat` batch file. The shortcut that is created by DB2 installation program is called **DB2 - DB2.lnk**, and is located in `c:\WINDOWS\Start Menu\Programs\Start\DB2 - DB2.lnk` on most systems.
- Add your own shortcut to run the batch file, and delete the shortcut that is added by the DB2 installation program. Use the following command to delete the DB2 shortcut:

```
del "C:\WINDOWS\Start Menu\Programs\Startup\DB2 - DB2.lnk"
```

If you decide to use your own shortcut, you should set the *close on exit* attribute for the shortcut. If you do not set this attribute, the DOS command prompt is left in the task bar even after the **db2start** command has successfully completed. To prevent the DOS window from being opened during the **db2start** process, you can create this shortcut (and the DOS window it runs in) set to run minimized.

Anmerkung: As an alternative to starting DB2 during the boot of the system, DB2 can be started prior to the running of any application that uses DB2. See note 5 auf Seite 185 for details.

If you use a batch file to issue the **db2logon** command before the **db2start** command is run, and your users occasionally log on, the **db2start** command will continue to work, the only difference being that DB2 will use the user ID of the logged on user. See note 1 for additional details.

Anmerkungen:

1. The **db2logon** command simulates a user logon. The format of the `db2logon` command is:

```
db2logon userid /p:password
```

The user ID that is specified for the command must meet the DB2 naming requirements (see note 2 auf Seite 185 for more information). If the command is issued without a user ID and password, a window opens to prompt the user for the user ID and password. If the only parameter provided is a user ID, the user is not prompted for a password; under certain conditions a password is required, as described below.

The user ID and password values that are set by the **db2logon** command are only used if the user did not log on using either the Windows logon window or the Microsoft Networking logon window. If the user has logged on, and a **db2logon** command has been issued, the user ID from the **db2logon** command is used for all DB2 actions, but the password specified on the **db2logon** command is ignored.

When the user has not logged on using the Windows logon window or the Microsoft Networking logon window, the user ID and password that are provided through the **db2logon** command are used as follows:

- The **db2start** command uses the user ID when it starts, and does not require a password.
- In the absence of a high-level qualifier for actions like creating a table, the user ID is used as the high-level qualifier. For example:
 - a. If you issue the following: `db2logon db2local`

b. Then issue the following: create table tabl

The table is created with a high-level qualifier as db2local.tabl.

You should use a user ID that is equal to the schema name of your tables and other objects.

- When the system acts as client to a server, and the user issues a **CONNECT** statement without a user ID and password (for example, **CONNECT TO TEST**) and authentication is set to server, the user ID and password from the **db2logon** command are used to validate the user at the remote server. If the user connects with an explicit user ID and password (for example, **CONNECT TO TEST USER *userID* USING *password***), the values that are specified for the **CONNECT** statement are used.
2. In Version 7, the user ID that is either used to log on or specified for the **db2logon** command must conform to the following DB2 requirements:
 - It cannot be any of the following: **USERS**, **ADMINS**, **GUESTS**, **PUBLIC**, **LOCAL**, or any SQL reserved word that is listed in the *SQL Reference*.
 - It cannot begin with: **SQL**, **SYS** or **IBM**
 - Characters can include:
 - A through Z (Windows 95, Windows 98, and Windows ME support case-sensitive user IDs)
 - 0 through 9
 - @, #, or \$
 3. You can prevent the creation of the **db2start** shortcut in the Startup folder during a customized interactive installation, or if you are performing a response file installation and specify the **DB2.AUTOSTART=NO** option. If you use these options, there is no **db2start** shortcut in the Startup folder, and you must add your own shortcut to run the **db2start.bat** file.
 4. On **Windows 98** and **Windows ME** an option is available that you can use to specify a user ID that is always logged on when Windows 98 or Windows ME is started. In this situation, the Windows logon window will not appear. If you use this option, a user is logged on and the **db2start** command will succeed if the user ID meets DB2 requirements (see note 2 for details). If you do not use this option, the user will always be presented with a logon window. If the user cancels out of this window without logging on, the **db2start** command will fail unless the **db2logon** command was previously issued, or invoked from the batch file, as described above.
 5. If you do not start DB2 during a system boot, DB2 can be started by an application. You can run the **db2start.bat** file as part of the initialization of applications that use DB2. Using this method, DB2 will only be started when the application that will use it is started. When the user exits the application, a **db2stop** command can be issued to stop DB2. Your business applications can start DB2 in this way, if DB2 is not started during the system boot.

To use the DB2 Synchronizer application or call the synchronization APIs from your application, DB2 must be started if the scripts that are download for execution contain commands that operate either against a local instance or a local database. These commands can be in database scripts, instance scripts, or embedded in operating system (OS) scripts. If an OS script does not contain Command Line Processor commands or DB2 APIs that use an instance or a database, it can be run without DB2 being started. Because it may be difficult to tell in advance what commands will be run from your scripts during the synchronization process, DB2 should normally be started before synchronization begins.

If you are calling either the **db2sync** command or the synchronization APIs from your application, you would start DB2 during the initialization of your

application. If your users will be using the DB2 Synchronizer shortcut in the DB2 for Windows folder to start synchronization, the DB2 Synchronization shortcut must be modified to run a `db2sync.bat` file. The batch file should contain the following commands to ensure that DB2 is running before synchronization begins:

```
@echo off
db2start.bat
db2sync.exe
db2stop.exe
cls
exit
```

In this example, it is assumed that the `db2start.bat` file invokes the **db2logon** and **db2start** commands as described above. If you decide to start DB2 when the application starts, ensure that the installation of DB2 does not add a shortcut to the Startup folder to start DB2. See note 3 auf Seite 185 for details.

Chapter 2. Troubleshooting the DB2 Universal Database Server

Under the „Locking and Deadlocks“ section, under the „Applications Slow or Appear to Hang“ subsection, change the description under „Lock waits or deadlocks are not caused by next key locking“ to :

Next key locking guarantees Repeatable Read (RR) isolation level by automatically locking the next key for all INSERT and DELETE statements and the next higher key value above the result set for SELECT statements. For UPDATE statements that alter key parts of an index, the original index key is deleted and the new key value is inserted. Next key locking is done on both the key insertion and key deletion. It is required to guarantee ANSI and SQL92 standard RR, and is the DB2 default.

Examine snapshot information for the application. If the problem appears to be with next key locking, you can set the `DB2_RR_TO_RS` option on if none of your applications rely on Repeatable Read (RR) behavior and it is acceptable for scans to skip over uncommitted deletes.

When `DB2_RR_TO_RS` is on, RR behavior cannot be guaranteed for scans on user tables because next key locking is not done during index key insertion and deletion. Catalog tables are not affected by this option. The other change in behavior is that with `DB2_RR_TO_RS` on, scans will skip over rows that have been deleted but not committed, even though the row may have qualified for the scan.

For example, consider the scenario where transaction A deletes the row with `column1=10` and transaction B does a scan where `column1>8` and `column1<12`.

With `DB2_RR_TO_RS` off, transaction B will wait for transaction A to commit or rollback. If it rolls back, the row with `column1=10` will be included in the result set of transaction B's query.

With `DB2_RR_TO_RS` on, transaction B will not wait for transaction A to commit or rollback. It will immediately receive query results that do not include the deleted row.

Do not use this option if you require ANSI and SQL92 standard RR or if you do not want scans to skip uncommitted deletes.

Verwenden von DB2 Universal Database auf 64-Bit-Plattformen

Kapitel 5. Konfiguration

LOCKLIST

Die folgenden Informationen sollten Tabelle 2 hinzugefügt werden.

Parameter	Vorherige Obergrenze	Aktuelle Obergrenze
LOCKLIST	60000	524288

shmsys:shminfo_shmmax

DB2-Benutzer des 64-Bit-Betriebssystems Solaris sollten gegebenenfalls den Wert von `shmsys:shminfo_shmmax` in `/etc/system` erhöhen, um großen gemeinsam genutzten Datenbankspeicher zuordnen zu können. Das Handbuch *DB2 für UNIX Einstieg* empfiehlt, diesen Parameter auf 90% des physischen RAM in der Maschine (in Byte) einzustellen. Diese Empfehlung gilt auch für 64-Bit-Implementierungen.

Es gibt jedoch ein Problem mit der folgenden Empfehlung im Handbuch *DB2 für UNIX Einstieg*: Wenn ein Benutzer bei 32-Bit-Systemen mit mehr als 4 GB RAM (bis zu 64 GB ist insgesamt im Solaris-Betriebssystem möglich) den Wert von `shmmax` auf einen größeren Wert als 4 GB einstellt und einen 32-Bit-Kernel verwendet, beachtet der Kernel nur die unteren 32 Bit der Zahl, was manchmal zu einem sehr kleinen Wert für

`shmmax`

führt.

Kapitel 6. Einschränkungen

Derzeit wird LDAP auf 64-Bit-Betriebssystemen nicht unterstützt.

32-Bit- und 64-Bit-Datenbanken können nicht im selben Pfad erstellt werden. Wenn z. B. eine 32-Bit-Datenbank in `<meinpfad>` vorhanden ist, schlägt der folgende Befehl:

```
db2 create db <meinedb> on <meinpfad>
```

mit "SQL10004C Beim Zugriff auf das Datenbankverzeichnis trat ein E/A-Fehler auf" fehl, wenn er von einem 64-Bit-Exemplar abgesetzt wird.

XML Extender Verwaltung und Programmierung

Release-Informationen für IBM DB2 XML Extender finden Sie auf der Website von DB2 XML:

<http://www-4.ibm.com/software/data/db2/extenders/xmlxt/library.html>

MQSeries

In diesem Abschnitt wird beschrieben, wie DB2 und MQSeries zur Erstellung von Anwendungen verwendet werden können, die sowohl Nachrichtenübertragung als auch Datenbankzugriff ermöglichen. Der Schwerpunkt liegt hierbei auf einer Reihe von Funktionen, die benutzerdefinierten Funktionen (UDFs - User-Defined Functions) ähneln und in DB2 Universal Database Version 7.2 optional aktiviert werden können. Durch diese Basisfunktionen können zahlreiche Anwendungen unterstützt werden, von der einfachen Ereignisaufzeichnung bis zum Data Warehousing.

Weitere Informationen zu Data Warehousing-Anwendungen finden Sie in „Integration von MQSeries in die Data Warehouse-Zentrale“ auf Seite 251.

Installation und Konfiguration für die DB2 MQSeries-Funktionen

In diesem Abschnitt wird beschrieben, wie eine DB2-Umgebung zur Verwendung der DB2 MQSeries-Funktionen konfiguriert werden muss. Nach erfolgreicher Durchführung der folgenden Prozedur können die DB2 MQSeries-Funktionen in SQL aufgerufen werden. Eine Beschreibung der Funktionen finden Sie im Abschnitt zum Handbuch „SQL Reference“ in den Release-Informationen. Zusätzliche Informationen, die aktuelle Dokumentation sowie Tipps und Hinweise finden Sie unter folgender Adresse:

<http://www.ibm.com/software/data/integration/MQSeries> .

Die Konfiguration und Aktivierung der DB2 MQSeries-Funktionen erfolgt grundsätzlich wie folgt:

1. Installieren Sie MQSeries.
2. Installieren Sie MQSeries AMI.
3. Aktivieren und konfigurieren Sie die DB2 MQSeries-Funktionen.

Zur Verwendung der P/S-Funktionen (Publish/Subscribe - Veröffentlichen/Teilnehmen), die innerhalb der DB2 MQSeries-Funktionen zur Verfügung stehen, müssen entweder MQSeries Integrator oder die P/S-Funktionen von MQSeries installiert werden. Informationen zu MQSeries Integrator finden Sie unter <http://www.ibm.com/software/ts/mqseries/integrator> . Informationen zur P/S-Funktionalität von MQSeries finden Sie unter <http://www.ibm.com/software/ts/mqseries/txppacs> („category 3“).

Installieren von MQSeries

Zunächst muss sichergestellt werden, dass MQSeries Version 5.2 auf dem DB2-Server installiert ist. Wenn dies der Fall ist, können Sie mit dem nächsten Schritt („Installieren von MQSeries AMI“) fortfahren. DB2 Version 7.2 enthält eine Kopie des MQSeries-Servers, der mit DB2 verwendet werden muss. Plattformspezifische Anweisungen zum Installieren von MQSeries oder Aktualisieren einer vorhandenen MQSeries-Installation finden Sie in den zugehörigen Einstiegshandbüchern unter folgender Adresse:

<http://www.ibm.com/software/ts/mqseries/library/manuals>. Stellen Sie sicher, dass für den Installationsprozess ein Standard-Warteschlangenmanager definiert wird.

Installieren von MQSeries AMI

Der nächste Schritt besteht in der Installation von MQSeries Application Messaging Interface (AMI). Dabei handelt es sich um eine Erweiterung der Programmierschnittstellen von MQSeries, wobei eine klare Trennung von verwaltungs- und programmierbezogenen Tasks vorgenommen wird. Für die DB2 MQSeries-Funktionen ist die Installation dieser Schnittstelle erforderlich. Wenn MQSeries AMI bereits auf dem DB2-Server installiert ist, fahren Sie mit dem nächsten Schritt („Aktivieren der DB2 MQSeries-Funktionen“) fort. Wenn MQSeries AMI nicht installiert ist, können Sie das Produkt über das mit DB2 7.2 gelieferte Installationspaket installieren, oder eine AMI-Kopie von der MQSeries-Website für Unterstützungspakete unter <http://www.ibm.com/software/ts/mqseries/txppacs> herunterladen. Die AMI befindet sich unter „Category 3 – Product Extensions“. Der Einfachheit halber gehört eine Kopie von MQSeries AMI zum Lieferumfang von DB2. Diese Datei befindet sich im Verzeichnis `sql1lib/cfg`. Der Name der Datei ist betriebssystemabhängig:

AIX Version ab 4.3	ma0f_ax.tar.Z
HP-UX	ma0f_hp.tar.Z
Solaris-Betriebsumgebung	ma0f_sol7.tar.Z
Windows (32-Bit)	ma0f_nt.zip

Führen Sie den normalen AMI-Installationsprozess gemäß den Beschreibungen in der Readme-Datei zu AMI durch, die im komprimierten Installationsimage enthalten ist.

Aktivieren der DB2 MQSeries-Funktionen

In diesem Schritt wird eine Datenbank für die DB2 MQSeries-Funktionen konfiguriert und aktiviert. Das Dienstprogramm **enable_MQFunctions** ist ein flexibler Befehl, der zunächst sicherstellt, dass die richtige MQSeries-Umgebung eingerichtet wurde. Dann wird eine Standardkonfiguration für die DB2 MQSeries-Funktionen erstellt und installiert. Anschließend wird die angegebene Datenbank mit diesen Funktionen aktiviert, und das ordnungsgemäße Funktionieren der Konfiguration wird bestätigt.

1. Fahren Sie für Windows NT oder Windows 2000 mit Schritt 5 fort.
2. Richten Sie Gruppen unter UNIX ein. Wenn Sie die Funktionen unter UNIX aktivieren, müssen der MQSeries-Gruppe `mqm` zuerst der DB2-Exemplareigner (häufig `db2inst1`) und die den abgeschirmten UDFs (häufig `db2fenc1`) zugeordnete Benutzer-ID hinzugefügt werden. Dies ist erforderlich, damit die DB2-Funktionen auf MQSeries zugreifen können.
3. Setzen Sie die DB2-Umgebungsvariablen unter UNIX. Fügen Sie die Umgebungsvariable `AMT_DATA_PATH` der von DB2 verarbeitbaren Liste hinzu. Die Datei `$INSTHOME/sql1lib/profile.env` kann editiert werden. Fügen Sie `DB2ENVLIST` die Umgebungsvariable `AMT_DATA_PATH` hinzu. Der Befehl **db2set** kann ebenfalls verwendet werden.
4. Starten Sie unter UNIX das Datenbankexemplar erneut. Das Datenbankexemplar muss erneut gestartet werden, damit die Änderungen an der Umgebungsvariablen wirksam werden.
5. Wechseln Sie unter UNIX in das Verzeichnis `$INSTHOME/sql1lib/cfg` und unter Windows in das Verzeichnis `%DB2PATH%/cfg`.
6. Führen Sie den Befehl **enable_MQFunctions** aus, um eine Datenbank für die DB2 MQSeries-Funktionen zu konfigurieren und zu aktivieren. Eine komplette Beschreibung dieses Befehls finden Sie in „enable_MQFunctions“ auf Seite 204.

Einige allgemeine Beispiele werden weiter unten erläutert. Nach erfolgreicher Durchführung wird die angegebene Datenbank aktiviert und die Konfiguration getestet.

7. Setzen Sie nach Herstellung einer Verbindung zur aktivierten Datenbank zum Testen der Funktionen mit dem Befehlszeilenprozessor die folgenden Befehle ab:

```
values DB2MQ.MQSEND('ein Test')
values DB2MQ.MQRECEIVE()
```

Die erste Anweisung sendet die Nachricht "ein Test" an die Warteschlange DB2MQ_DEFAULT_Q. Mit der zweiten Anweisung wird die Nachricht wieder empfangen.

Anmerkung: Bei Ausführung von **enable_MQFunctions** wird eine MQSeries-Standardumgebung erstellt. Der MQSeries-Warteschlangenmanager DB2MQ_DEFAULT_MQM und die Standard-Warteschlange DB2MQ_DEFAULT_Q werden erstellt. Ferner werden die Dateien amt.xml, amthost.xml und amt.dtd erstellt, sofern sie nicht bereits in dem Verzeichnis enthalten sind, auf das AMT_DATA_PATH verweist. Wenn die Datei amthost.xml vorhanden ist und keine Definition für connectionDB2MQ enthält, wird der Datei eine Zeile mit den entsprechenden Informationen hinzugefügt. Eine Kopie der Originaldatei wird als DB2MQSAVE.amthost.xml gespeichert.

MQSeries - Nachrichtendarstellung

MQSeries unterstützt drei Nachrichtenübertragungsmodelle: Datagramme, Veröffentlichen/Teilnehmen (Publish/Subscribe - P/S) und Anfordern/Antworten (Request/Reply - R/R).

Nachrichten, die als Datagramme gesendet werden, werden an ein einzelnes Ziel gesendet, ohne dass eine Antwort erwartet wird. Beim P/S-Modell sendet mindestens ein Herausgeber eine Nachricht an einen Veröffentlichungsservice, der die Nachricht an mindestens einen interessierten Subskribenten verteilt. Das R/R-Modell ist dem Datagramm-Modell ähnlich, der Sender erwartet jedoch eine Antwort.

Nachrichtenstruktur

MQSeries unterstützt oder erfordert keine bestimmte Struktur der zu übertragenden Nachrichten.

Andere Produkte, wie beispielsweise MQSeries Integrator (MQSI), bieten dagegen Unterstützung für Nachrichten, die als C-, Cobol- oder XML-Zeichenfolgen erstellt wurden. Strukturierte Nachrichten werden in MQSI durch ein Nachrichten-Repository definiert. XML-Nachrichten verfügen normalerweise über eine selbsterklärende Nachrichtenstruktur und können ebenfalls über das Repository verwaltet werden. Nachrichten können auch unstrukturiert sein. In diesem Fall ist zum syntaktischen Analysieren oder Erstellen des Nachrichteninhalts Benutzercode erforderlich. Nachrichten dieser Art sind häufig semistrukturiert, d. h. zum Unterteilen der Felder innerhalb der Nachricht werden Bytepositionen oder feste Begrenzer verwendet. Der Assistent „MQSeries Assist“ bietet Unterstützung für semistrukturierte Nachrichten. XML-Nachrichten werden durch einige der neuen Funktionen von DB2 XML Extender unterstützt.

MQSeries - Funktionsübersicht

DB2 UDB Version 7.2 enthält eine Reihe von MQSeries-Funktionen zum Einfügen von Nachrichtenübertragungsoperationen in SQL-Anweisungen. Diese Unterstützung gilt folglich für alle Anwendungen, die in einer der unterstützten Sprachen (z. B. C, Java oder SQL) unter Verwendung einer beliebigen Datenbankschnittstelle geschrieben wurden. Bei allen nachfolgenden Beispielen wurde SQL verwendet. In diesem Fall kann SQL von anderen Programmiersprachen für standardmäßige Verarbeitungsvorgänge verwendet werden. Alle oben beschriebenen Nachrichtend- arstellungen von MQSeries werden unterstützt. Weitere Informationen zu den MQSeries-Funktionen finden Sie im Abschnitt zum Handbuch „SQL Reference“ in den Release-Informationen.

In einer Basiskonfiguration befinden sich ein MQSeries-Server und DB2 auf dem Datenbankserver. Die MQSeries-Funktionen werden in DB2 installiert und ermöglichen den Zugriff auf den MQSeries-Server. DB2-Clients können auf allen Maschinen installiert sein, auf die der DB2-Server zugreifen kann. Mehrere Clients können über die Datenbank gleichzeitig auf die MQSeries-Funktionen zugreifen. Über die bereitgestellten Funktionen können DB2-Clients innerhalb von SQL-Anweisungen Nachrichtenübertragungsoperationen durchführen. Diese Nachrichtenübertragungsoperationen ermöglichen DB2-Anwendungen die Kommunikation untereinander oder mit anderen MQSeries-Anwendungen.

Der Befehl **enable_MQFunctions** wird zur Aktivierung einer DB2-Datenbank für die MQSeries-Funktionen verwendet. Dabei wird eine einfache Standardkonfiguration erstellt, die von Client-Anwendungen ohne weitere Verwaltungsmaßnahmen verwendet werden kann. Eine Beschreibung hierzu finden Sie in den Abschnitten „enable_MQFunctions“ und „disable_MQFunctions“. Die Standardkonfiguration ermöglicht Anwendungsprogrammierern einen schnellen Einstieg und bietet eine einfachere Entwicklungsschnittstelle. Weitere Funktionen können bei Bedarf inkrementell konfiguriert werden.

Beispiel 1: Die SQL-Anweisung zum Senden einer einfachen Nachricht unter Verwendung der Standardkonfiguration lautet wie folgt:

```
VALUES DB2MQ.MQSEND('einfache nachricht')
```

Die Nachricht einfache nachricht wird an den MQSeries-Warteschlangenmanager und an die von der Standardkonfiguration definierte Warteschlange gesendet.

MQSeries Application Messaging Interface (AMI) nimmt eine klare Trennung zwischen den Nachrichtenübertragungsoperationen und den Definitionen für die Ausführung dieser Operationen vor. Die Definitionen befinden sich in einer externen Repository-Datei und werden vom AMI-Verwaltungs-Tool verwaltet. Aus diesem Grund ist die Entwicklung und Verwaltung von AMI-Anwendungen einfach. Die DB2 MQSeries-Funktionen basieren auf der MQSeries-Schnittstelle AMI. AMI unterstützt zur Speicherung von Konfigurationsdaten eine externe Konfigurationsdatei, die AMI-Repository genannt wird. Die Standardkonfiguration enthält ein MQSeries AMI-Repository, das für DB2 konfiguriert ist.

Die beiden Schlüsselkonzepte von MQSeries AMI, Servicepunkte und Richtlinien, werden in die DB2 MQSeries-Funktionen übertragen. Ein Servicepunkt ist ein logischer Endpunkt, von dem eine Nachricht gesendet oder empfangen werden kann. Im AMI-Repository wird jeder Servicepunkt durch den Namen einer Warteschlange und einen Warteschlangenmanager von MQSeries definiert. Richtlinien legen die Qualität von Serviceoptionen fest, die für einen bestimmten Nachrichtenübertragungsvorgang verwendet werden. Die wichtigsten Qualitäten sind die

Nachrichtenpriorität und der Datenbankzugriff. Dabei werden Standardservicepunkte und Standard-Richtliniendefinitionen zur Verfügung gestellt, die der Entwickler zur weiteren Anwendungsvereinfachung verwenden kann. Beispiel 1 kann umgeschrieben werden, um den Standardservicepunkt und den Richtliniennamen wie folgt explizit anzugeben:

Beispiel 2:

```
VALUES DB2MQ.MQSEND('DB2.DEFAULT.SERVICE', 'DB2.DEFAULT.POLICY', 'einfache Nachricht')
```

Warteschlangen können von mindestens einer Anwendung auf dem Server bedient werden, auf dem sich die Warteschlangen und Anwendungen befinden. In zahlreichen Konfigurationen werden zur Unterstützung unterschiedlicher Anwendungen und Einsatzmöglichkeiten mehrere Warteschlangen definiert. Aus diesem Grund ist es oft wichtig, bei MQSeries-Anforderungen unterschiedliche Servicepunkte zu definieren. Das folgende Beispiel veranschaulicht dies:

Beispiel 3:

```
VALUES DB2MQ.MQSEND('ODS_Eingabe', 'einfache Nachricht')
```

Anmerkung: Da in diesem Beispiel keine Richtlinie angegeben wurde, wird die Standardrichtlinie verwendet.

Einschränkungen

MQSeries bietet die Möglichkeit, Nachrichten- und Datenbankoperationen in einer einzelnen Arbeitseinheit als autarke Transaktion zu kombinieren. Diese Funktion wird von den MQSeries-Funktionen unter Unix und Windows ursprünglich nicht unterstützt.

Bei Verwendung der Sende- oder Empfangsfunktionen ist die maximale Länge von Nachrichten 4000 Zeichen. Das ist auch die maximale Größe einer Nachricht für die Veröffentlichung mit MQPublish.

Fehlercodes

Die Rückkehrcodes von MQSeries-Funktionen finden Sie in Anhang B des Handbuchs zu MQSeries Application Messaging Interface.

Einsatzszenarios

Die MQSeries-Funktionen können in einer Vielzahl von Szenarios verwendet werden. In diesem Abschnitt werden einige der allgemeineren Szenarios, einschließlich Basis-Nachrichtenübertragung, Anwendungskonnektivität und Datenveröffentlichung, beschrieben.

Basis-Nachrichtenübertragung

Die grundlegendste Form der Nachrichtenübertragung mit den DB2 MQSeries-Funktionen findet statt, wenn alle Datenbankanwendungen mit demselben DB2-Server verbunden sind. Die Clients können mit dem Server lokal oder in einer Netzwerkumgebung verbunden sein.

In einem einfachen Szenario ruft Client A die Funktion MQSEND auf, um eine benutzerdefinierte Zeichenfolge an den standardmäßig definierten Servicestandort zu senden. Die MQSeries-Funktionen werden dann in DB2 auf dem Datenbankserver ausgeführt. Zu einem späteren Zeitpunkt ruft Client B die Funktion MQRECEIVE auf, um die Nachricht am Kopfsatz der vom Standardservice definierten

Warteschlange zu entfernen und an den Client zurückzugeben. Die MQSeries-Funktionen für diesen Vorgang werden wieder von DB2 ausgeführt.

Datenbank-Clients können die einfache Nachrichtenübertragung auf unterschiedliche Weise verwenden. Im Folgenden werden einige allgemeine Verwendungsmöglichkeiten der Nachrichtenübertragung erläutert:

- Datenerfassung - Informationen werden in Form von Nachrichten von einer oder mehreren, möglicherweise unterschiedlichen Informationsquellen empfangen. Bei diesen Informationsquellen kann es sich um Handelsanwendungen wie SAP oder intern entwickelte Anwendungen handeln. Die diesbezüglichen Daten können von Warteschlangen empfangen und in Datenbanktabellen zur weiteren Verarbeitung oder Analyse gespeichert werden.
- Verteilung der Auslastung - Verarbeitungsaufträge werden an eine Warteschlange übergeben, die von mehreren Exemplaren derselben Anwendung gemeinsam verwendet wird. Wenn ein Exemplar einen Auftrag übernehmen kann, empfängt es eine Nachricht mit dem entsprechenden Verarbeitungsauftrag vom Anfang der Warteschlange. Auf diese Weise kann die Auslastung, die sich aus einer Warteschlange mit einem Pool von Aufträgen ergibt, auf mehrere Exemplare aufgeteilt werden.
- Senden von Anwendungssignalen - Wenn mehrere Prozesse zusammenarbeiten, werden Nachrichten häufig zur Koordinierung der entsprechenden Verarbeitungsvorgänge verwendet. Nachrichten dieser Art können auszuführende Befehle oder Verarbeitungsaufträge enthalten. Normalerweise werden die Anwendungssignale hier nur in eine Richtung gesendet. Der die Nachricht auslösende Teilnehmer erwartet keine Antwort. Weitere Informationen hierzu finden Sie in „R/R-Kommunikation“ auf Seite 200.
- Senden von Anwendungshinweisen - Das Senden von Anwendungshinweisen ähnelt dem Senden von Anwendungssignalen, da der Initiator die Daten ohne Erwartung einer Antwort sendet. Normalerweise enthalten Anwendungshinweise jedoch Daten zu bereits stattgefundenen Geschäftsereignissen. „P/S“ auf Seite 201 stellt eine erweiterte Form des Sendens von Anwendungshinweisen dar.

Das folgende Szenario ist eine Erweiterung des obigen Szenarios und berücksichtigt die Nachrichtenübertragung von einem fernen Standort. Nachrichten werden hierbei zwischen Maschine A und Maschine B ausgetauscht. Die Vorgehensweise ist wie folgt:

1. Der DB2-Client gibt bei der Ausführung eines MQSEND-Aufrufs einen Ziel-service an, der eine ferne Warteschlange auf Maschine B repräsentiert.
2. Die eigentliche MQSeries-Arbeit zum Senden der Nachricht wird von den DB2 MQSeries-Funktionen ausgeführt. Der MQSeries-Server auf Maschine A akzeptiert die Nachricht und garantiert, dass sie an die vom Servicepunkt und der aktuellen MQSeries-Konfiguration von Maschine A definierte Zieladresse gesendet wird. Der Server stellt fest, dass es sich um eine Warteschlange auf Maschine B handelt. Dann versucht er, die Nachricht an den MQSeries-Server auf Maschine B zu übergeben, wobei der Vorgang bei Bedarf transparent wiederholt wird.
3. Der MQSeries-Server auf Maschine B akzeptiert die Nachricht vom Server auf Maschine A und übergibt sie an die Bestimmungwarteschlange auf Maschine B.
4. Ein MQSeries-Client auf Maschine B fordert die Nachricht am Kopfsatz der Warteschlange an.

Senden von Nachrichten

Unter Verwendung von MQSEND legt ein DB2-Benutzer oder -Entwickler fest, welche Daten zu welchem Zeitpunkt an welche Zieladresse gesendet werden. In der IT-Branche wird dieser Vorgang als "Senden und Vergessen" (Send and Forget) bezeichnet. Damit ist gemeint, dass der Sender die Nachricht nur sendet und sich im Hinblick auf den ordnungsgemäßen Empfang der Nachricht auf die garantierten Übergabeprotokolle von MQSeries verlässt. Diese Vorgehensweise wird in den folgenden Beispielen verdeutlicht.

Beispiel 4: Senden einer benutzerdefinierten Zeichenfolge an den Servicepunkt meinPlatz mit der Richtlinie hohePriorität:

```
VALUES DB2MQ.MQSEND('meinPlatz','hohePriorität','test')
```

In diesem Fall bezieht sich die Richtlinie hohePriorität auf eine Richtlinie, die im AMI-Repository definiert ist, die MQSeries-Priorität auf die höchste Stufe setzt und möglicherweise andere Servicequalitäten, wie beispielsweise den Datenbankzugriff, ebenfalls anpasst. Der Nachrichteninhalt kann aus beliebigen gültigen Kombinationen von SQL- und benutzerdefinierten Daten bestehen. Dazu gehören verschachtelte Funktionen, Operatoren und Umsetzungsausdrücke. Unter Verwendung der Tabelle EMPLOYEE mit den VARCHAR-Spalten LASTNAME, FIRSTNAME und DEPARTMENT gehen Sie zum Senden einer Nachricht, die diese Informationen für jeden Mitarbeiter in Abteilung 5LGA enthält, beispielsweise wie folgt vor:

Beispiel 5:

```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT)
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

Wenn die Tabelle eine ganzzahlige Spalte AGE enthält, kann sie wie folgt eingefügt werden:

Beispiel 6:

```
SELECT DB2MQ.MQSEND(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT || ' '
|| char(AGE))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

Das folgende Beispiel zeigt schließlich, wie Nachrichteninhalt mit einem gültigem SQL-Ausdruck abgeleitet werden kann. Unter Verwendung einer zweiten Tabelle DEPT mit den VARCHAR-Spalten DEPT_NO und DEPT_NAME können Nachrichten gesendet werden, die LASTNAME und DEPT_NAME für den Mitarbeiter enthalten:

Beispiel 7:

```
SELECT DB2MQ.MQSEND(e.LASTNAME || ' ' || d.DEPTNAME) FROM EMPLOYEE e, DEPT d
WHERE e.DEPARTMENT = d.DEPTNAME
```

Nachrichten abrufen

Mit den DB2 MQSeries-Funktionen können Nachrichten empfangen oder gelesen werden. Der Unterschied zwischen Lesen und Empfangen besteht darin, dass die Nachricht beim Lesen an den Kopfsatz der Warteschlange zurückgegeben, also nicht aus der Warteschlange entfernt wird. Bei Empfangsoperationen wird die Nachricht aus der Warteschlange entfernt. Eine Nachricht, die über eine Empfangsoperation abgerufen wird, kann nur einmal abgerufen werden. Eine Nachricht, die

über einen Lesevorgang abgerufen wird, kann dagegen viele Male abgerufen werden. Die folgenden Beispiele veranschaulichen dies:

Beispiel 8:

```
VALUES DB2MQ.MQREAD()
```

Dieses Beispiel gibt eine VARCHAR-Zeichenfolge mit der Nachricht am Kopfsatz der Warteschlange zurück, die über den Standardservice unter Verwendung der Standardrichtlinie für die Servicequalität definiert wurde. Wichtig ist zu beachten, dass ein Nullwert zurückgegeben wird, wenn keine Nachrichten zum Lesen vorhanden sind. Die Warteschlange wird durch diese Operation nicht geändert.

Beispiel 9:

```
VALUES DB2MQ.MQRECEIVE('Employee_Changes')
```

Das obige Beispiel zeigt, wie eine Nachricht aus dem Kopfsatz der Warteschlange entfernt werden kann, die vom Service Employee_Changes unter Verwendung der Standardrichtlinie definiert wurde.

Eine äußerst leistungsstarke Funktion von DB2 besteht darin, eine Tabelle aus einer benutzerdefinierten (oder von DB2 gelieferten) Funktion zu erstellen. Mit dieser Tabellenfunktion können Sie den Inhalt einer Warteschlange als DB2-Tabelle speichern. Das folgende Beispiel zeigt eine vereinfachte Form dieses Vorgangs:

Beispiel 10:

```
SELECT t.* FROM table ( DB2MQ.MQREADALL() ) t
```

Diese Abfrage gibt eine Tabelle zurück, die aus allen Nachrichten der vom Standardservice definierten Warteschlange und den Metadaten zu diesen Nachrichten besteht. Die vollständige Definition der zurückgegebenen Tabellenstruktur ist im Anhang definiert. Die erste Spalte gibt den Inhalt der Nachricht wieder, und die restlichen Spalten enthalten die Metadaten. Wenn nur die Nachrichten zurückgegeben werden sollen, kann das Beispiel wie folgt geändert werden:

Beispiel 11:

```
SELECT t.MSG FROM table (DB2MQ.MQREADALL() ) t
```

Die von einer Tabellenfunktion zurückgegebene Tabelle unterscheidet sich nicht von einer Tabelle, die direkt aus der Datenbank abgerufen wird. Für diese Tabelle gibt es folglich zahlreiche Verwendungsmöglichkeiten. Sie können den Inhalt der Tabelle beispielsweise mit einer anderen Tabelle verbinden oder die Anzahl der Nachrichten in einer Warteschlange zählen:

Beispiel 12:

```
SELECT t.MSG, e.LASTNAME
       FROM table (DB2MQ.MQREADALL() ) t, EMPLOYEE e
       WHERE t.MSG = e.LASTNAME
```

Beispiel 13:

```
SELECT COUNT(*) FROM table (DB2MQ.MQREADALL() ) t
```

Ferner können Sie die Tatsache verdecken, dass die Tabellenquelle eine Warteschlange ist, indem Sie eine Sicht für die Tabellenfunktion erstellen. Das folgende Beispiel erstellt beispielsweise eine Sicht namens NEW_EMP der Warteschlange, auf die von einem Service namens NEW_EMPLOYEES verwiesen wird:

Beispiel 14:

```
CREATE VIEW NEW_EMP (msg) AS
  SELECT t.msg FROM table (DB2MQ.MQREADALL()) t
```

In diesem Fall wird die Sicht nur durch eine einzelne Spalte definiert, die eine vollständige Nachricht enthält. Wenn die Nachrichten einfach strukturiert sind (z. B. bestehend aus zwei Feldern mit einer festen Länge), können die integrierten DB2-Funktionen zur syntaktischen Analyse der beiden Spalten verwendet werden. Wenn Sie beispielsweise wissen, dass Nachrichten an eine bestimmte Warteschlange stets einen Nachnamen aus 18 Zeichen, gefolgt von einem Vornamen aus 18 Zeichen enthalten, können Sie wie folgt eine Sicht definieren, die jedes Feld als separate Spalte enthält:

Beispiel 15:

```
CREATE VIEW NEW_EMP2 AS
  SELECT left(t.msg,18) AS LNAME, right(t.msg,18) AS FNAME
  FROM table(DB2MQ.MQREADALL()) t
```

Der MQSeries-Assistent, eine neue Funktion von DB2 Stored Procedure Builder, kann zur Erstellung neuer DB2-Tabellenfunktionen und DB2-Sichten verwendet werden, die Spalten begrenzte Nachrichtenstrukturen zuordnen.

Schließlich wird häufig gewünscht, den Inhalt mindestens einer Nachricht in der Datenbank zu speichern. Hierzu kann zum Bearbeiten und Speichern des Nachrichteninhalts die volle SQL-Funktionalität genutzt werden. Das folgende Beispiel veranschaulicht dies vermutlich auf einfachste Weise:

Beispiel 16:

```
INSERT INTO MESSAGES
  SELECT t.msg FROM table (DB2MQ.MQRECEIVEALL()) t
```

Unter Verwendung der Tabelle MESSAGES, die eine einzelne VARCHAR(2000)-Spalte enthält, fügt die obige Anweisung die Nachrichten aus der Standard-Servicewarteschlange in die Tabelle ein. Diese Vorgehensweise kann für unterschiedliche Situationen weiter abgewandelt werden.

Anwendungsübergreifende Konnektivität

Die Anwendungsintegration ist ein allgemeiner Bestandteil vieler Lösungen. Ob es um die Integration einer erworbenen Anwendung in eine vorhandene Infrastruktur oder die Integration einer neu entwickelten Anwendung in eine vorhandene Umgebung geht - in vielen Fällen stellt sich die Aufgabe, einen heterogenen Verbund von Subsystemen als funktionierende Einheit zusammenzufügen. MQSeries wird im Allgemeinen als notwendiges Tool für die Anwendungsintegration erachtet. Da MQSeries von den meisten Hardware- und Softwarekomponenten bzw. Sprachumgebungen aufgerufen werden kann, ist eine Verbindung zwischen äußerst heterogenen Anwendungsgruppen möglich.

In diesem Abschnitt werden einige Anwendungsintegrationsszenarios und deren Verwendung mit DB2 beschrieben. Da das Themengebiet sehr groß ist, kann die Anwendungsintegration in diesem Rahmen nicht umfassend behandelt werden. Deshalb liegt der Schwerpunkt auf zwei einfachen Themen: R/R-Kommunikation sowie MQSeries Integrator und P/S.

R/R-Kommunikation

Die R/R-Kommunikation ist eine Methode, die von Anwendungen häufig zur Anforderung von Services anderer Anwendungen verwendet wird. Eine Möglichkeit besteht darin, dem Serviceprovider eine Nachricht mit einer Arbeitsanforderung zu senden. Nachdem die entsprechende Arbeit durchgeführt wurde, kann der Provider die Ergebnisse (oder nur eine Beendigungsbestätigung) an den Anforderer zurücksenden. Bei Verwendung der oben beschriebenen Basis-Nachrichtenübertragungsmethoden ist die Anforderung des Senders jedoch in keiner Weise mit der Antwort des Serviceproviders verbunden. Sofern der Anforderer vor dem Fortfahren nicht auf eine Antwort wartet, muss für die Zuordnung der Antwort zur entsprechenden Anforderung ein bestimmtes Verfahren verwendet werden. Hierzu ist allerdings keine gesonderte Entwicklung notwendig, da MQSeries eine Korrelations-ID zur Verfügung stellt, die einen Bezug zwischen den beteiligten Nachrichten herstellt.

Dieses Verfahren kann auf unterschiedliche Weise eingesetzt werden. Am einfachsten ist es jedoch, wenn der Anforderer die Nachricht wie im folgenden Beispiel gezeigt mit einer bekannten Korrelations-ID kennzeichnet:

Beispiel 17:

```
DB2MQ.MQSEND ('meinRequester','meineRichtlinie','SendeStatus:kund1','Anf1')
```

Durch diese Anweisung wird der Anweisung MQSEND der Parameter Anf1 hinzugefügt, um die Korrelations-ID für die Anforderung anzugeben.

Verwenden Sie zum Empfangen einer Antwort die entsprechende Anweisung MQRECEIVE, um die erste vom angegebenen Service definierte Nachricht selektiv abzurufen, die mit der Korrelations-ID wie folgt übereinstimmt:

Beispiel 18:

```
DB2MQ.MQRECEIVE('meinEmpfänger','meineRichtlinie','Anf1')
```

Wenn die die Anforderung verarbeitende Anwendung ausgelastet ist und der Anforderer vor dem Senden der Antwort die Anweisung MQRECEIVE ausgibt, werden keine Nachrichten gefunden, die mit dieser Korrelations-ID übereinstimmen.

Zum Empfangen von Serviceanforderung und Korrelations-ID wird eine Anweisung wie die folgende verwendet:

Beispiel 19:

```
SELECT msg, correlid FROM  
table (DB2MQ.MQRECEIVEALL('aServiceProvider','meineRichtlinie',1)) t
```

Dadurch wird die Nachricht und die Korrelations-ID der ersten Anforderung vom Service aServiceProvider zurückgegeben.

Sobald der Service ausgeführt wurde, wird eine entsprechende Antwortnachricht an die von aRequester beschriebene Warteschlange gesendet. In der Zwischenzeit können vom Serviceanforderer andere Tätigkeiten ausgeführt werden. Es gibt keine Sicherheit, dass die ursprüngliche Serviceanforderung innerhalb des festgelegten Zeitraums eine Antwort erhält. Zeitlimits auf Anwendungsebene wie diese fallen in den Aufgabenbereich der Entwickler. Der Anforderer muss eine Abfrage durchführen, um das Vorliegen einer Antwort zu prüfen.

Der Vorteil dieser zeitlich unabhängigen asynchronen Verarbeitung liegt darin, dass der Anforderer und der Serviceprovider völlig unabhängig voneinander arbeiten können. Sie kann beispielsweise in Umgebungen eingesetzt werden, in denen Anwendungen nur zeitweilig miteinander verbunden sind, oder in eher stapel-orientierten Umgebungen, in denen mehrere Anforderungen oder Antworten vor der Verarbeitung in Spalten zusammengefasst werden. Diese Art der Spaltenberechnung wird häufig in Data Warehouse-Umgebungen verwendet, um ein Data Warehouse oder einen aktiven Datenspeicher regelmäßig zu aktualisieren.

P/S

Einfache Datenveröffentlichung: Ein weiteres häufiges Szenario der Anwendungsintegration besteht darin, dass eine Anwendung andere Anwendungen über wichtige Ereignisse informiert. Hierzu wird einfach eine Nachricht an eine Warteschlange gesendet, die von einer anderen Anwendung überwacht wird. Der Inhalt der Nachricht kann aus einer benutzerdefinierten Zeichenfolge oder aus Datenbankspalten bestehen. Oft muss nur eine Nachricht unter Verwendung der Funktion MQSEND gesendet werden. Wenn diese Nachricht gleichzeitig an mehrere Empfänger gesendet werden soll, kann die Verteilerfunktion von MQSeries AMI verwendet werden.

Ein Verteiler wird mithilfe des Verwaltungs-Tools von AMI definiert. Ein Verteiler umfasst eine Liste individueller Services. Eine an einen Verteiler gesendete Nachricht wird an jeden innerhalb der Liste definierten Service gesendet. Dies ist vor allem dann besonders nützlich, wenn einige Services immer an allen Nachrichten interessiert sind. Das folgende Beispiel zeigt, wie eine Nachricht an die interessierten Parteien im Verteiler gesendet wird:

Beispiel 20:

```
DB2MQ.MQSEND('interessierteGruppen','allgemein interessante Informationen');
```

Wenn für Nachrichten an bestimmte Services bessere Kontrollmöglichkeiten zur Verfügung stehen sollen, muss die P/S-Funktion verwendet werden. P/S-Systeme bieten normalerweise eine skalierbare und sichere Umgebung, in der sich viele Subskribenten registrieren können, um Nachrichten von mehreren Herausgebern zu empfangen. Zur Unterstützung dieser Funktionalität kann die MQPublish-Schnittstelle in Verbindung mit MQSeries Integrator oder der P/S-Funktion von MQSeries verwendet werden.

MQPublish ermöglicht dem Benutzer, der Nachricht optional ein Thema hinzuzufügen. Mit dem Thema kann der Subskribent die zu akzeptierenden Nachrichten genauer definieren. Zu diesem Zweck müssen die folgenden Schritte ausgeführt werden:

1. Ein MQSeries-Administrator konfiguriert die P/S-Funktionalität von MQSeries Integrator.
2. Interessierte Anwendungen nehmen an Subskriptionspunkten teil, die über die MQSI-Konfiguration definiert werden, und geben optional für sie interessante Themen an. Jeder Subskribent wählt die für ihn relevanten Themen aus und kann ferner auf die inhaltsbasierten Subskriptionsmethoden von MQSeries Integrator V2 zurückgreifen. Wichtig ist zu beachten, dass Warteschlangen, die über einen Servicenamen dargestellt werden, den Subskribenten definieren.

3. Eine DB2-Anwendung veröffentlicht eine Nachricht an den Servicepunkt "Wetter". Die Nachricht gibt als Wetter Schneeregen und als Thema München an. Interessierte Subskribenten werden also darüber informiert, dass es in München Schneeregen gibt.
4. Die tatsächliche Veröffentlichung der Nachricht wird von den DB2 MQSeries-Funktionen übernommen. Die Nachricht wird an MQSeries Integrator unter Verwendung des Servicenamens "Wetter" gesendet.
5. MQSI akzeptiert die Nachricht, führt die von der MQSI-Konfiguration definierten Verarbeitungsschritte durch, und bestimmt die entsprechenden Subskriptionen. Dann wird die Nachricht an die betroffenen Subskribenten-Warteschlangen weitergeleitet.
6. Anwendungen, die am Service "Wetter" teilnehmen und für München registriert sind, empfangen die Nachricht Schneeschauer im jeweiligen Empfangsservice.

Zum Veröffentlichenden dieser Daten unter Verwendung aller Standardeinstellungen und eines Nullthemas wird die folgende Anweisung verwendet:

Beispiel 21:

```
SELECT DB2MQ.MQPUBLISH(LASTNAME || ' ' || FIRSTNAME || ' ' || DEPARTMENT ||
' ' ||char(AGE))
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

Bei vollständiger Angabe aller Parameter und Vereinfachung der Nachricht, so dass nur LASTNAME enthalten ist, lautet die Anweisung wie folgt:

Beispiel 22:

```
SELECT DB2MQ.MQPUBLISH('HR_INFO_PUB', 'SPECIAL_POLICY', LASTNAME,
'ALL_EMP:5LGA', 'MANAGER')
FROM EMPLOYEE
WHERE DEPARTMENT = '5LGA'
```

Diese Anweisung veröffentlicht Nachrichten an den Veröffentlichungsservice HR_INFO_PUB, wobei der Service SPECIAL_POLICY verwendet wird. Die Nachrichten weisen darauf hin, dass der Sender das Thema MANAGER ist. Die Zeichenfolge für das Thema zeigt, dass mehrere Themen angegeben werden können, die mit einem Doppelpunkt (:) miteinander verknüpft werden. In diesem Beispiel können sich die Subskribenten aufgrund der Verwendung von zwei Themen für ALL_EMP oder nur für 5LGA registrieren, um die Nachrichten zu empfangen.

Für den Empfang veröffentlichter Nachrichten muss sich der Benutzer zunächst für Nachrichten mit einem bestimmten Thema registrieren und den Namen des Subskribentenservices angeben, an die die Nachrichten gesendet werden sollen. Wichtig ist zu beachten, dass ein AMI-Subskribentenservice einen Broker- und einen Empfängerservice definiert. Der Broker-Service gibt an, wie der Subskribent mit dem P/S-Broker kommuniziert. Der Empfängerservice ist der Service, an den die mit der Subskription übereinstimmenden Nachrichten gesendet werden. Bei der folgenden Anweisung erfolgt eine Registrierung für das Thema ALL_EMP.

Beispiel 23:

```
DB2MQ.MQSUBSCRIBE('einSubskribent', 'ALL_EMP')
```

Nachdem eine Teilnehmerberechtigung für eine Anwendung eingerichtet wurde, werden Nachrichten mit dem Betreff ALL_EMP an den vom Subskribentenservice definierten Empfängerservice gesendet. Eine Anwendung kann über mehrere

gleichzeitig geltende Subskriptionen verfügen. Um die entsprechenden Nachrichten für eine Subskription zu erhalten, kann eine der Standardfunktionen für den Nachrichtenabruf verwendet werden. Wenn der Subskribentenservice einSubskribent den Empfängerservice beispielsweise als einSubskribentEmpfänger definiert, wird die erste Nachricht von der folgenden Anweisung ohne Löschen gelesen:

Beispiel 24:

```
DB2MQ.MQREAD('einSubskribentEmpfänger')
```

Zur Ermittlung der Nachrichten und der zugehörigen Themen kann eine der Tabellenfunktionen verwendet werden. Die folgende Anweisung empfängt die ersten fünf Nachrichten von einSubskribentEmpfänger und zeigt sowohl Nachricht als auch Thema an:

Beispiel 25:

```
SELECT t.msg, t.topic FROM table (DB2MQ.MQRECEIVEALL('einSubskribentEmpfänger',5)) t
```

Zum Lesen aller Nachrichten mit dem Betreff ALL_EMP kann unter Verwendung von SQL folgende Anweisung ausgegeben werden:

Beispiel 26:

```
SELECT t.msg FROM table (DB2MQ.MQREADALL('einSubskribentEmpfänger')) t
WHERE t.topic = 'ALL_EMP'
```

Anmerkung: Wichtig ist zu beachten, dass bei Verwendung von MQRECEIVEALL mit einer Integritätsbedingung die gesamte Warteschlange verarbeitet wird und nicht nur Nachrichten mit dem Betreff ALL_EMP. Dies liegt daran, dass vor Anwendung der Integritätsbedingung die Tabellenfunktion ausgeführt wird.

Wenn an einem bestimmten Thema kein Interesse mehr besteht, muss die Teilnehmerberechtigung mit einer Anweisung wie folgt explizit aufgehoben werden:

Beispiel 27:

```
DB2MQ.MQUNSUBSCRIBE('einSubskribent', 'ALL_EMP')
```

Nach Ausgabe dieser Anweisung sendet der P/S-Broker keine für diese Subskription relevanten Nachrichten mehr.

Automatisierte Veröffentlichung: Ein weiteres wichtiges Verfahren der Datenbank-Nachrichtenübertragung ist die automatisierte Veröffentlichung. Unter Verwendung des Auslöser-Tools in DB2 können Nachrichten im Rahmen eines Auslöseraufrufs automatisch veröffentlicht werden. Zwar können für die automatisierte Datenveröffentlichung auch andere Verfahren verwendet werden, der auslöserbasierte Ansatz gewährt Administratoren und Entwicklern jedoch große Freiheit bei der Erstellung von Nachrichteninhalt und hohe Flexibilität bei der Definition von Auslöse-Aktionen. Wie bei anderen Verfahren müssen auch bei der Verwendung von Auslösern die Häufigkeit der Ausführung und die diesbezüglichen Kosten berücksichtigt werden. Die folgenden Beispiele veranschaulichen die Verwendung von Auslösern mit den DB2 MQSeries-Funktionen.

Das folgende Beispiel zeigt, wie einfach es ist, bei der Einstellung eines neuen Mitarbeiters eine Nachricht zu veröffentlichen. Alle Benutzer oder Anwendungen, die am Service HR_INFO_PUB teilnehmen und sich für NEW_EMP registrieren, erhalten eine Nachricht mit dem Einstellungsdatum, dem Namen und der Abteilung des neuen Mitarbeiters.

Beispiel 28:

```
CREATE TRIGGER neuer_mitarbeiter AFTER INSERT ON employee REFERENCING NEW AS n
FOR EACH ROW MODE DB2SQL
VALUES DB2MQ.MQPUBLISH('HR_INFO_PUB&', 'NEW_EMP',
current date || ' ' || LASTNAME || ' ' || DEPARTMENT)
```

enable_MQFunctions

enable_MQFunctions

Aktiviert die DB2 MQSeries-Funktionen für die angegebene Datenbank und stellt sicher, dass die Funktionen ordnungsgemäß ausgeführt werden können. Der Befehl schlägt fehl, wenn MQSeries und MQSeries AMI nicht installiert und konfiguriert sind.

Berechtigung

Die folgenden Parameter sind verfügbar:

- *sysadm*
- *dbadm*
- IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder explizite Schemenname der Funktion nicht vorhanden ist.
- Zugriffsrecht CREATEIN auf dem Schema, wenn der Schemenname DB2MQ vorhanden ist.

Befehlssyntax

```
▶▶ enable_MQFunctions --n datenbank --u benutzer-id --p kennwort ▶▶
┌ force ─┘ ┌ noValidate ─┘
```

Befehlsparameter

-n datenbank

Gibt den Namen der Datenbank an, die aktiviert werden soll.

-u benutzer-id

Gibt die Benutzer-ID für die Verbindung mit der Datenbank an.

-p kennwort

Gibt das Kennwort für die Benutzer-ID an.

-force Gibt an, dass Warnungen während einer Reinstallation ignoriert werden sollen.

-noValidate

Gibt an, dass die Gültigkeitsprüfung der DB2 MQSeries-Funktionen nicht ausgeführt wird.

Beispiele

Im folgenden Beispiel werden DB2 MQ-Funktionen erstellt. Der Benutzer stellt eine Verbindung zur Datenbank SAMPLE her. Das Standardschema DB2MQ wird verwendet.

```
enable_MQFunctions -n sample -u benutzer1 -p kennwort1
```

Hinweise

Die DB2 MQ-Funktionen werden unter dem Schema DB2MQ ausgeführt, das durch diesen Befehl automatisch erstellt wird.

Vor Ausführung des Befehls:

- Stellen Sie sicher, dass MQ und AMI installiert sind und MQSeries ab Version 5.2 verwendet wird.
- Stellen Sie sicher, dass die Umgebungsvariable \$AMT_DATA_PATH definiert ist.
- Wechseln Sie in das Unterverzeichnis cfg von DB2PATH.

Unter UNIX:

- Verwenden Sie db2set, um DB2ENVLIST AMT_DATA_PATH hinzuzufügen.
- Stellen Sie sicher, dass das der UDF-Ausführung zugeordnete Benutzerkonto ein Element der mqm-Gruppe ist.
- Stellen Sie sicher, dass der Benutzer, der diesen Befehl aufruft, ein Element der mqm-Gruppe ist.

Anmerkung: AIX 4.2 wird von MQSeries 5.2 nicht unterstützt.

disable_MQFunctions

disable_MQFunctions

Inaktiviert die DB2 MQSeries-Funktionen für die angegebene Datenbank.

Berechtigung

Die folgenden Parameter sind verfügbar:

- *sysadm*
- *dbadm*
- IMPLICIT_SCHEMA für die Datenbank, wenn der implizite oder explizite Schemenname der Funktion nicht vorhanden ist.
- Zugriffsrecht CREATEIN für das Schema, wenn der Schemenname DB2MQ vorhanden ist.

Befehlssyntax

```
►► disable_MQFunctions -n datenbank -u benutzer-id -p kennwort ◀◀
```

Befehlsparameter

-n datenbank

Gibt den Namen der Datenbank an.

-u benutzer-id

Gibt die Benutzer-ID für die Verbindung mit der Datenbank an.

-p kennwort

Gibt das Kennwort für die Benutzer-ID an.

Beispiele

Im folgenden Beispiel werden die DB2 MQ-Funktionen für die Datenbank SAMPLE inaktiviert.

```
disable_MQFunctions -n sample -u benutzer1 -p kennwort1
```

Teil 4. Verwaltungs-Tools

Steuerzentrale	209
Fähigkeit zur Verwaltung von DB2-Servern für VSE und VM	209
Java 1.2-Unterstützung für die Steuerzentrale.	209
Verknüpfungsfehler bei der Verwendung der Onlinehilfefunktion im Windows-Betriebssystem	210
Java-Steuerzentrale unter OS/2	210
Fehler "Dateizugriff verweigert" beim Versuch, einen beendeten Job im Journal im Windows- Betriebssystem anzuzeigen	210
Testverbindung für Aktualisierung auf mehreren Systemen	210
Steuerzentrale für DB2 für OS/390	210
Erforderliche Korrektur für die Steuerzentrale für OS/390	212
Änderung am Dialogfenster "Räumlichen Verweis erstellen"	212
Informationen zur Fehlerbehebung für die DB2- Steuerzentrale	213
Fehlerbehebung der Steuerzentrale auf UNIX- basierten Systemen	213
Mögliches Problem der Kurzhilfe bei OS/2	213
Hilfe für den Konfigurationsparameter "jdk11_path".	213
Solaris-Systemfehler (SQL10012N) bei der Verwen- dung der Prozedurzentrale oder des Journals.	214
Hilfe für die Datei DPREPL.DFT	214
Starten mehrerer Steuerzentralen-Applets	214
Onlinehilfefunktion für die Steuerzentrale bei der Ausführung als Applet	214
Ausführen der Steuerzentrale im Applet-Modus (Windows 95)	215
Arbeiten mit großen Abfrageergebnissen	215
Information - Unterstützung	217
Verknüpfungsfehler im Windows-Betriebssystem	217
Öffnen externer Web-Links in Netscape Navigator, wenn Netscape bereits geöffnet ist (UNIX-basierte Systeme)	217
Probleme beim Starten von "Information - Unter- stützung".	217
Assistenten	219
Einstellen der Speicherbereichsgröße in "Assistent: Datenbank erstellen"	219
Assistent "MQSeries Assist"	219
Assistent "OLE DB Assist"	219

Steuerzentrale

Fähigkeit zur Verwaltung von DB2-Servern für VSE und VM

Die Unterstützung von Datenbanken mit DB2 Server für VSE und VM durch die Steuerzentrale von DB2 Universal Database Version 7 wurde erweitert. Alle Datenbankobjekte von DB2 Server für VSE und VM können von der Steuerzentrale angezeigt werden. Unterstützt werden auch die Anweisungen CREATE INDEX, REORGANIZE INDEX und UPDATE STATISTICS sowie der Befehl REBIND. REORGANIZE INDEX und REBIND erfordern die Ausführung einer gespeicherten Prozedur auf den Hosts mit DB2 Server für VSE und VM. Diese gespeicherte Prozedur wird von der Steuerzentrale für VSE und VM von DB2 Server für VSE und VM bereitgestellt.

Durch die voll integrierte Steuerzentrale kann der Benutzer DB2 unabhängig von der Plattform verwalten, auf der der DB2-Server ausgeführt wird. Objekte von DB2 Server für VSE und VM werden im Hauptfenster der Steuerzentrale zusammen mit Objekten von DB2 Universal Database angezeigt. Die entsprechenden Funktionen und Dienstprogramme zur Verwaltung dieser Objekte werden durch Auswählen des Objekts aufgerufen. Beispielsweise kann ein Benutzer die Indizes einer bestimmten Datenbank auflisten, einen der Indizes auswählen und diesen reorganisieren. Der Benutzer kann auch die Tabellen einer Datenbank auflisten und die Statistik aktualisieren oder eine Tabelle als Replikationsquelle definieren.

Informationen zum Konfigurieren der Steuerzentrale für Verwaltungsaufgaben an Objekten von DB2 Server für VSE und VM finden Sie im *DB2 Connect Benutzerhandbuch* oder im Handbuch *Installation und Konfiguration Ergänzung*.

Java 1.2-Unterstützung für die Steuerzentrale

Die Steuerzentrale unterstützt bidirektionale Sprachen wie Arabisch oder Hebräisch mit der Bidi-Unterstützung von Java 1.2. Diese Unterstützung steht nur für die Windows NT-Plattform zur Verfügung.

Java 1.2 muss installiert sein, damit die Steuerzentrale es erkennen und verwenden kann:

1. JDK 1.2.2 ist auf der DB2 UDB-CD im Verzeichnis DB2\bidi\NT verfügbar. `ibm-inst-n122p-win32-x86.exe` ist das Installationsprogramm, `ibm-jdk-n122p-win32-x86.exe` die Datei für die Verteilung von JDK. Kopieren Sie beide Dateien in ein temporäres Verzeichnis auf Ihrer Festplatte und führen Sie das Installationsprogramm von dort aus.
2. Installieren Sie das Produkt unter `<DB2PATH>\java\Java12`. Dabei ist `<DB2PATH>` der Installationspfad von DB2.
3. Wählen Sie JDK/JRE nicht als System-VM aus, wenn Sie von der JDK/JRE-Installation dazu aufgefordert werden.

Nach der erfolgreichen Installation von Java 1.2 wird beim normalen Starten der Steuerzentrale Java 1.2 verwendet.

Wenn Sie Java 1.2 nicht mehr verwenden wollen, können Sie entweder JDK/JRE im Verzeichnis `<DB2PATH>\java\Java12` deinstallieren oder das Unterverzeichnis `<DB2PATH>\java\Java12` umbenennen.

Anmerkung: Verwechseln Sie bitte nicht <DB2PATH>\java\Java12 mit <DB2PATH>\Java12.<DB2PATH>\Java12 ist Teil der DB2-Installation und enthält die JDBC-Unterstützung für Java 1.2.

Verknüpfungsfehler bei der Verwendung der Onlinehilfefunktion im Windows-Betriebssystem

Bei der Verwendung der Onlinehilfefunktion der Steuerzentrale kann ein Fehler u. b. ein Verknüpfungsfehler auftreten. Wenn Sie vor kurzem einen neuen Web-Browser oder eine neue Version eines Web-Browsers installiert haben, stellen Sie sicher, dass HTML- und HTM-Dokumente dem korrekten Browser zugeordnet sind. Weitere Informationen finden Sie im Abschnitt der Windows-Hilfe zum Ändern des Programms, das beim Öffnen einer Datei gestartet wird.

Java-Steuerzentrale unter OS/2

Die Steuerzentrale muss auf einem HPFS-Laufwerk installiert werden.

Fehler "Dateizugriff verweigert" beim Versuch, einen beendeten Job im Journal im Windows-Betriebssystem anzuzeigen

Bei DB2 Universal Database für Windows NT tritt der Fehler "Dateizugriff verweigert" auf, wenn versucht wird, das Journal zu öffnen, um die Details eines Jobs anzuzeigen, der in der Prozedurzentrale erstellt wurde. Als Jobstatus wird „Beendet“ angezeigt. Dieses Verhalten tritt auf, wenn ein Job, der in der Prozedurzentrale erstellt wurde, den Befehl START enthält. Verwenden Sie START/WAIT anstelle von START in der Stapeldatei und im Job, um dieses Verhalten zu vermeiden.

Testverbindung für Aktualisierung auf mehreren Systemen

Die Funktionalität für eine Testverbindung für eine Aktualisierung auf mehreren Systemen in der Steuerzentrale der Version 7 wird von der Version des Zielexemplars eingeschränkt. Das Zielexemplar muss mindestens Version 7 sein, damit die "ferne" Testverbindungsfunktionalität ausgeführt werden kann. Um diese Funktionalität in Version 6 auszuführen, müssen Sie die Steuerzentrale lokal im Zielexemplar laden und von dort ausführen.

Steuerzentrale für DB2 für OS/390

Mit der Steuerzentrale von DB2 UDB für OS/390 können Sie die Verwendung Ihrer lizenzierten IBM DB2-Dienstprogramme verwalten. Dienstprogrammfunktionen, die Teil separat zu erwerbender Funktionen von DB2 UDB für OS/390 sind, müssen lizenziert und in Ihrer Umgebung installiert sein, bevor sie von der DB2-Steuerzentrale verwaltet werden können.

Die Datenbank "CC390", die bei der Konfiguration eines DB2 für OS/390-Subsystems zusammen mit der Steuerzentrale definiert wird, wird für die interne Unterstützung der Steuerzentrale verwendet. Ändern Sie diese Datenbank nicht.

Obwohl DB2 für OS/390 Version 7.1 nicht ausdrücklich im Inhaltsverzeichnis der Steuerzentrale oder in den Informationen von **Information - Unterstützung** erwähnt wird, unterstützt die Dokumentation die Funktionen von DB2 für OS/390 Version 7.1. Viele der spezifischen Funktionen von DB2 für OS/390 Version 6 beziehen sich auch auf DB2 für OS/390 Version 7.1 und einige Funktionen, die laut

Inhaltsverzeichnis spezifisch für DB2 für OS/390 Version 7.1 sind, haben keine Versionsbezeichnung. Wenn Sie ein Subsystem von DB2 für OS/390 Version 7.1 auf Ihrer Steuerzentrale konfiguriert haben, haben Sie Zugriff auf die gesamte Dokumentation für diese Version.

Für den Zugriff und die Verwendung der Funktion **DDL generieren** von der Steuerzentrale von DB2 für OS/390 muss die Funktion **DDL generieren** installiert sein:

- Für Version 5 installieren Sie DB2Admin 2.0 mit DB2 für OS/390 Version 5.
- Für Version 6 installieren Sie die kleine Programmierungserweiterung, die als vorläufige Programmkorrektur für die DB2Admin-Funktion von DB2 für OS/390 Version 6 verfügbar ist.
- Bei Version 7.1 ist die Funktion **DDL generieren** Teil der separat zu erwerbenden DB2Admin-Funktion von DB2 für OS/390 Version 7.1.

Sie können von der Steuerzentrale auf Stored Procedure Builder zugreifen. Dazu muss dieses Programm jedoch beim Starten der DB2 UDB-Steuerzentrale bereits installiert sein. Es ist Teil von DB2 Application Development Client.

Wenn Sie ein Subsystem von DB2 für OS/390 direkt auf der Workstation katalogisieren wollen, geben Sie an, dass **Client-Konfiguration - Unterstützung** verwendet werden soll.

1. Aktivieren Sie auf der Seite **Quelle** den Radioknopf **Verbindung zu einer Datenbank manuell konfigurieren**.
2. Geben Sie auf der Seite **Protokoll** die entsprechenden Kommunikationsinformationen ein.
3. Geben Sie auf der Seite **Datenbank** den Subsystemnamen in das Feld **Datenbankname** ein.
4. Wählen Sie auf der Seite **Knotenoptionen** das optionale Markierungsfeld **Knotenoptionen konfigurieren** aus.
5. Wählen Sie **MVS/ESA, OS/390** aus der Liste im Feld **Betriebssystem** aus.
6. Klicken Sie **Beenden** an, um die Konfiguration abzuschließen.

Wenn Sie ein Subsystem von DB2 für OS/390 über eine Gateway-Maschine katalogisieren wollen, führen Sie Schritt 1-6 oben auf der Gateway-Maschine und anschließend die folgenden Schritte aus:

1. Starten Sie die Steuerzentrale auf der Client-Maschine.
2. Klicken Sie den Systemordner mit Maustaste 2 an, und wählen Sie **Hinzufügen** aus.
3. Geben Sie im Dialogfenster **System hinzufügen** den Namen der Gateway-Maschine in das Feld **Systemname** ein.
4. Geben Sie DB2DAS00 in das Feld für das ferne Exemplar ein.
5. Geben Sie für das TCP/IP-Protokoll in den Protokollparametern den Host-Namen der Gateway-Maschine in das Feld **Host-Name** ein.
6. Geben Sie 523 in das Feld für den Servicenamen ein.
7. Klicken Sie **OK** an, um das System hinzuzufügen. Die Gateway-Maschine sollte jetzt im Ordner **Systeme** enthalten sein.
8. Erweitern Sie den Namen der Gateway-Maschine.
9. Klicken Sie den Exemplarordner mit Maustaste 2 an, und wählen Sie **Hinzufügen** aus.

10. Klicken Sie **Aktualisieren** im Dialogfenster **Exemplar hinzufügen** an, um die auf der Gateway-Maschine verfügbaren Exemplare anzuzeigen. Wenn die Gateway-Maschine ein Windows NT-System ist, wurde das DB2 für OS/390-Subsystem möglicherweise unter dem Exemplar **DB2** katalogisiert.
11. Wählen Sie das Exemplar aus. Die Protokollparameter werden für dieses Exemplar automatisch eingegeben.
12. Klicken Sie **OK** an, um das Exemplar hinzuzufügen.
13. Öffnen Sie den Ordner **Exemplare**, um das gerade hinzugefügte Exemplar anzuzeigen.
14. Erweitern Sie das Exemplar.
15. Klicken Sie den Datenbankordner mit Maustaste 2 an, und wählen Sie **Hinzufügen** aus.
16. Klicken Sie **Aktualisieren** an, um die lokalen Datenbanken auf der Gateway-Maschine anzuzeigen. Wenn Sie ein DB2-Subsystem im Dialogfenster **Datenbank hinzufügen** hinzufügen, geben Sie den Subsystemnamen in das Feld **Datenbankname** ein. Optional: Geben Sie einen lokalen Aliasnamen für das Subsystem (oder die Datenbank) ein).
17. Klicken Sie **OK** an.

Sie haben das Subsystem erfolgreich in der Steuerzentrale hinzugefügt. Wenn Sie die Datenbank öffnen, sollte das DB2 für OS/390-Subsystem angezeigt werden.

Der erste Absatz im Abschnitt zur Steuerzentrale für OS/390 lautet:

Mit der Steuerzentrale von DB2 UDB für OS/390 können Sie die Verwendung Ihrer lizenzierten IBM DB2-Dienstprogramme verwalten. Dienstprogrammfunktionen, die Teil separat zu erwerbender Funktionen von DB2 UDB für OS/390 sind, müssen lizenziert und in Ihrer Umgebung installiert sein, bevor sie von der DB2-Steuerzentrale verwaltet werden können.

Der Abschnitt sollte nun wie folgt lauten:

Mit der Steuerzentrale für OS/390 können Sie die Verwendung Ihrer lizenzierten IBM DB2-Dienstprogramme verwalten. Dienstprogrammfunktionen, die Teil separat zu erwerbender Funktionen sind, müssen lizenziert und in Ihrer Umgebung installiert sein, bevor sie von der DB2-Steuerzentrale verwaltet werden können.

Erforderliche Korrektur für die Steuerzentrale für OS/390

Sie müssen APAR PQ36382 auf 390 Enablement Feature von DB2 für OS/390 Version 5 und DB2 für OS/390 Version 6 anwenden, um diese Subsysteme mit der DB2 UDB-Steuerzentrale für Version 7 verwalten zu können. Ohne diese Korrektur können Sie die DB2 UDB-Steuerzentrale für Version 7 nicht für die Ausführung von Dienstprogrammen für diese Subsysteme verwenden.

Die APAR-Korrektur sollte auf die folgenden Funktionsstatus-IDs (FMID) angewendet werden:

DB2 für OS/390 Version 5 390 Enablement: FMID JDB551D
 DB2 für OS/390 Version 6 390 Enablement: FMID JDB661D

Änderung am Dialogfenster "Räumlichen Verweis erstellen"

Die Druckknöpfe "<<" und ">>" wurden aus dem Dialogfenster **Räumlichen Verweis erstellen** entfernt.

Informationen zur Fehlerbehebung für die DB2-Steuerzentrale

Im Kapitel *Installation und Konfiguration der Steuerzentrale* Ihres Handbuchs *Einstieg* werden Sie im Abschnitt *Informationen zur Fehlerbehebung* aufgefordert, in einem Befehlsfenster die Einstellung der Variablen CLASSPATH Ihres Client-Browsers aufzuheben, wenn beim Ausführen der Steuerzentrale als Applet Probleme auftreten. In diesem Abschnitt werden Sie auch aufgefordert, Ihren Browser aus demselben Befehlsfenster zu starten. Der Befehl zum Starten Ihres Browsers ist jedoch nicht angegeben. Geben Sie `start iexplore` ein, und drücken Sie die Eingabetaste, um den Internet Explorer zu starten. Wenn Sie Netscape starten wollen, geben Sie `start netscape` ein, und drücken Sie die Eingabetaste. Bei diesen Befehlen wird davon ausgegangen, dass der Pfad Ihres Browsers in PATH definiert ist. Ist dies nicht der Fall, fügen Sie eine PATH-Definition für Ihren Browser-Pfad hinzu und wechseln Sie in das Installationsverzeichnis Ihres Browsers, und setzen Sie den Startbefehl erneut ab.

Fehlerbehebung der Steuerzentrale auf UNIX-basierten Systemen

Wenn Sie die Steuerzentrale auf einem UNIX-basierten System nicht starten können, stellen Sie die Umgebungsvariable JAVA_HOME so ein, dass sie auf Ihre Java-Verteilung zeigt:

- Wenn Java unter `/usr/jdk118` installiert ist, setzen Sie JAVA_HOME auf `/usr/jdk118`.
- Für die sh-, ksh- oder bash-Shell:

```
export JAVA_HOME=/usr/jdk118.
```
- Für die csh- oder tcsh-Shell:

```
setenv JAVA_HOME /usr/jdk118
```

Mögliches Problem der Kurzhilfe bei OS/2

Wenn Sie die Steuerzentrale unter OS/2 mit einer Bildschirmauflösung von 1024x768 und 256 Farben verwenden und **Palette für Arbeitsoberfläche optimieren** aktiviert ist, werden Kurzhilfen, die über den Rand des aktuellen Fensters hinausragen, möglicherweise mit schwarzem Text auf schwarzem Hintergrund angezeigt. Zur Behebung dieses Problems ändern Sie entweder die Anzeigeneinstellung auf mehr als 256 Farben oder inaktivieren Sie **Palette für Arbeitsoberfläche optimieren**.

Hilfe für den Konfigurationsparameter "jdk11_path"

In der Hilfe der Steuerzentrale fehlt in der Beschreibung des Konfigurationsparameters für den Installationspfad von Java Development Kit 1.1 (`jdk11_path`) eine Zeile unter der Unterüberschrift **Gilt für**. Die vollständige Liste unter **Gilt für** lautet:

- Datenbank-Server mit lokalen und fernen Clients
- Client
- Datenbank-Server mit lokalen Clients
- Partitionierter Datenbank-Server mit lokalen und fernen Clients
- Satellitendatenbank-Server mit lokalen Clients

Solaris-Systemfehler (SQL10012N) bei der Verwendung der Prozedurzentrale oder des Journals

Wenn ein Solaris-System über die Prozedurzentrale oder das Journal ausgewählt wird, kann der folgende Fehler auftreten:

```
SQL10012N - Während des Ladens der angegebenen Bibliothek
"/udbprod/db2as/sqllib/function/unfenced/db2scdar!ScheduleInfoOpenScan"
trat ein unerwarteter Betriebssystemfehler auf. SQLSTATE=42724.
```

Dies wird durch einen Programmfehler im Solaris-Laufzeit-Verbindungseditor verursacht. Wenden Sie die folgende Programmkorrektur an, um das Problem zu beheben:

105490-06 (107733 ersetzt 105490) für Solaris 2.6

Hilfe für die Datei DPREPL.DFT

In der Hilfe der Steuerzentrale für die Seite **Replikation** des Notizbuchs **Tool - Einstellungen** heißt es in Schritt 5d:

Speichern Sie die Datei im Arbeitsverzeichnis für die Steuerzentrale (z. B. SQLLIB\BIN), so dass das System sie als Standarddatei verwenden kann.

Schritt 5d sollte jedoch folgendermaßen lauten:

Speichern Sie die Datei im Arbeitsverzeichnis für die Steuerzentrale (SQLLIB\CC), so dass das System sie als Standarddatei verwenden kann.

Starten mehrerer Steuerzentralen-Applets

Sie können auf einer Maschine jeweils nur ein Steuerzentralen-Applet starten. Diese Einschränkung gilt für Steuerzentralen-Applets, die in allen unterstützten Browsern ausgeführt werden.

Onlinehilfefunktion für die Steuerzentrale bei der Ausführung als Applet

Wenn die Steuerzentrale als Applet ausgeführt wird, funktioniert die Taste **F1** nur in Fenstern und Notizbüchern, die Kurzhilfe enthalten.

Sie können die Kurzhilfe in den folgenden Komponenten mit der Taste **F1** aufrufen:

- DB2 Universal Database für OS/390
- Assistenten

Bei den übrigen Komponenten der Steuerzentrale wird über die Taste **F1** keine Hilfe aufgerufen. Wenn Sie Hilfe für die anderen Komponenten anzeigen wollen, verwenden Sie den Druckknopf oder das Menü **Hilfe**.

Ausführen der Steuerzentrale im Applet-Modus (Windows 95)

Ein Versuch, die Prozedurzentrale zu öffnen, kann fehlschlagen, wenn eine ungültige Benutzer-ID oder ein ungültiges Kennwort angegeben werden. Stellen Sie sicher, dass Sie beim Anmelden an der Steuerzentrale eine gültige Benutzer-ID und ein gültiges Kennwort angeben.

Arbeiten mit großen Abfrageergebnissen

Es ist einfach für einen Benutzer, eine Abfrage zu erstellen, die eine große Zahl an Zeilen zurückgibt. Vorauszusagen, wieviele Zeilen tatsächlich zurückgegeben werden könnten, ist für einen Benutzer nicht so einfach. Eine Abfrage, die möglicherweise Tausende (oder Millionen) Zeilen zurückgeben könnte, verursacht die folgenden zwei Probleme:

1. Es kann lange dauern, bis das Ergebnis abgerufen wird.
2. Eine große Menge an Client-Speicherkapazität kann erforderlich sein, um das Ergebnis aufzubewahren.

DB2 unterteilt große Ergebnismengen in Blöcke, um diesen Prozess zu ermöglichen. Die Ergebnisse einer Abfrage werden blockweise abgerufen und angezeigt.

Das heißt Folgendes:

1. Die Anzeigezeit wird reduziert, weil der erste Block einer Abfrage zum Anzeigen verfügbar ist, während die restlichen Blöcke abgerufen werden.
2. Der Speicherbedarf auf dem Client wird verkleinert, weil zu jeder Zeit nur ein Block eines Abfrageergebnisses auf dem Client gespeichert ist.

Gehen Sie wie folgt vor, um die Anzahl an Abfrageergebniszeilen im Speicher zu steuern:

1. Öffnen Sie die Seite „Allgemein“ im Notizbuch „Tools - Einstellungen“.
2. Wählen Sie im Abschnitt **Maximalgröße** Folgendes aus:
 - **Beispielinhalt**, um die Anzahl der Ergebniszeilen zu begrenzen, die im Fenster „Beispielinhalt“ angezeigt werden. Geben Sie die Blockgröße der Ergebnismenge (Anzahl Zeilen) in dem Eingabefeld an.
 - **Befehlszentrale**, um die Anzahl der Ergebniszeilen zu begrenzen, die auf der Seite „Abfrageergebnisse“ in der Befehlszentrale angezeigt werden. Geben Sie die Blockgröße der Ergebnismenge (Anzahl Zeilen) in dem Eingabefeld an.

Wenn Sie mit den Ergebnissen einer Abfrage im Fenster „Beispielinhalt“ oder auf der Seite „Abfrageergebnisse“ in der Befehlszentrale arbeiten, gibt das Feld **Zeilen im Speicher** die Anzahl Zeilen an, die für die Abfrage im Speicher stehen. Diese Zahl ist niemals größer als die festgelegte **Maximalgröße**. Klicken Sie **Weiter** an, um den nächsten Block der Ergebnismenge abzurufen. Wenn **Weiter** inaktiv ist, haben Sie das Ende der Ergebnismenge erreicht.

Information - Unterstützung

Verknüpfungsfehler im Windows-Betriebssystem

Bei der Verwendung von **Information - Unterstützung** kann ein Verknüpfungsfehler auftreten. Wenn Sie vor kurzem einen neuen Web-Browser oder eine neue Version eines Web-Browsers installiert haben, stellen Sie sicher, dass HTML- und HTM-Dokumente dem korrekten Browser zugeordnet sind. Weitere Informationen finden Sie im Abschnitt der Windows-Hilfe zum Ändern des Programms, das beim Öffnen einer Datei gestartet wird.

Öffnen externer Web-Links in Netscape Navigator, wenn Netscape bereits geöffnet ist (UNIX-basierte Systeme)

Wenn Netscape Navigator bereits geöffnet ist und entweder ein lokales DB2-HTML-Dokument oder eine externe Website anzeigt, führt der Versuch, eine externe Website von **Information - Unterstützung** aus zu öffnen, zu einem Netscape-Fehler. Der Fehler gibt an, dass Netscape die Datei oder das Verzeichnis mit dem Namen der externen Site nicht finden kann.

Schließen Sie das offene Netscape-Browser-Fenster, *bevor* Sie die externe Website öffnen, um dieses Problem zu umgehen. Netscape wird neu gestartet und zeigt die externe Website an.

Dieser Fehler tritt nicht auf, wenn Sie versuchen, ein lokales DB2-HTML-Dokument zu öffnen, wenn Netscape bereits geöffnet ist.

Probleme beim Starten von "Information - Unterstützung"

Auf einigen Systemen kann es vorkommen, dass **Information - Unterstützung** nur langsam startet, wenn Sie es über **Erste Schritte** im Menü **Start** oder mit dem Befehl **db2ic** aufrufen. Wenn dieses Problem auftritt, starten Sie die Steuerzentrale und wählen Sie Hilfe —> Information - Unterstützung aus.

Assistenten

Einstellen der Speicherbereichsgröße in "Assistent: Datenbank erstellen"

Mit **Assistent: Datenbank erstellen** können die Parameter für die Speicherbereichsgröße (EXTENTSIZE) und die Vorabesezugriffsgröße (PREFETCHSIZE) für den Benutzertabellenbereich (nicht jedoch für die Katalogtabellenbereiche oder temporären Tabellenbereiche) der neuen Datenbank eingestellt werden. Diese Funktion ist nur aktiviert, wenn mindestens ein Behälter für den Benutzertabellenbereich auf der Seite **Benutzertabellen** des Assistenten angegeben ist.

Assistent "MQSeries Assist"

DB2 Version 7.2 enthält den neuen Assistenten „MQSeries Assist“. Dieser Assistent erstellt eine Tabellenfunktion, die aus einer MQSeries-Warteschlange liest, indem sie die DB2-MQSeries-Funktionen verwendet, die in Version 7.2 ebenfalls neu sind. Der Assistent kann jede MQSeries-Nachricht als begrenzte Zeichenfolge behandeln oder als Spaltenzeichenfolge mit fester Länge, abhängig von Ihren Spezifikationen. Die erstellte Tabellenfunktion analysiert die Zeichenfolge syntaktisch und gibt jede MQSeries-Nachricht als Zeile der Tabellenfunktion zurück. Der Assistent ermöglicht es Ihnen auch, eine Sicht der Tabellenfunktion zu erstellen und eine MQSeries-Nachricht und das Tabellenfunktionsergebnis voranzuzeigen. Dieser Assistent kann über Stored Procedure Builder oder über die Data Warehouse-Zentrale gestartet werden.

Voraussetzungen für diesen Assistenten:

- MQSeries Version 5.2
- MQSeries Application Messaging Interface (AMI)
- DB2-MQSeries-Funktionen

Einzelangaben zu diesen Voraussetzungen finden Sie in „MQSeries“ auf Seite 191.

Wenn Sie Beispiele und Lernprogramme für den Assistenten „MQSeries Assist“ benötigen, gehen Sie in den Abschnitt mit den Lernprogrammen auf der Website <http://www.ibm.com/software/data/db2/udb/ide>

Assistent "OLE DB Assist"

Dieser Assistent hilft Ihnen dabei, eine Tabellenfunktion zu erstellen, die Daten von einem anderen Datenbankanbieter liest, der den OLE DB-Standard von Microsoft unterstützt. Sie können optional eine DB2-Tabelle mit den Daten erstellen, die von der OLE DB-Tabellenfunktion gelesen werden, und Sie können eine Sicht für die OLE DB-Tabellenfunktion erstellen. Dieser Assistent kann über Stored Procedure Builder oder über die Data Warehouse-Zentrale gestartet werden.

Voraussetzungen für diesen Assistenten:

- Ein OLE DB-Anbieter (wie z. B. Oracle oder Microsoft SQL Server)
- OLE DB-Unterstützungsfunktionen

Wenn Sie Beispiele und Lernprogramme für den Assistenten „OLE DB Assist“ benötigen, gehen Sie in den Abschnitt mit den Lernprogrammen auf der Website

<http://www.ibm.com/software/data/db2/udb/ide>

Teil 5. Informationsmanagement

Lernprogramm für das Informationsmanagement	225
Überarbeitetes Lernprogramm für das Informationsmanagement	225
Data Warehouse-Zentrale Verwaltung	227
Fehlerbehebung	227
Definieren von Excel als Warehouse-Quelle	227
Definieren und Ausführen von Prozessen	227
Dialog "Metadaten exportieren"	227
Definieren von Werten für das Programm "OS/390-JCL-Jobstrom übergeben (VWPMVS)"	227
Änderungen des Anhangs mit der Data Warehouse-Beispieldatenbank	228
Nachrichten der Data Warehouse-Zentrale.	228
Erstellen einer Modellstruktur und Laden von Daten in den DB2 OLAP Integration Server	228
Verwenden von Classic Connect mit der Data Warehouse-Zentrale	228
Umgebungsdatenstruktur der Data Warehouse-Zentrale	229
Verwenden des Umsetzungsprogramms "Daten vertauschen"	229
Zugreifen auf Daten von DB2 Version 5 mit dem Warehouse-Agenten von DB2 Version 7.	229
Migrieren von Servern von DB2 Version 5.	229
Ändern der Agentenkonfiguration	230
UNIX-Warehouse-Agenten	230
Warehouse-Agenten unter Microsoft Windows NT, Windows 2000 und OS/2	230
Das Programm "ERwin MetaData Extract".	231
Inhalt	231
Softwarevoraussetzungen	232
Programmdateien	232
Erstellen von Befehlssprachendatei	233
Importieren einer Befehlssprachendatei in die Data Warehouse-Zentrale	234
Importieren einer Befehlssprachendatei in Information Catalog Manager	235
Fehlerbehebung	236
Zuordnung von ERwin zur DB2 Data Warehouse-Zentrale	238
Zuordnung von ERwin zu Information Catalog Manager.	239
Namens- und Adressbereinigung in der Data Warehouse-Zentrale	241
Voraussetzungen	241
Komponenten von Trillium Software System	242
Verwenden von Trillium Batch System mit der Data Warehouse-Zentrale	242
Importieren von Trillium-Metadaten.	243
Zuordnen der Metadaten	245
Einschränkungen	245
Schreiben einer Trillium Batch System-JCL-Datei	246
Schreiben einer Trillium Batch System-Script-Datei unter UNIX und Windows	247
Definieren eines Trillium Batch System-Schritts	247
Verwenden des benutzerdefinierten Programms von Trillium Batch System	247
Fehlerbehandlung	249
Fehlercode	250
Protokolldatei	251
Integration von MQSeries in die Data Warehouse-Zentrale	251
Erstellen von Sichten für MQSeries-Nachrichten	251
Voraussetzungen	251
Einschränkungen	252
Erstellen einer Sicht für MQSeries-Nachrichten	252
Importieren von MQSeries-Nachrichten und XML-Metadaten	252
Voraussetzungen	252
Einschränkungen	252
Importieren von MQSeries-Nachrichten und XML-Metadaten	252
Verwenden von benutzerdefinierten Programmen für MQSeries	254
Fehlercode	254
Fehlerprotokolldatei	255
Unterstützung von Microsoft OLE DB und Data Transaction Services	255
Erstellen von Sichten für OLE DB-Tabellenfunktionen	256
Erstellen von Sichten für DTS-Pakete	256
Verwenden von inkrementellen COMMIT-Operationen mit Ersetzen	257
Komponenten-Trace-Datendateinamen	257
Open Client für Sybase-Quellen unter AIX und der Solaris-Betriebsumgebung erforderlich	257
Korrigierte Beispieleinträge.	257
Kapitel 3. Definieren von Warehouse-Quellen.	258
Zuordnen des Memofields in Microsoft Access zu einer Warehouse-Quelle	258
Kapitel 10. Verwalten der Warehouse-Datenbank	258
Verknüpfen von Tabellen mit einem Schrittsubtyp für das Programm DB2 UDB RUNSTATS.	258
Standard-Warehouse-Steuerungsdatenbank	258
Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung"	258
Ändern der aktiven Warehouse-Steuerungsdatenbank.	259
Erstellen und Initialisieren einer Warehouse-Steuerungsdatenbank.	259
Erstellen von Momentaufnahmen für SQL-Schritte	260
Ändern von Quellen und Zielen im Fenster "Prozessmodell"	260
Hinzufügen von Beschreibungen zu Objekten der Data Warehouse-Zentrale	260
Ausführen von Beispieldaten	260
Bearbeiten einer SQL-Anweisung zur DDL-Erstellung	261

Migrieren von Geschäftssichten von Visual Warehouse	261
Generieren von Zieltabellen und Primärschlüsseln	261
Verwenden von Merant ODBC-Treibern	262
Neuer ODBC-Treiber	262
Definieren einer Warehouse-Quelle oder eines Warehouse-Ziels in einer OS/2-Datenbank.	262
Überwachen des Status der Warehouse-Steuerungsdatenbank.	262
Verwenden von SQL Assist mit der Beispieldatenbank TBC_MD	262
Verwenden der Funktion FormatDate	262
Ändern der Spracheinstellung	263
Verwenden des Umsetzungsprogramms "Schlüssel-tabelle generieren".	263
Aufrechterhalten der Verbindungen zu Datenbanken.	263
Definieren eines fernen Clients der Data Warehouse-Zentrale	263
Definieren einer Warehouse-Quelle von DB2 für VM.	264
Definieren einer Zieltabelle von DB2 für VM oder DB2 für VSE.	264
Aktivieren der Unterstützung für begrenzte Bezeichner	264
DataJoiner-Fehler weist auf ein Bindeproblem hin	264
Einrichten und Ausführen von Replikation mit der Data Warehouse-Zentrale	265
Tipps zur Fehlerbehebung	268
Zugreifen auf Quellen und Ziele	268
Ergänzungen zu unterstützten Datenbankquellen von Fremdherstellern.	270
Manuelles Erstellen einer Datenquelle in der Data Warehouse-Zentrale	272
Importieren und Exportieren von Metadaten mit CWMI (Common Warehouse Metadata Interchange)	272
Einführung	272
Importieren von Metadaten.	273
Aktualisieren der Metadaten nach dem Ausführen des Importdienstprogramms	275
Exportieren von Metadaten.	275
Schritt des OS/390-Dienstprogramms Runstats	277
Schritt des OS/390-Dienstprogramms Load	277
XML-Unterstützung für Common Warehouse Meta-model (CWM)	278
Prozesserstellungsprogramm	278
Schemamodellierungsprogramm	279
Musseingabefelder	279
Erweiterungen des Assistenten für die Data Warehouse-Zentrale	279
Ausgeben von Schrittinformationen in eine Datei	279
Data Warehouse Center Application Integration Guide	281
Additional metadata templates	281
Commit.tag	281
Tokens	282
Examples of values	282
ForeignKey.tag	282
Tokens	282

Examples of values	284
ForeignKeyAdditional.tag	284
Tokens	284
Examples of values	286
PrimaryKey.tag	286
Tokens	286
Examples of values	287
PrimaryKeyAdditional.tag	287
Tokens	288
Examples of values	289
Data Warehouse-Zentrale - Onlinehilfefunktion	291
Definieren von Tabellen oder Sichten zur Replikation.	291
Ausführen von Essbase VWP's mit dem AS/400-Agenten	291
Verwenden der Fenster "Veröffentlichen von Metadaten der Data Warehouse-Zentrale" und dem zugehörigen Merkmalfenster	291
Fremdschlüssel	291
Replikationsnotizbücher	291
Importieren einer Befehlssprachendatei	291
Links zum Hinzufügen von Daten	291
Importieren von Tabellen	292
Korrektur der Onlinehilfefunktion zu RUNSTATS und REORGANIZE TABLE.	292
Seite "Hinweis" (Anzeige der Warehouse-Merkmale und Notizbuch "Zeitplan")	292
Das Agentenmodulfeld im Notizbuch "Agenten-Site"	292
DB2 OLAP Starter Kit	293
OLAP Server-Website.	293
Unterstützte Betriebssystemservicestufen	293
Abschließen der Installation von DB2 OLAP Starter Kit unter UNIX.	293
Konfigurieren von ODBC für OLAP Starter Kit	294
Konfigurieren von Datenquellen auf UNIX-Systemen.	294
Konfigurieren der ODBC-Umgebungsvariablen.	295
Editieren der Datei odbcc.ini	295
Hinzufügen einer Datenquelle zu der Datei odbcc.ini	295
Beispiel für ODBC-Einstellungen für DB2	295
Beispiel für ODBC-Einstellungen für Oracle	296
Konfigurieren des OLAP-Metadatenkatalogs auf UNIX-Systemen	296
Konfigurieren von Datenquellen auf Windows-Systemen.	296
Konfigurieren des OLAP-Metadatenkatalog auf Windows-Systemen	297
Vorgehensweise nach dem Konfigurieren einer Datenquelle	299
Anmelden über OLAP Starter Kit Desktop	299
Beispiel für Starter Kit-Anmeldung	300
Manuelles Erstellen und Konfigurieren der Beispieldatenbanken für OLAP Starter Kit.	300
Migrieren von Anwendungen nach OLAP Starter Kit Version 7.2	301
Bekanntes Probleme und Einschränkungen.	302

Fehlende EQD-Dateien des OLAP- Tabellenkalkulations-Add-In	304
Information Catalog Manager Systemverwaltung	305
Initialisierungsprogramm von Information Catalog Manager	305
Lizenzfragen	305
Installationsfragen	305
Zugreifen auf Informationskataloge von DB2 Ver- sion 7 mit Information Catalog Manager von DB2 Version 7	306
Einrichten eines Informationskatalogs	307
Austauschen von Metadaten mit anderen Produk- ten	307
Austauschen von Metadaten mit dem Befehl "flgn- xoln"	308
Austauschen von Metadaten mit dem Befehl "MDIS- DGC"	308
Aufrufen von Programmen	308
Information Catalog Manager Programming Guide and Reference	309
Information Catalog Manager Reason Codes	309
Information Catalog Manager Benutzer- handbuch	311
Information Catalog Manager: Onlinenachrich- ten	313
Nachricht FLG0260E	313
Nachricht FLG0051E	313
Nachricht FLG0003E	313
Nachricht FLG0372E	313
Nachricht FLG0615E	313
Information Catalog Manager: Onlinehilfefunk- tion	315
Information Catalog Manager für das Web	315
DB2 Warehouse Manager Installation	317
Softwarevoraussetzungen für Warehouse- Umsetzungsprogramme	317
Connector für SAP R/3	317
Vorbedingungen zur Installation	317
Connector für das Web	317
Vorbedingungen zur Installation	317
Query Patroller Administration Guide	319
DB2 Query Patroller Client is a Separate Compo- nent	319
Migrating from Version 6 of DB2 Query Patroller Using dqpmigrate	319
Enabling Query Management	319
Location of Table Space for Control Tables	319
New Parameters for dqpstart Command	320
New Parameter for iwmcmd Command	320
New Registry Variable: DQP_RECOVERY_INTER- VAL	321
Starting Query Administrator	321
User Administration	321

Creating a Job Queue	321
Using the Command Line Interface	321
Query Enabler Notes	322
DB2 Query Patroller Tracker may Return a Blank Column Page	322
Query Patroller and Replication Tools	322
Appendix B. Troubleshooting DB2 Query Patroller Clients	322

Lernprogramm für das Informationsmanagement

Überarbeitetes Lernprogramm für das Informationsmanagement

FixPak 2 enthält überarbeitete Versionen des Lernprogramms für das Informationsmanagement und der Beispieldatenbank der Data Warehouse-Zentrale. Dadurch werden einige Probleme beseitigt, die in Version 7.1 bestanden. Gehen Sie wie folgt vor, um die überarbeitete Version der Beispieldatenbank für die Data Warehouse-Zentrale einzusetzen:

Falls Sie die Beispieldatenbanken noch nicht installiert haben, erstellen Sie über die Klickstartleiste **Erste Schritte** neue Beispieldatenbanken. Klicken Sie **Start** an, und wählen Sie **Programme** → **IBM DB2** → **Erste Schritte** aus.

Wenn Sie die Beispieldatenbanken bereits früher installiert hatten, löschen Sie die Beispieldatenbanken DWCTBC, TBC_MD und TBC. Hatten Sie Daten hinzugefügt, die Sie auch weiterhin in den Beispieldatenbanken verwenden wollen, sichern Sie diese vor dem Löschen der Datenbanken. Gehen Sie wie folgt vor, um die drei Beispieldatenbanken zu löschen:

1. Klicken Sie **Start** an, und wählen Sie **Programme** → **IBM DB2** → **Befehlsfenster** aus, um das DB2-Befehlsfenster zu öffnen.
2. Geben Sie im DB2-Befehlsfenster die folgenden drei Befehle ein, wobei Sie nach jedem Befehl die Eingabetaste drücken müssen:

```
db2 drop database dwctbc  
db2 drop database tbc_md  
db2 drop database tbc
```
3. Schließen Sie das DB2-Befehlsfenster.
4. Erstellen Sie über die Klickstartleiste **Erste Schritte** neue Beispieldatenbanken. Klicken Sie **Start** an, und wählen Sie **Programme** → **IBM DB2** → **Erste Schritte** aus.

Data Warehouse-Zentrale Verwaltung

Fehlerbehebung

Informationen zur Fehlerbehebung in der Data Warehouse-Zentrale finden Sie im Handbuch „DB2 Troubleshooting Guide“.

Definieren von Excel als Warehouse-Quelle

Im Abschnitt zum Einrichten von Warehouse-Quellen für Nicht-DB2-Datenbanken unter Windows in Kapitel 3 fehlt im Abschnitt zu Microsoft Excel ein Schritt. Der neue Schritt ist unten als Schritt 3 aufgeführt.

Wenn Sie den ODBC-Treiber von Microsoft Excel 95/97 zum Zugriff auf Excel-Tabellenkalkulationen verwenden, müssen Sie eine benannte Tabelle für jedes der Arbeitsblätter in der Tabellenkalkulation erstellen. Gehen Sie wie folgt vor, um eine benannte Tabelle für jedes Arbeitsblatt zu erstellen:

1. Wählen Sie die gewünschten Spalten und Zeilen aus.
 2. Klicken Sie **Excel** → **Insert** → **Name** → **Define** an.
 3. Stellen Sie sicher, dass das Feld **Refers to** des Fensters **Define Name** die in Schritt 1 ausgewählten Zellen enthält. Ist dies nicht der Fall, klicken Sie das Symbol ganz rechts im Feld **Refers to** an, um alle ausgewählten Zellen aufzunehmen.
 4. Geben Sie einen Namen für die markierten Daten ein, oder verwenden Sie den Standardnamen.
 5. Klicken Sie **OK** an.
-

Definieren und Ausführen von Prozessen

Im Abschnitt zum Starten eines Schritts von außerhalb der Data Warehouse-Zentrale in Kapitel 5, Definieren und Ausführen von Prozessen, sollte angemerkt werden, dass auf der Warehouse-Server-Workstation auf der Agenten-Site JDK 1.1.8 oder höher erforderlich ist, wenn Sie einen Schritt mit einem DBCS-Namen starten.

Dialog "Metadaten exportieren"

Schritt 5 im Unterabschnitt "Exportieren von Metadaten in eine Befehlssprachendatei" im Abschnitt "Exportieren und Importieren von Metadaten der Data Warehouse-Zentrale" in Kapitel 12 sollte wie folgt lauten:

Wenn Sie keine Zeitplaninformationen exportieren wollen, die zu den Prozessen gehören, die Sie exportieren, nehmen Sie die Markierung des Markierungsfelds **Prozesszeitpläne einschließen** zurück.

Definieren von Werten für das Programm "OS/390-JCL-Jobstrom übergeben (VWPMVS)"

Im Abschnitt "Definieren von Werten für das Programm "OS/390-JCL-Jobstrom übergeben" (VWPMVS)" wird in Schritt 8 angegeben, dass Sie eine Datei .netrc in demselben Verzeichnis wie die JES-Datei definieren müssen. In Wirklichkeit wird die Datei .netrc vom Programm erstellt. Wenn die Datei nicht vorhanden ist,

erstellt das Programm sie im Ausgangsverzeichnis. Wenn im Ausgangsverzeichnis bereits eine Datei .netrc vorhanden ist, benennt das Programm sie um und erstellt eine neue Datei. Wenn das Programm die Verarbeitung beendet hat, löscht es die neu erstellte Datei .netrc und benennt die ursprüngliche Datei wieder in .netrc um.

Änderungen des Anhangs mit der Data Warehouse-Beispieldatenbank

- Im Anhang mit der Data Warehousing-Beispieldatenbank sollte im Abschnitt "Anzeigen und Ändern der Beispielmetadaten" die Tabelle GEOGRAPHIES in der Liste der Quellentabellen enthalten sein.
- Im Abschnitt "Hochstufen der Schritte" des Anhangs mit der Data Warehousing-Beispieldatenbank ist die folgende Aussage in der Prozedur zum Hochstufen von Schritten in den Produktionsmodus nicht korrekt, da die Zieltabelle beim Hochstufen des Schritts in den Testmodus erstellt wurde:

Die Data Warehouse-Zentrale startet die Erstellung der Zieltabelle und zeigt ein Statusfenster an.

Nachrichten der Data Warehouse-Zentrale

Unter Microsoft Windows NT und Windows 2000 zeichnet die Data Warehouse-Zentrale Ereignisse im Systemereignisprotokoll auf. Die Ereignis-ID stimmt mit der Nachrichtennummer der Data Warehouse-Zentrale überein. Informationen zu den Nachrichten der Data Warehouse-Zentrale finden Sie in *Fehlernachrichten*.

BEIM ERSTELLEN DER BEISPIELDATENBANK ANGEZEIGTE NACHRICHTEN

Die beim Erstellen der Beispieldatenbank für die Data Warehouse-Zentrale angezeigten Nachrichten sind momentan leider nur in Englisch verfügbar.

Erstellen einer Modellstruktur und Laden von Daten in den DB2 OLAP Integration Server

In dem Beispiel in Abbildung 20 ist ein Fehler. Folgende Befehle sind richtig:

```
"C:\IS\bin\olapicmd" < "C:\IS\Batch\my_script.script" >
"C:\IS\Batch\my_script.log"
```

Die doppelten Anführungszeichen um "C:\IS\bin\olapicmd" sind erforderlich, wenn der Name eines Verzeichnisses im Pfad ein Leerzeichen enthält, beispielsweise "Meine Dateien".

Verwenden von Classic Connect mit der Data Warehouse-Zentrale

- In Anhang F, Verwendung von Classic Connect für die Data Warehouse-Zentrale, wurde der Abschnitt "Installieren des ODBC-Treibers CROSS ACCESS" durch folgende Informationen ersetzt:

Installieren Sie den ODBC-Treiber CROSS ACCESS, indem Sie eine angepasste Installation von DB2 Warehouse Manager Version 7 durchführen und die Komponente für die Classic Connect-Treiber auswählen. Der Treiber wird nicht im Rahmen einer Standardinstallation von DB2 Warehouse Manager installiert.

Der ODBC-Treiber CROSS ACCESS wird im Unterverzeichnis ODBC32 des Verzeichnisses SQLLIB installiert. Nach Abschluss der Installation müssen Sie den Pfad für den Treiber (beispielsweise C:\Programmdateien\SQLLIB\ODBC32) manuell zur Systemumgebungsvariablen PATH hinzufügen. Wenn Sie eine andere Version des ODBC-Treibers CROSS

ACCESS verwenden, fügen Sie den Pfad ...\\SQLLIB\\ODBC32\\ vor dem Pfad der anderen Version ein. Das Betriebssystem verwendet das erste Verzeichnis im Pfad, das den ODBC-Treiber CROSS ACCESS enthält.

- Die folgende Prozedur sollte Anhang F, Verwendung von Classic Connect für die Data Warehouse-Zentrale, hinzugefügt werden:

Installieren des ODBC-Treibers von Classic Connect:

1. Legen Sie Ihre DB2 Warehouse Manager-CD-ROM in das CD-ROM-Laufwerk ein. Der Assistent wird geöffnet.
2. Klicken Sie im Assistenten auf **Install**.
3. Stellen Sie im Fenster **Select Products** sicher, dass das Markierungsfeld **DB2 Warehouse Manager** ausgewählt ist, und klicken Sie **Next** an.
4. Wählen Sie **Custom** im Fenster **Select Installation Type** aus und klicken Sie **Next** an.
5. Wählen Sie **Classic Connect Drivers** und **Warehouse Agent** im Fenster **Select Components** aus, heben Sie die Markierung aller anderen Felder auf, und klicken Sie **Next** an.
6. Prüfen Sie Ihre Auswahlen im Fenster **Start Copying Files**. Wenn Sie einige Ihrer Auswahlen ändern wollen, klicken Sie **Back** an, um zum Fenster zurückzukehren, in dem Sie die Auswahl ändern können. Klicken Sie **Next** an, um die Installation zu starten.

Umgebungsdatenstruktur der Data Warehouse-Zentrale

Tabelle 34 in Anhang G, Umgebungsdatenstruktur der Data Warehouse-Zentrale, enthält einen falschen Eintrag. C:\Programmdateien\SQLLIB\ODBC32 wird der Umgebungsvariablen PATH *nicht* hinzugefügt. Die einzige Aktualisierung, die an der Umgebungsvariable PATH vorgenommen wird, ist C:\Programmdateien\SQLLIB\BIN.

Verwenden des Umsetzungsprogramms "Daten vertauschen"

Im Handbuch wird angegeben, dass das Umsetzungsprogramm "Daten vertauschen" eine Zieltabelle auf der Grundlage von Parametern erstellen kann. Nicht erwähnt wird jedoch, dass die generierte Zieltabelle nicht die gewünschten Ausgabespalten hat, die explizit in der Zieltabelle erstellt werden müssen.

Zugreifen auf Daten von DB2 Version 5 mit dem Warehouse-Agenten von DB2 Version 7

Warehouse-Agenten von DB2 Version 7 unterstützen bei einer Konfiguration mit dem Installationsprozess von DB2 Version 7 Daten von DB2 Version 6 und Version 7. Wenn Sie auf Daten von DB2 Version 5 zugreifen müssen, haben Sie dazu zwei Möglichkeiten:

- Migrieren Sie die Server von DB2 Version 5 auf DB2 Version 6 oder Version 7.
- Modifizieren Sie auf dem entsprechenden Betriebssystem die Konfiguration des Agenten so, dass der Zugriff auf Daten von DB2 Version 5 möglich wird.

Warehouse-Agenten von DB2 Version 7 bieten keine Unterstützung für den Zugriff auf Daten von DB2 Version 2 oder sonstigen früheren Versionen.

Migrieren von Servern von DB2 Version 5

Informationen zum Migrieren von Servern von DB2 Version 5 finden Sie in *DB2 Universal Database Einstieg (Quick Beginnings)* für Ihre Betriebssystem.

Ändern der Agentenkonfiguration

Nachfolgend wird beschrieben, wie Sie auf den einzelnen Betriebssystemen die Konfiguration des Agenten ändern können. Wenn Sie die DB2-Server auf DB2 Version 6 oder neuer migrieren, entfernen Sie die Änderungen, die Sie an der Konfiguration vorgenommen haben.

UNIX-Warehouse-Agenten

Gehen Sie wie folgt vor, um einen UNIX-Warehouse-Agenten für den Zugriff auf Daten von DB2 Version 5 mit CLI oder ODBC einzurichten:

1. Installieren Sie DB2 Run-Time Client für Version 6. Sie können Run-Time Client durch Auswahl der entsprechenden Funktion an der folgenden URL herunterladen:
`http://www.ibm.com/software/data/db2/udb/support`
2. Aktualisieren Sie die Konfigurationsdatei des Warehouse-Agenten, damit die Umgebungsvariable DB2INSTANCE auf ein Exemplar von DB2 Version 6 zeigt.
3. Katalogisieren Sie alle Datenbanken in diesem Exemplar von DB2 Version 6, auf das der Warehouse-Agent zugreifen soll.
4. Stoppen Sie den Dämonprozess, indem Sie den Befehl KILL mit der Dämonprozess-ID des Agenten absetzen. Daraufhin wird der Agentendämon automatisch neu gestartet. Sie benötigen Root-Berechtigung, um den Prozess mit dem Befehl KILL abzubrechen.

Warehouse-Agenten unter Microsoft Windows NT, Windows 2000 und OS/2

Gehen Sie wie folgt vor, um einen Warehouse-Agenten unter Microsoft NT, Windows 2000 oder OS/2 für den Zugriff auf Daten von DB2 Version 5 einzurichten:

1. Installieren Sie DB2 Connect Enterprise Edition Version 6 auf einer Workstation, auf welcher der Warehouse-Agent von DB2 Version 7 nicht installiert ist.
DB2 Connect Enterprise Edition ist im Lieferumfang von DB2 Universal Database Enterprise Edition und DB2 Universal Database Enterprise - Extended Edition enthalten. Wenn Version 6 eines dieser beiden Produkte installiert ist, brauchen Sie DB2 Connect nicht separat zu installieren.

Einschränkung: Sie können auf einer Windows NT- oder OS/2-Workstation nicht mehrere DB2-Versionen installieren. Sie können DB2 Connect auf einer anderen Windows NT-, OS/2- oder UNIX-Workstation installieren.
2. Konfigurieren Sie den Warehouse-Agenten und DB2 Connect Version 6 für den Zugriff auf Daten von DB2 Version 5. Weitere Informationen finden Sie im *DB2 Connect Benutzerhandbuch*. Die folgenden Schritte geben eine Übersicht über die erforderlichen Schritte:
 - a. Verwenden Sie auf einem System mit DB2 Version 5 den DB2- Befehlszeilenprozessor, um die Datenbank der Version 5 zu katalogisieren, auf die der Warehouse-Agent zugreifen soll.
 - b. Katalogisieren Sie auf dem DB2 Connect-System folgende Elemente mit dem DB2-Befehlszeilenprozessor:
 - Den TCP/IP-Knoten für das System mit DB2 Version 5
 - Die Datenbank für das System mit DB2 Version 5
 - Den DCS-Eintrag für das System mit DB2 Version 5
 - c. Katalogisieren Sie auf der Workstation des Warehouse-Agenten folgende Elemente mit dem DB2-Befehlszeilenprozessor:
 - Den TCP/IP-Knoten für das DB2 Connect-System
 - Die Datenbank für das DB2 Connect-System

Informationen zur Katalogisierung von Datenbanken finden Sie in *DB2 Universal Database Installation und Konfiguration Ergänzung*.

3. Binden Sie auf der Workstation des Warehouse-Agenten das DB2-CLI-Paket an alle Datenbanken, auf die mit DB2 Connect zugegriffen werden soll.

Die folgenden DB2-Befehle sind Beispiele für das Binden von v5database, einer gedachten Datenbank von DB2 Version 5. Setzen Sie die folgenden Befehle mit dem DB2-Befehlszeilenprozessor ab. db2cli.lst und db2ajgrt befinden sich im Verzeichnis \sqllib\bnd.

```
db2 connect to v5database user benutzer-id
using kennwort
db2 bind db2ajgrt.bnd db2 bind @db2cli.lst blocking all grant public
```

Dabei gilt: *benutzer-id* ist die Benutzer-ID für die Datenbank der Version 5 und *kennwort* ist das Kennwort für die Benutzer-ID.

Beim Binden von db2cli.list an die Datenbank von DB2 Version 5 tritt ein Fehler auf. Dies liegt daran, dass große Objekte (LOBs) in dieser Konfiguration nicht unterstützt werden. Dieser Fehler hat keine Auswirkungen auf den Zugriff des Warehouse-Agenten auf die Datenbank von DB2 Version.

Für den Zugriff auf Daten von DB2 Version 5 mit DB2 Connect ist das FixPak 14 für DB2 Universal Database Version 5 erforderlich, das seit Juni 2000 verfügbar ist. Informationen dazu finden Sie unter der APAR-Nummer JR14507 in diesem FixPak.

Das Programm "ERwin MetaData Extract"

Inhalt

Softwarevoraussetzungen

Programmdateien

Erstellen von Befehlssprachendateien

Importieren einer Befehlssprachendatei in die Data Warehouse-Zentrale

Importieren einer Befehlssprachendatei in Information Catalog Manager

Fehlerbehebung

Zuordnung von ERwin zur Data Warehouse-Zentrale

Zuordnung von ERwin zu Information Catalog Manager

Dieser Abschnitt beschreibt die Verwendung des Programms „ERwin Metadata Extract“ von IBM, um Metadaten aus einer ER1-Datei zu extrahieren und eine Befehlssprachendatei für die DB2-Data Warehouse-Zentrale oder Information Catalog Manager (DataGuide) zu erstellen.

Das Metadatenextraktionsprogramm extrahiert alle physischen Objekte wie z. B. Datenbanken, Tabellen und Spalten, die in einer ER1-Eingabedatei gespeichert sind und schreibt das Metadatenmodell in eine Befehlssprachendatei für die Data Warehouse-Zentrale oder Information Catalog Manager.

Das logische Modell für Information Catalog Manager, bestehend aus Entitäten und Attributen, wird auch extrahiert und erstellt, wobei alle wichtigen Abhängigkeitsbefehle zwischen Objekten erstellt werden, wie z. B. zwischen Datenbanken und Tabellen oder Tabellen und Entitäten. Für Tabellen ohne Datenbanken wird eine Standarddatenbank mit dem Namen DATABASE erstellt. Für Tabellen ohne Schema wird das Standardschema USERID verwendet. Der Name der ER1-Datei wird als Modellname verwendet. Weitere Informationen zur Zuordnung von ER1-Attributen zu Befehlen der Data Warehouse-Zentrale und von Information Catalog Manager finden Sie in den Abschnitten "Zuordnung von ERwin zur DB2 Data Warehouse-Zentrale" und "Zuordnung von ERwin zu Information Catalog Manager."

Das Metadatenextraktionsprogramm unterstützt alle ER1-Modelle mit relationalen Datenbanken, einschließlich DB2, Informix, Oracle, Sybase, ODBC-Datenquellen und Microsoft SQL Server.

Softwarevoraussetzungen

Die folgenden Softwarevoraussetzungen müssen erfüllt sein, um das Metadatenextraktionsprogramm ausführen zu können:

- Windows NT 4.0 oder höher
- ERwin 3.5.2 mit Service Pack 3 Build 466

Die folgenden Softwarevoraussetzungen müssen erfüllt sein, um die Befehls-sprachendatei von ERwin importieren zu können:

Für die Data Warehouse-Zentrale:

IBM DB2 Universal Database Version 7.2

Für Information Catalog Manager:

IBM DB2 Warehouse Manager 7.2

Die Schablonen für die Befehlssprachendateien (.tag) müssen sich in dem Verzeichnis befinden, auf das die Umgebungsvariable VWS_TEMPLATES zeigt.

Die Typen für die Befehlssprachendateien (.typ) müssen sich in dem Verzeichnis befinden, auf das die Umgebungsvariable DGWPATH zeigt.

Programmdateien

Das Metadatenextraktionsprogramm wird in dem Unterverzeichnis sqllib\bin in dem DB2-Verzeichnis von IBM installiert. Das Programm installiert die folgenden Dateien in Ihrem Verzeichnis:

flgerwin.exe

Hauptmigrationsprogramm

erwext.dll

Die DLL für den Befehlssprachendatei-Generator

cdmerwsn.dll

Die DLL für die API-Oberflächenklasse von ERwin

Setzen Sie den Befehl flgerwin an einer Eingabeaufforderung ab, um das Extraktionsprogramm zu starten.

Erstellen von Befehlssprachendatei

Führen Sie das Programm flgerwin.exe aus, und geben Sie zwei Hauptparameter an, um eine Befehlssprachendatei für die Data Warehouse-Zentrale oder Information Catalog Manager zu erstellen. Der erste Parameter ist die ER1-Datei, aus der die Metadaten extrahiert werden sollen. Der zweite Parameter ist der Name der Ausgabedatei der Befehlssprache. Das Extraktionsprogramm fügt standardmäßig den Parameter MERGE zur Befehlssprachendatei der Data Warehouse-Zentrale hinzu.

Der Befehl hat die folgende Syntax:

```
flgerwin eingabedatei.er1 ausgabedatei.tag [-dwc] [-icm]
[-m] [-u] [-a] [-d]
```

Wenn Sie ein Sternschema erstellen wollen, hat der Befehl die folgende Syntax:

```
flgerwin eingabedatei.er1 ausgabedatei.tag [-dwc]
[-starschema]
```

- dwc** Erstellt eine Befehlssprachendatei für die Data Warehouse-Zentrale. Für -dwc sind die optionalen Parameter -m und -starschema verfügbar.
- icm** Erstellt eine Befehlssprachendatei für Information Catalog Manager. Für -icm sind die optionalen Parameter -m, -u, -a und -d verfügbar.
- starschema**
Erstellt eine Befehlssprachendatei mit einem Modellsternschema für ERwin.
- m** Gibt die Aktion MERGE für das Objekt an.
- u** Gibt die Aktion UPDATE für das Objekt an.
- a** Gibt die Aktion ADD für das Objekt an.
- d** Gibt die Aktion DELETE für das Objekt an.

Das Metadatenextraktionsprogramm arbeitet mit Metadaten, nicht mit Daten. Nachdem Sie den Import der ERwin-Befehlssprachendatei beendet haben, und bevor Sie die Zieltabelle verwenden, müssen Sie die Kennwörter und Benutzer-IDs abgleichen. Gehen Sie wie folgt vor, um Metadaten mit vorhandenen Datenbankdaten zusammenzufügen:

Ändern Sie unter **Merkmale** —> **Datenbank** —> **Benutzer-ID** die Benutzer-ID und das Kennwort für die Data Warehouse-Zentrale, damit sie mit der Benutzer-ID und dem Kennwort der zusammengeführten Datenbank übereinstimmen.

Mit dem Metadatenextraktionsprogramm können Sie eine Befehlssprachendatei als Ziel importieren. Bei neu importierten Metadaten sind die Tabellen noch nicht gefüllt. Sie können diese Tabellen als logische oder physische Darstellungen anzeigen und danach einen Warehouse-Schritt erstellen, um die Tabellendefinitionen zu füllen, die aus ERwin importiert wurden.

Die ER1-Eingabedatei muss sich in einem beschreibbaren Status befinden. Nachdem das Metadatenextraktionsprogramm ausgeführt wurde, ist nur noch der Lesezugriff auf die Datei erlaubt. Verwenden Sie einen Befehl wie im folgenden Beispiel, um die Datei in den Schreib-/Lesemodus zu versetzen:

```
attrib -r erwinsimplemode.er1
```

Dabei ist erwinsimplemode.er1 der Name der unstrukturierten ERwin-Datei.

Das Metadatenextraktionsprogramm speichert die ER1-Datei als lesbare Datei, wenn die Datei in einer aktuellen ERwin-Sitzung verwendet wird oder wenn eine Fehlerbedingung entdeckt wurde. Sie erhalten möglicherweise eine Fehlermeldung über eine unnormale Programmbeendigung, wenn für die ER1-Datei nur Lesezugriff besteht. Das Metadatenextraktionsprogramm zeigt den Tabellennamen an, den es zurzeit verwendet. Sie empfangen eine Informationsnachricht, wenn das Metadatenextraktionsprogramm die Verarbeitung beendet hat.

Wenn Sie Sternschemen erstellen, indem Sie Dimensionstabellen automatisch mit Fakttabellen verknüpfen, kann die Verarbeitung lange dauern. Dies ist abhängig davon, wieviele Tabellen Sie verwenden. Während der Verarbeitung sind die automatisch verknüpften Zeilen grün. Nach dem Speichern sind die Zeilen schwarz.

Verwenden Sie den automatisch generierten Integritätsbedingungsnamen, um sicherzustellen, dass der Integritätsbedingungsname eindeutig ist.

Während der Verarbeitung erhalten Sie möglicherweise eine Nachricht, dass doppelte Spalten gefunden wurden und die Spalte nicht extrahiert wird. Das ist eine Informationsnachricht und hat keine Auswirkungen auf die erfolgreiche Fertigstellung des Extraktionsprogramms. Diese Nachricht wird angezeigt, wenn der physische Name eines Fremdschlüssels derselbe ist, wie der physische Name einer Spalte in der Tabelle, die zurzeit verarbeitet wird.

Importieren einer Befehlssprachendatei in die Data Warehouse-Zentrale

Es gibt zwei Möglichkeiten, eine Befehlssprachendatei in die Data Warehouse-Zentrale zu importieren. Sie können die Data Warehouse-Zentrale oder die Befehlszeile verwenden.

Gehen Sie wie folgt vor, um eine Befehlssprachendatei über die Data Warehouse-Zentrale zu importieren:

1. Klicken Sie **Start** → **Programme** → **IBM DB2** → **Steuerzentrale** an. Die DB2-Steuerzentrale wird geöffnet.
2. Öffnen Sie die Data Warehouse-Zentrale, und melden Sie sich an.
3. Klicken Sie **Warehouse** mit Maustaste 2 an. Das Fenster „Importieren“ wird geöffnet.
4. Klicken Sie **Metadaten importieren** → **ERwin** an. Das Fenster **Metadaten importieren** wird geöffnet.
5. Geben Sie den Namen der Eingabe-Befehlssprachendatei in das Feld **Eingabedatei** ein, und klicken Sie **OK** an.
6. Wählen Sie das Markierungsfeld **Sternschema extrahieren** aus, um ein ERwin-Sternschema-Metadatenmodell als Warehouse-Schema zu definieren.

Nachdem der Import beendet ist, können Sie **Sicht** → **Aktualisieren** anklicken, um den neuen Schritt anzuzeigen.

Geben Sie den folgenden Befehl ein, um eine Befehlssprachendatei über die Befehlszeile zu importieren:

```
iwh2imp2  
befehlsdateiname protokollpfadname  
zielsteuerungs-db benutzer-id  
kennwort
```

befehlsdateiname

Der vollständige Pfad und Dateiname der Befehlssprachendatei.

protokollpfadname

Der vollständige Pfadname der Protokolldatei.

zielsteuerungs-db

Der Name der Zieldatenbank für den Import.

benutzer-id

Die Benutzer-ID, die verwendet wird, um auf die Steuerungsdatenbank zuzugreifen.

kennwort

Das Kennwort, das verwendet wird, um auf die Steuerungsdatenbank zuzugreifen.

Sie können die Befehlssprachendatei ändern, um eine DB2-Datenbankdefinition zu ändern, damit Sie eine Quelle in der Data Warehouse-Zentrale ist:

- Ändern Sie für alle Datenbanken, die Sie als Quelle verwenden wollen, den Befehl ISWH von ISWH(Y) in ISWH(N).
- Ändern Sie für alle Datenbanken, die Sie als Quelle verwenden wollen, den Abhängigkeitsbefehl von :RELTYPE.TYPE(LINK) SOURCETYPE(SCGTARIR) TARGETYPE(DATABASE) in :RELTYPE.TYPE(LINK) SOURCETYPE(SCGSR CIR) TARGETYPE(DATABASE).

Wenn die Befehlssprachendatei importiert ist, empfangen Sie möglicherweise die folgende Nachricht:

Nachricht: DWC13238E Das Objekt des Typs "COLUMN", das durch das Schlüsselwort "DBNAME(____) OWNER(____) TABLE(____) COLUMNS(____)" identifiziert wird, ist in der Befehlssprachendatei zweimal definiert.

Dies ist eine Informationsnachricht, und Ihr Import war erfolgreich. Sie empfangen diese Nachricht möglicherweise, wenn Sie eine Entität haben, die einen Fremdschlüssel mit demselben Namen hat oder eine Entität mit ähnlich benannten Spalten, die durch Verkürzung oder andere ähnliche Umstände betroffen sind. Überprüfen Sie Ihr Modell auf doppelte Spaltennamen, und nehmen Sie notwendige Anpassungen vor.

Importieren einer Befehlssprachendatei in Information Catalog Manager

Es gibt zwei Möglichkeiten, eine Befehlssprachendatei in Information Catalog Manager zu importieren. Sie können Information Catalog Administrator oder die Befehlszeile verwenden.

Gehen Sie wie folgt vor, um eine Befehlssprachendatei über Information Catalog Administrator zu importieren:

1. Klicken Sie **Start** —> **Programme** —> **DB2** —> **Information Catalog Manager** an.
2. Klicken Sie **Katalog** —> **Importieren** an. Das Fenster „Importieren“ wird geöffnet.
3. Klicken Sie **Suchen** an, um nach der Befehlssprachendatei zu suchen, und klicken Sie danach **Importieren** an.

Nachdem der Import beendet ist, können Sie das Symbol **Themen** doppelt anklicken, um ein Fenster zu öffnen, das alle importierten Modelle und Datenbanken anzeigt.

Geben Sie den folgenden Befehl ein, um eine Befehlssprachendatei über die Befehlsschnittstelle zu importieren:

```
DGUIDE /USERID  
benutzer-id /PASSWORD kennwort  
/DGNAME icm-name /IMPORT dateiname /LOGFILE  
dateiname /ADMIN /RESTART (B|C)
```

/USERID

Die Benutzer-ID, die verwendet wird, um auf die Steuerungsdatenbank zuzugreifen.

/PASSWORD

Das Kennwort für diese Benutzer-ID.

/DGNAME

Der Information Catalog Manager-Name.

/IMPORT

Der vollständige Pfad und Dateiname der Befehlssprachendatei.

/LOGFILE

Der vollständige Pfadname der Protokolldatei.

/ADMIN

Gibt an, dass Sie sich als Administrator anmelden.

/RESTART

Gibt an, dass der Import am Anfang der Befehlssprachendatei (Auswahl B) oder am letzten COMMIT-Punkt (Standardauswahl C) starten soll.

Fehlerbehebung

Wenn Sie eine Fehlernachricht empfangen, schlagen Sie die Nachricht hier nach, um zu erfahren, mit welchen Aktionen Sie den Fehler auflösen können.

ERwin-ER1-Eingabedatei oder Befehlsausgabedatei fehlt.

Das Metadatenextraktionsprogramm erfordert zwei Parameter in bestimmter Reihenfolge. Der erste Name ist der Name der ER1-Datei, und der zweite Name ist der Name der Befehlsausgabedatei. Wenn Sie den Namen einer vorhandenen Befehlssprachendatei angeben, wird diese Datei überschrieben.

Windows system abnormal program termination.

Der Status der ER1-Eingabedatei erlaubt wahrscheinlich nur den Lesezugriff. Diese Nachricht kann angezeigt werden, wenn beim Speichern der ER1-Datei ein Fehler aufgetreten ist und das Metadatenextraktionsprogramm nur den Lesezugriff auf die Datei erlaubt. Setzen Sie den Befehl

```
attrib -r eingabedatei.erl
```

in einer Befehls-Shell ab, um den Schreib-/Lesezugriff auf die ER1-Datei zu erlauben.

Befehlsdatei ... konnte nicht geöffnet werden.

Überprüfen Sie, ob Systemfehler vorhanden sind, die verhindern können, dass eine Datei auf dem aktuellen Laufwerk erstellt oder geöffnet wird.

Pfad zu Schablonendateien wurde nicht gefunden.

Die Umgebungsvariable VWS_TEMPLATES ist nicht gesetzt. Überprüfen Sie, ob die Data Warehouse-Zentrale installiert ist.

Pfad zu Typendateien wurde nicht gefunden.

Die Umgebungsvariable DGWPATH ist nicht gesetzt. Überprüfen Sie, ob die Data Warehouse-Zentrale installiert ist.

Nicht unterstützte Serverversion: ...

Die ER1-Eingabedatei, aus der Sie extrahieren wollen, ist auf einem Zielsystem gespeichert, der von dem Programm nicht unterstützt wird. Starten Sie ERwin, öffnen Sie die ER1-Datei, und klicken Sie danach **Server** —> **Ziel-Server** und die passende Version [siehe Softwarevoraussetzungen] an. Speichern Sie die ER1-Datei.

Unbekannter ERwAPI-Fehler.

Ein ERwin-API-Fehler ist aufgetreten, und das Programm konnte keine weiteren Informationen zu dem Fehler ermitteln. Stellen Sie sicher, dass ERwin 3.5.2 installiert ist. Sie müssen die ERwin-API registrieren.

Führen Sie den folgenden Befehl in dem Verzeichnis aus, in dem Ihre ERwin-Programmdateien installiert sind, um die ERwin-API zu registrieren: **regsvr32 er2api32.dll** . Sie erhalten eine Nachricht, dass der Befehl „DllRegisterServer“ in der Datei er2api32.dll nicht erfolgreich war. Sie können das Extraktionsprogramm über die Data Warehouse-Zentrale oder durch Absetzen des Befehls „flgerwin“ an einer Befehls-Shell starten.

Extraktionsprogrammfehler: ...

Überprüfen Sie die Fehlernachricht, und führen Sie die entsprechenden Aktionen aus. In den meisten Fällen handelt es sich dabei um einen internen Extraktionsprogrammfehler, und das Problem muss an einen IBM Ansprechpartner übergeben werden.

Unbekannter Extraktionsprogrammfehler.

Ein unbekannter Fehler ist aufgetreten. In den meisten Fällen handelt es sich dabei um einen internen Fehler, und das Problem muss an einen IBM Ansprechpartner übergeben werden.

Extraktionsprogramm wurde mit Fehler(n) beendet.

Ein Fehler ist aufgetreten, der die Fertigstellung des Extraktionsprogramms verhindert. Überprüfen Sie zusätzliche Fehlernachrichten, um das Problem zu lösen, oder kontaktieren Sie einen IBM Ansprechpartner.

Zuordnung von ERwin zur DB2 Data Warehouse-Zentrale

Dieser Abschnitt zeigt, wie die Hauptobjektattribute von ERwin den Befehlen der Data Warehouse-Zentrale entsprechen:

Datenbank - WarehouseDatabase.tag oder SourceDatabase.tag

ERwin	Befehlszeilenbefehl	Data Warehouse-Zentrale
Diagrammname	NAME	Name der Warehouse-Quelle oder des Warehouse-Ziels
Diagrammverfasser	RESPNSBL	Ansprechpartner
Datenbankname	DBNAME	Datenbankname
Datenbankversion	DBTYPE	Datenbanktyp
Diagrammbeschreibung	SHRTDESC	Beschreibung

Tabelle - Table.tag

ERwin	Befehlszeilenbefehl	Data Warehouse-Zentrale
Tabellenname	NAME	Tabellenname
Tabellenname	TABLES	Tabellenname
Datenbankname	DBNAME	n. v.
Tabelleneigner	OWNER	Tabellenschema
Tabellenkommentar	SHRTDESC	Beschreibung

Spalten - Column.tag

ERwin	Befehlszeilenbefehl	Data Warehouse-Zentrale
Spaltenname	NAME	Spaltenname
Datentyp	NATIVEDT	Datentyp
Länge	LENGTH	Länge
Anzahl der Kommastellen	SCALE	Anzahl der Kommastellen
Null-Option	NULLABLE	Nullen zulassen (Markierungsfeld)
Position	POSNO	n. v.
Primärschlüssel	KEYPOSNO	n. v.
Datenbankname	DBNAME	n. v.
Tabelleneigner	OWNER	n. v.
Tabellenname	TABLES	n. v.
Spaltenkommentar	SHRTDESC	Beschreibung

Zuordnung von ERwin zu Information Catalog Manager

Dieser Abschnitt zeigt, wie die Hauptobjektattribute von ERwin den Befehlen von Information Catalog Manager entsprechen:

Datenbank - Database.tag

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
Diagrammname	NAME	Datenbankname
Diagrammverfasser	RESPNSBL	Datenbankeigner
Datenbankname	DBNAME	Datenbankname
Datenbankversion	DBTYPE	Datenbanktyp
Diagramm- beschreibung	SHRTDESC	Kurzbeschreibung

Tabelle - TableOrView.tag

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
Tabellenname	NAME	Tabellenname
Tabellenname	TABLES	Tabellenname
Datenbankname	DBNAME	Datenbankname
Tabelleneigner	OWNER	Tabelleneigner
Tabellenkommentar	SHRTDESC	Kurzbeschreibung
ERwin-API	TABLVIEW	Definition stellt eine Sicht dar

Spalten - ColumnOrField.tag

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
Spaltenname	NAME	Spaltenname
Datentyp	DATATYPE	Datentyp einer Spalte
Länge	LENGTH	Länge einer Spalte
Anzahl der Komma- stellen	SCALE	Anzahl der Kommastellen einer Spalte
Null-Option	NULLS	Können Spalten Null sein (?)
Position	POSNO	Spaltenposition
Primärschlüssel	KEYPOSNO	Position der Spalte im Primärschlüssel
ERwin-API	ISKEY	Ist die Spalte Teil eines Schlüssels (?)
ERwin-API	UNIKEY	Ist die Spalte ein eindeutiger Schlüssel (?)
Datenbankname	DBNAME	Datenbankname
Tabelleneigner	OWNER	Tabelleneigner
Tabellenname	TABLES	Tabellenname
Spaltenkommentar	SHRTDESC	Kurzbeschreibung
ERwin	ISTEXT	Bestehen die Daten aus Text (?)

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
ERwin-API	IDSRES	Auflösung der Daten

Modell - Model.tag

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
ER1-Dateiname	NAME	Modellname
Diagrammverfasser	RESPNSBL	Weitere Informationen...
Diagramm- beschreibung	SHRTDESC	Kurzbeschreibung

Entität - Entity.tag

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
Entitätenname	NAME	Entitätenname
Notizen	SHRTDESC	Kurzbeschreibung
Definition	LONGDESC	Langbeschreibung
Entitäteneigner	RESPNSBL	Weitere Informationen...

Attribut - Attribute.tag

ERwin	Befehlszeilenbefehl	Schnittstelle von Information Catalog Manager
Attributname	NAME	Attributname
Notizen	SHRTDESC	Kurzbeschreibung
Definition	LONGDESC	Langbeschreibung
Datentyp	DATATYPE	Datentyp eines Elements
Länge	LENGTH	Länge eines Elements

Namens- und Adressbereinigung in der Data Warehouse-Zentrale

Verwenden Sie die Data Warehouse-Zentrale und Trillium Software System, um Namens- und Adressdaten zu bereinigen. Trillium Software System ist ein Produkt zur Namens- und Adressbereinigung, das Namens- und Adressdaten erneut formatiert, standardisiert und bestätigt. Sie können Trillium Software System in der Data Warehouse-Zentrale verwenden, indem Sie die Trillium Batch System-Programme in einem benutzerdefinierten Programm starten. Das benutzerdefinierte Programm wird zur Warehouse-Baumstruktur hinzugefügt, wenn Sie die Metadaten aus einem Trillium Batch System-Script oder aus JCL importieren.

Die Data Warehouse-Zentrale bietet bereits die Integration der Tools von Vality and Evolutionary Technologies, Inc.

Voraussetzungen

- Sie müssen Trillium Software System auf der Warehouse-Agenten-Site oder auf einem fernen Host installieren.
- Auf UNIX- und Windows-Plattformen muss der Pfad zu dem Verzeichnis „bin“ von Trillium Software System zur Systemumgebungsvariablen PATH hinzugefügt werden, um den Prozess des Agenten für die Ausführung der Programme von Trillium Batch System zu aktivieren. Unter UNIX muss dies getan werden, indem Sie die Variable PATH zu der Datei IWH.environment hinzufügen, bevor Sie den Prozess vwdaemon starten.
- Die Benutzer müssen Arbeitskenntnisse der Trillium-Software haben.

In der folgenden Tabelle finden Sie die Softwarevoraussetzungen:

Betriebssystem	Erforderliche Software
UNIX	Trillium Software System Version 4.0 Data Warehouse Manager Version 7.2 Warehouse-Agent
Windows NT und Windows 2000	Trillium Software System Version 4.0 Data Warehouse Manager Version 7.2 Warehouse-Agent Für fernen Zugriff müssen auf dem Host der ftpd- und der rexecd-Dämon installiert sein.
OS/390	Trillium Software System Version 4.0, das auf dem fernen OS/390-Host installiert ist Data Warehouse Manager Version 7.2 Warehouse-Agent, das unter UNIX oder Windows NT installiert ist TCP/IP 3.2 oder höher muss installiert sein Das OS/390-Betriebssystem wird nur als ferner Host unterstützt

Komponenten von Trillium Software System

Trillium Software System besteht aus vier Hauptkomponenten: Converter, Parser, Geocoder und Matcher. Verwenden Sie die Komponenten als Funktionsmenge, um Namens- und Adressbereinigungsoperationen auszuführen. Sie können die Komponenten in dem benutzerdefinierten Programm „Trillium Batch System“ ausführen.

Converter

Verwenden Sie Converter, um die Quelldaten zu standardisieren und in das angegebene Ausgabeformat zu konvertieren.

Parser Verwenden Sie den Parser, um Namens- und Adressquelldaten zu interpretieren und Metadaten über die Quelldaten zu erstellen.

Geocoder

Verwenden Sie Geocoder, um die Quelldaten mit Postservicedaten zu vergleichen, um fehlende Informationen wie z. B. Kurierdienst oder Postleitzahlen zu ergänzen. Geocoder führt außerdem Abgleichsoperationen mit Volkszählungsdaten in den Vereinigten Staaten aus.

Matcher

Verwenden Sie Matcher, um ähnliche Namen und Adressen zu vergleichen, um doppelte Einträge zu erkennen. Sie können einen Verweisabgleich ausführen, indem Sie Matcher zum Vergleichen eines Datensatzes mit einer Gruppe von Datensätzen zu verwenden.

Verwenden von Trillium Batch System mit der Data Warehouse-Zentrale

In der Data Warehouse-Zentrale können Sie Trillium Batch System-Metadaten importieren und einen benutzerdefinierten Programmschritt erstellen. Dieser Schritt ruft ein Trillium Batch System-Script auf der lokalen oder auf einer fernen Warehouse-Agenten-Site auf. In der Data Warehouse-Zentrale ist das Trillium Batch System-Script ein Schritt mit einer Quellen- und Zieldatei. Die Quelldatei ist die Eingabedatendatei, die für den ersten Trillium Batch System-Befehl verwendet wird. Die Zieldatei ist die Ausgabedatendatei, die durch den letzten Trillium-Befehl in dem Script erstellt wird. Danach kann der Schritt in einen anderen Prozess kopiert werden, damit er mit anderen Schritten verwendet werden kann.

Die folgenden Abbildungen zeigen die Beziehung zwischen Ein- und Ausgabedatendateien von Trillium Batch System und den Quellen- und Zieldateien in der Data Warehouse-Zentrale.

```
REM Running the converter
pfcondrv -parmfile c:\tril40\us_proj\parms\pfcondrv.par
REM Running the parser
pfprsdrv -parmfile c:\tril40\us_proj\parms\pfprsdrv.par
REM Running the Matcher
cfmatdrv -parmfile c:\tril40\us_proj\parms\pfmatdrv.par
```

Abbildung 1. Trillium-Script-Beispieldatei

```
INP_FNAME01 c:\tril40\us_proj\data\convinp
INP_DDL01 c:\tril40\us_proj\dict\input.ddl
```

Abbildung 2. Inhalte der Datei pfcondrv.par

```
OUT_DDNAME c:\tril40\us_proj\data\maout
DDL_OUT_FNAME c:\tril40\us_proj\dict\parseout.ddl
```

Abbildung 3. Inhalte der Datei *pfmatdrv.par*

```
c:\Tril40\us_proj\data\convinp (Quellendatei) ->
Trillium Batch System-Schritt -> c:\tril40\us_proj\data\maout (Zieldatei)
```

Abbildung 4. Die Schrittdefinition von Trillium Batch System

Importieren von Trillium-Metadaten

Gehen Sie wie folgt vor, um Trillium-Metadaten in die Data Warehouse-Zentrale zu importieren:

1. Erstellen Sie ein Trillium Batch System-Script oder eine JCL. Sie können ein beliebiges Tool für das Schreiben von Scripts oder JCL verwenden, um die Script- oder JCL-Datei zu erstellen.
2. Klicken Sie **Warehouse** mit Maustaste 2 an, und klicken Sie **Metadaten importieren** —> **Trillium** an, um das Fenster „Trillium Batch System“ zu öffnen.
3. Geben Sie den Namen des Trillium Batch System-Scripts oder der JCL-Datei, die Sie ausführen wollen, in das Feld **Script oder JCL** ein.
4. Geben Sie den Namen der Eingabedatendatei für das Trillium Batch System-Programm, das zuerst in der angegebenen Script- oder JCL-Datei ausgeführt wird, in das Feld **Eingabedatei** ein.
5. Geben Sie den Namen der Eingabe-DDL-Datei, die die Eingabedatendatei beschreibt, in das Feld **Eingabe-DDL** ein. Diese Datei muss auf einer Warehouse-Agenten-Site zur Verfügung stehen.
6. Geben Sie den Namen der Ausgabedatendatei für das letzte Trillium Batch System-Programm in der Script- oder JCL-Datei in das Feld **Ausgabedatei** ein.
7. Geben Sie den Namen der Ausgabe-DDL-Datei, die die Ausgabedatendatei beschreibt, in das Feld **Ausgabe-DDL** ein. Diese Datei muss auf einer Warehouse-Agenten-Site zur Verfügung stehen.
8. Optional: Geben Sie den Namen der Ausgabefehlerdatei, die Sie verwenden wollen, in das Feld **Ausgabefehlerdatei** ein. Diese Fehlerdatei erfasst die Fehler bei der Ausführung des Trillium Batch System-Programms. Diese Fehler werden in dem Protokoll „stderr“ aufgezeichnet. Für lokale Hosts wird eine Standardausgabedatei erstellt, wenn Sie hier keinen Namen angeben. Weitere Informationen zu der Ausgabefehlerdatei finden Sie im Abschnitt "Fehlerbehandlung."
9. Klicken Sie die Indexzunge **Verbindung** an.
10. Wenn sich die Daten, die Sie importieren wollen, auf der Warehouse-Agenten-Site befinden, klicken Sie **Lokaler Host** an.
Wenn sich die Daten, die Sie importieren wollen, nicht auf der Warehouse-Agenten-Site befinden, klicken Sie **Ferner Host** an, und geben Sie den fernen Host an. Siehe den Abschnitt "Angaben des fernen Hosts" weiter unten in diesem Abschnitt.
11. Klicken Sie **OK** an, um die Trillium-Metadaten zu importieren und das Notizbuch zu schließen.
12. Wenn das Script oder JCL nicht auf der Standard-Agenten-Site ausgeführt wird, geben Sie die Warehouse-Agenten-Site, die Sie verwenden, in dem Notizbuch „Merkmale“ des Trillium Batch System-Schritts an.

Die folgenden Warehouse-Objekte werden zu der Warehouse-Baumstruktur hinzugefügt, wenn die Importoperation beendet ist.

- Die Schablone Trillium Batch System.*scriptname*, wobei *scriptname* der Name der Script- oder JCL-Datei ist.
- Der Trillium Batch System-Prozess.
- Der Trillium Batch System-Schritt, der das benutzerdefinierte Programm ausführt.
- Die Warehouse-Dateiquelle und das Warehouse-Dateiziel, die Sie angegeben haben, als Sie die Metadaten importiert haben. Die Dateiquelle und das Dateiziel sind festgelegt.
- Die Trillium Batch System-Programmgruppe.

Angeben des fernen Hosts

Gehen Sie wie folgt vor, um einen fernen Host anzugeben:

1. Klicken Sie **Ferner Host** an, und geben Sie den TCP/IP-Host-Namen des fernen Systems ein, das die Metadaten enthält, die Sie importieren. Wenn Sie „Ferner Host“ auswählen, wird die Zielfeld als lokale Datei erstellt, weil ferne Zielfelder nicht unterstützt werden. Sie können einen FTP-Schritt hinzufügen, um die ferne Datei in die angegebene lokale Zielfeld zu schreiben.
2. Klicken Sie in der Liste **Fernes Betriebssystem** das Betriebssystem des fernen Hosts an, auf den Sie zugreifen.
3. Geben Sie die Benutzer-ID für den fernen Host, auf den Sie zugreifen in das Feld **Ferne Benutzer-ID** ein.
4. Wählen Sie eine der folgenden Kennwortoptionen aus der Liste **Kennwortoption** aus, die Sie für den fernen Host verwenden wollen, auf den Sie zugreifen.

Kennwort nicht erforderlich

Gibt an, dass kein Kennwort erforderlich ist, um auf die Metadaten auf dem fernen Host zuzugreifen.

Kennwort abrufen

Gibt an, dass das Kennwort von einem benutzerdefinierten Programm abgerufen wird.

Geben Sie den Namen des Programms, das das Kennwort abrufen wird, in das Feld **Kennwortprogramm** ein. Das Programm muss sich auf der Warehouse-Agenten-Site befinden und das Kennwort in die erste Zeile einer Ausgabedatei schreiben.

Geben Sie die Parameter für das Kennwortprogramm in das Feld **Programmparameter** ein. Der erste Parameter muss die Ausgabedatei sein, in die das Kennwort geschrieben wird.

Kennwort später eingeben

Gibt an, dass das Kennwort zu einem späteren Zeitpunkt eingegeben wird.

Geben Sie das Kennwort in das Notizbuch „Merkmale“ für den Schritt ein, der das Trillium Batch System-Programm ausführt.

Zuordnen der Metadaten

Trillium liest die Trillium-DDL-Dateien, um die Metadaten für die Quellen- und Zieldateien zu erstellen. Die DDL-Datei wird in die folgenden Datentypen der Data Warehouse-Zentrale konvertiert:

DDL-Datentypen für UNIX, Windows NT und andere Betriebssysteme	Warehouse-Datentyp
ASCII CHARACTER	CHARACTER(<i>n</i>)
ASCII NUMERIC	
EBCDIC CHARACTER	
EBCDIC NUMERIC	
Andere Typen	NUMERIC
Anmerkung: Die Datentypen EBCDIC CHARACTER und EBCDIC NUMERIC werden nur unterstützt, wenn Trillium Software System auf dem OS/390-Betriebssystem ausgeführt wird.	

Die Variable *n* ist die Anzahl Zeichen in der Zeichenfolge.

Einschränkungen

Sie können mit Trillium-DDL und der Metadatenoperation in der Data Warehouse-Zentrale überlappende Felder in den Ein- und Ausgabe-DDL-Dateien angeben. Die entsprechenden Warehouse-Quellen- und -Zieldateien können in der Data Warehouse-Zentrale allerdings nicht mit dem SQL-Schritt oder dem Inhalt der Beispieldatenbank verwendet werden. Weil die Operation zum Importieren von Metadaten überlappende Felder ignoriert, die den gesamten Datensatz einschließen, können Sie diese Felder zwar angeben, aber sie werden nicht als Spalten in den Ergebnisquellen- und -zieldateien verwendet.

Wenn eine Fehlerdatei angegeben wird, darf der Name des Scripts keine Leerstellen enthalten.

Schreiben einer Trillium Batch System-JCL-Datei

Die folgenden Voraussetzungen müssen erfüllt werden, wenn Sie eine Trillium Batch System-JCL-Datei schreiben.

- Der Jobname muss der Benutzer-ID und einem zusätzlichen Zeichen entsprechen.
- Der Job muss an die die Verbindung haltende Ausgabeklasse weitergeleitet werden.
- Jeder Jobabschnitt, der ein Trillium Batch System-Program ausführt muss eine Anweisung SYSTEM DD enthalten, die einen permanenten Datensatz definieren. Der Datensatz enthält die Fehler aus den Trillium Batch System-Programmen. Dieser Datensatz wird automatisch gelöscht, bevor JCL übergeben wird. Weitere Informationen zu der Fehlerbehandlung und zum Berichten finden Sie im Abschnitt "Fehlerbehandlung."

Die Ausgabefehlerdatei muss angegeben werden, wenn das Script oder die JCL auf einem fernen Host ausgeführt wird. Die Fehlermeldungen werden sonst nicht erfasst und an die Data Warehouse-Zentrale zurückgegeben. Unter UNIX oder Windows besteht die einfachste Möglichkeit zum Erfassen der Fehlermeldungen im Schreiben eines anderen Scripts, das das Trillium Batch System-Script aufruft und die Standardfehler über eine Pipe in eine Ausgabedatei leitet.

```
//SYSTEM DD UNIT=&UNIT,  
//          DISP=(MOD,CATLG,KEEP),  
//          SPACE=(400,(20,20),,ROUND),  
//          DSN=&PROJPREF.&TRILVER.&PROJECT.STDERR;
```

Abbildung 5. Beispiel eines Jobabschnitts, der eine Anweisung "SYSTEM DD" enthält

Schreiben einer Trillium Batch System-Script-Datei unter UNIX und Windows

Wenn die Trillium Batch System-Script- oder -Parameterdateien relative Pfade oder Eingabedateien enthält, muss der Benutzer eine Anweisung zum Wechseln des Verzeichnisses in das Verzeichnis der Script-Datei an den Anfang der Script-Datei stellen.

Definieren eines Trillium Batch System-Schritts

Sie müssen die Trillium-Metadaten importieren, die Sie in dem Prozess verwenden wollen, bevor Sie einen Trillium Batch System-Schritt definieren. Gehen Sie wie folgt vor, um einem Prozess einen Trillium Batch System-Schritt hinzuzufügen:

1. Öffnen Sie den Prozess in dem Prozesserstellungsprogramm.
2. Klicken Sie das Symbol **Trillium Batch System** in der Palette an.
3. Klicken Sie **Trillium Batch System** —> *programmname* an. Dabei ist *programmname* der Name des Trillium Batch System-Programms, das Sie verwenden wollen.
4. Klicken Sie die Position in der Grafik an, an dem der Schritt angezeigt werden soll.
5. Führen Sie die Schritte im Abschnitt "Definieren eines Schritts, der ein benutzerdefiniertes Programm ausführt" der *DB2 Universal Database-Hilfe* aus.

Verwenden des benutzerdefinierten Programms von Trillium Batch System

Das benutzerdefinierte Programm von Trillium Batch System ist in der DB2-Data Warehouse-Zentrale Version 7.2 für Windows NT und UNIX enthalten. Der Trillium Batch System-Schritt, der erstellt wird, wenn Sie Trillium-Metadaten importieren, führt das benutzerdefinierte Programm von Trillium Batch System aus. Das benutzerdefinierte Programm ruft das Trillium Batch System-Script oder die JCL auf. Die folgende Tabelle enthält die Parameter für das Trillium Batch System-Script oder die JCL:

Parameter	Werte
Ferner Host	<ul style="list-style-type: none">• Der Standardwert ist localhost. Verwenden sie diesen Wert, wenn Trillium Batch System auf der Warehouse-Agenten-Site installiert ist.• Der Name des fernen Host, wenn Trillium Batch System auf einem fernen Betriebssystem installiert ist.
Script oder JCL	Der Name des Scripts oder der JCL
Fernes Betriebssystem	Der Name des Betriebssystems auf dem fernen Host. Dieser Parameter wird ignoriert, wenn localhost der Wert des Parameters "Ferner Host" ist. Die gültigen Werte sind: <ul style="list-style-type: none">• MVS für das OS/390-Betriebssystem• UNIX für AIX-, SUN Solaris-, HP-UX- und NUMA/Q-Betriebssysteme• WIN für die Betriebssysteme Windows NT oder 2000

Ferne Benutzer-ID	Die Benutzer-ID mit der Berechtigung, den fernen Befehl auszuführen. Dieser Parameter wird ignoriert, wenn als Name für den fernen Host localhost angegeben wurde.
Kennwortoption	<p>Die Methode zum Abfragen des Kennworts. Die gültigen Werte sind:</p> <p>ENTERPASSWORD Verwenden Sie diesen Wert, wenn das Kennwort im nächsten Parameter übergeben wird.</p> <p>PASSWORDNOTREQUIRED Verwenden Sie diesen Wert, wenn kein Kennwort erforderlich ist.</p> <p>GETPASSWORD Verwenden Sie diesen Wert, wenn ein Programmname im nächsten Parameter übergeben wird.</p> <p>Einschränkungen:</p> <ul style="list-style-type: none"> • Das Programm muss sich auf der Agenten-Site befinden, das Kennwort in die erste Zeile einer Ausgabedatei schreiben und 0 zurückgeben, wenn es erfolgreich ausgeführt wurde. • Der Wert des Parameters "Kennwort" muss der Name des Kennwortprogramms sein. • Der Wert des Parameters "Programmparameter" muss eine Zeichenfolge sein, die in doppelte Anführungszeichen eingeschlossen ist. • Der erste Parameter in der Zeichenfolge muss der Name der Ausgabedatei sein, in die das Kennwort geschrieben werden soll
Kennwort	Ein gültiger Wert ist das Kennwort oder der Name eines Kennwortprogramms. Das Kennwortprogramm muss sich lokal auf der Warehouse-Agenten-Site befinden.
Programmparameter	Die Parameter für das Kennwortprogramm.
Ausgabefehlerdatei	Der Name der Ausgabefehlerdatei.
Anmerkung: Der Datentyp für alle Parameter in dieser Tabelle ist CHARACTER.	

Fehlerbehandlung

Die Trillium Batch System-Programme schreiben Fehlnachrichten unter den Betriebssystemen Windows NT und UNIX in die Standardfehlerdatei (stderr) und unter OS/390 in den Datensatz SYSTERM.

Auf Windows NT- und UNIX-Betriebssystemen müssen die Standardfehler in eine Ausgabefehlerdatei umgeleitet werden, um die Fehler der Trillium Batch System-Programme zu erfassen.

Die JCL muss eine Anweisung SYSTERM DD enthalten, um die Fehler der Trillium Batch System-Programme auf dem OS/390-Betriebssystem zu erfassen.

Wenn Sie den Ausgabefehlerdateinamen im Fenster "Metadaten importieren" angeben, müssen Sie die Standardfehlerausgabe umleiten oder in der Fehlerdatei speichern. Die Data Warehouse-Zentrale liest die Datei und berichtet alle Zeilen als Fehlnachrichten, die die Zeichenfolge ERROR enthalten. Alle Fehlnachrichten der Trillium Batch System-Programme enthalten die Zeichenfolge ERROR.

Wenn die Ausgabefehlerdatei in einem Script oder in einer JCL, die auf der Warehouse-Agenten-Site ausgeführt wird, nicht angegeben ist, erstellt die Data Warehouse-Zentrale automatisch einen Dateinamen und leitet die Standardfehlerausgabe in diese Datei um. Wenn ein Fehler auftritt, wird die Fehlerdatei nicht gelöscht. Die Fehlerdatei wird in dem Verzeichnis gespeichert, das durch die Umgebungsvariable VWS_LOGGING angegeben wird. Der Dateiname ist *tbsudp-datum-zeit.err*. Dabei ist *datum* das Systemdatum und *zeit* die Systemzeit, zu der die Datei erstellt wird. Der folgende Dateiname zeigt das Format des Ausgabefehlerdateinamens:

tbsudp-150201-155606.err

Fehlercode

Fehlernummer	Beschreibung
0	Erfolg
4	<p>Warnung. Entweder konnte die Kennwortdatei nicht gelöscht werden, oder ein interner Fehler ist aufgetreten, während ein benutzerdefiniertes Programm von Trillium Batch System auf eine temporäre Datei zugegriffen hat.</p> <p>Überprüfen Sie den Status der Kennwortdatei oder alle temporären Dateien, die im Verzeichnis erstellt worden sind, das durch die Umgebungsvariable VWS_LOGGING angegeben wird.</p>
8	Die Anzahl an Werten der Parameter ist nicht korrekt. Schlagen Sie die korrekte Syntax in der Protokolldatei oder in der Dokumentation nach.
12	<p>Ein Problem ist aufgetreten, während das benutzerdefinierte Programm von Trillium Batch System über FTP die Verbindung zu einem fernen Host herstellen wollte.</p> <p>Überprüfen Sie die FTP-Verbindung oder den Hostnamen, die Benutzer-ID und das Kennwort.</p>
16	<p>Das benutzerdefinierte Programm von Trillium Batch System kann das Protokoll oder eine interne Datei nicht erstellen.</p> <p>Überprüfen Sie, ob der Benutzer die korrekten Berechtigungen hat und ob die Platte voll ist.</p>
20	<p>Entweder kann die OS/390-JCL nicht ausgeführt werden oder ein Fehler ist aufgetreten, während das benutzerdefinierte Programm eine Datei mit FTP vom OS/390-System löschen oder abrufen wollte.</p> <p>Überprüfen Sie die Datei JESLogFile, um die Ursache zu bestimmen.</p>
48	<p>Die Umgebungsvariable VWS_LOGGING kann nicht gefunden werden, oder die Protokolldatei kann nicht erstellt werden.</p> <p>Überprüfen Sie die Protokolldatei auf weitere Informationen.</p>
56	<p>Entweder kann das Windows NT- oder UNIX-Script nicht ausgeführt werden oder ein Fehler ist aufgetreten, während das benutzerdefinierte Programm von Trillium Batch System sich mit dem fernen Host verbinden wollte. Überprüfen Sie die Verbindung oder den Hostnamen, die Benutzer-ID und das Kennwort.</p>

500	Die Script- oder JCL-Datei gibt einen Fehler zurück, oder sie gibt keinen Fehler zurück, aber die Fehlerdatei enthält Daten. Überprüfen Sie die Protokolldatei auf weitere Informationen. Prüfen Sie unter OS/390 auch die Datei JESLogFile.
-----	--

Protokolldatei

Die Data Warehouse-Zentrale speichert alle Diagnoseinformationen in einer Protokolldatei, wenn das benutzerdefinierte Programm von Trillium Batch System ausgeführt wird. Der Name der Protokolldatei ist *tbsudp-datum-zeit.log*. Dabei ist *datum* das Systemdatum und *zeit* die Systemzeit, zu der die Datei erstellt wird. Die Protokolldatei wird in dem Verzeichnis erstellt, das durch die Umgebungsvariable *VWS_LOGGING* auf der Agenten-Site angegeben wird. Die Protokolldatei wird gelöscht, wenn das benutzerdefinierte Programm von Trillium Batch System erfolgreich ausgeführt wird.

Integration von MQSeries in die Data Warehouse-Zentrale

Die Data Warehouse-Zentrale ermöglicht Ihnen jetzt, auf Daten aus einer MQSeries-Nachrichtenwarteschlange als DB2-Datenbanksicht zuzugreifen. Es wird ein Assistent angeboten, um eine DB2-Tabellenfunktion und die DB2-Sicht zu erstellen, mit der Sie auf die Daten zugreifen können. Jede MQSeries-Nachricht wird als begrenzte Nachricht behandelt, die anhand Ihrer Spezifikationen syntaktisch analysiert wird und als Ergebniszeile zurückgegeben wird. Auf MQSeries-Nachrichten, die XML-Dokumente sind, kann außerdem als Warehouse-Quelle zugegriffen werden. Mit der Data Warehouse-Zentrale können Sie Metadaten aus einer MQSeries-Nachrichtenwarteschlange und aus einer DB2-XML Extender-Dokumentzugriffsdefinition-Datei importieren.

Erstellen von Sichten für MQSeries-Nachrichten

Voraussetzungen

DB2 Universal Database Version 7.2

DB2 Warehouse Manager Version 7.2

MQSeries-Unterstützung. Weitere Informationen zu MQSeries-Voraussetzungen finden Sie unter „MQSeries“ auf Seite 191. Schlagen Sie im Konfigurationsabschnitt für benutzerdefinierte Funktionen nach, um Informationen zum Einrichten der Warehouse-Quelle zu erhalten.

Einschränkungen

- Wenn Sie eine Warehouse-Quellendatenbank katalogisieren, wird der Aliasname der Datenbank auf der Agentenmaschine katalogisiert. Wenn Sie allerdings MQSeries- und XML-Sichten erstellen, nimmt die Data Warehouse-Zentrale an, dass der Aliasname der Datenbank auch auf der Client-Maschine definiert ist und versucht, die Verbindung zur Client-Maschine herzustellen. Dabei werden die Benutzer-ID und das Kennwort der Warehouse-Quellendatenbank verwendet. Wenn das funktioniert, wird der Assistent aufgerufen, und Sie können die Sicht erstellen. Wenn die Verbindung fehlschlägt, wird eine Warnung angezeigt, und Sie müssen einen anderen Aliasnamen der Datenbank im Assistenten katalogisieren oder auswählen.
- Schlagen Sie die maximale Länge der MQ-Nachrichten im Abschnitt "SQL Reference" in den Release-Informationen nach.

Erstellen einer Sicht für MQSeries-Nachrichten

Gehen Sie wie folgt vor, um eine Sicht für MQSeries-Nachrichten zu erstellen:

1. Erweitern Sie die Baumstruktur **Warehouse-Quellen** im Fenster der Data Warehouse-Zentrale.
2. Erweitern Sie die Warehouse-Quelle, die die Sicht enthalten soll.
3. Klicken Sie den Ordner **Sichten** mit Maustaste 2 an, und klicken Sie **Für MQSeries-Nachrichten erstellen...** an.

Der MQSeries-Assistent wird geöffnet. Wenn Sie den Assistenten beendet haben, wird eine neue Sicht in der Data Warehouse-Zentrale erstellt. Wenn die Sicht ausgewählt wird, wird auf die MQSeries-Warteschlange zugegriffen und jede Nachricht wird anhand Ihrer Spezifikationen im Assistenten als begrenzte Zeichenfolge syntaktisch analysiert.

Importieren von MQSeries-Nachrichten und XML-Metadaten

Voraussetzungen

DB2 Universal Database Version 7.2

DB2 XML Extender Version 7.2

MQSeries-Unterstützung. Weitere Informationen zu MQSeries-Voraussetzungen finden Sie unter „MQSeries“ auf Seite 191. Schlagen Sie im Konfigurationsabschnitt für benutzerdefinierte Funktionen nach, um Informationen zum Einrichten der Warehouse-Quelle zu erhalten.

Einschränkungen

Der Import schlägt fehl, wenn die Zieltabelle mit Primär- oder Fremdschlüsseln vorhanden ist. Sie müssen diese Schlüssel vor dem Importieren in der Data Warehouse-Zentrale manuell löschen.

Importieren von MQSeries-Nachrichten und XML-Metadaten

Gehen Sie wie folgt vor, um MQSeries-Metadaten in die Data Warehouse-Zentrale zu importieren:

1. Bereiten Sie die Warehouse-Zieltabelle vor:
 - Sie müssen das Warehouse-Ziel definieren und Umsetzungsprogramme registrieren und aktivieren.

- Sie müssen das Warehouse-Ziel für DB2 XML Extender aktivieren. Weitere Informationen finden Sie in den Release-Informationen zu DB2 XML Extender Version 7.2.
 - Erstellen Sie eine XML Extender-Datenzugriffsdefinitionsdatei (DAD), um der Data Warehouse-Zentrale mitzuteilen, wie die Inhalte des XML-Dokuments auf Warehouse-Tabellen abgebildet werden sollen. Aktivieren Sie eine XML-Datensammlung mit der DAD-Datei für die Datenbank. Weitere Informationen finden Sie in den Release-Informationen zu DB2 XML Extender Version 7.2.
2. Klicken Sie **Warehouse** mit Maustaste 2 an, und klicken Sie **Metadaten importieren** —> **MQSeries** an, um das Fenster „Trillium Batch System“ zu öffnen.
 3. Geben Sie den Servicepunkt, an den eine Nachricht gesendet oder von dem eine Nachricht abgerufen wird, in das Feld **AMI-Service** ein.
 4. Geben Sie die Richtlinie, die das Nachrichtensystem verwenden wird, um die Operation auszuführen, in das Feld **AMI-Richtlinie** ein.
 5. Geben Sie den Namen der DB2-XML Extender-DAD-Datei in das Feld **DAD-Datei** ein, oder suchen Sie nach einer auswählbaren Datei, indem Sie die Ellipse (...) anklicken. Diese Datei muss lokal sein.
 6. Wählen Sie aus der kombinierten Liste des Felds **Warehouse-Ziel** den Namen des Warehouse-Ziels aus, auf dem der Schritt ausgeführt wird. Das Warehouse-Ziel muss bereits definiert sein.
 7. Geben Sie den Namen eines Schemas in das Feld **Schema** ein, um die Tabellennamen in der DAD-Datei zu qualifizieren, die keinen Qualifizierer haben. Die Anmelde-Benutzer-ID des Warehouse-Ziels, mit dem Sie zuletzt verbunden waren, wird als Standardwert verwendet.
 8. Wählen Sie eine **Zieloption** aus:
 - Wenn Sie wollen, dass der Schritt die Inhalte der Ziel-tabelle zur Laufzeit ersetzt, klicken Sie den Radioknopf **Tabelleninhalt ersetzen** an.
 - Wenn Sie wollen, dass der Schritt zur Laufzeit Daten an die Inhalte der Zieltabelle anhängt, klicken Sie den Radioknopf **An Tabelleninhalt anhängen** an.
 9. Klicken Sie **OK** an.
Das Fenster "Metadaten importieren" wird geschlossen.

Die folgenden Warehouse-Objekte werden der Warehouse-Baumstruktur hinzugefügt, wenn die Importoperation beendet ist.

- Ein Themenbereich mit dem Namen MQSeries und XML
- Ein Prozess mit dem Namen MQSeries und XML
- Eine benutzerdefinierte Programmgruppe mit dem Namen MQSeries und XML
- Definitionen aller Warehouse-Zieltabellen, die in der DAD-Datei beschrieben werden
- Der Schritt <servicename>.<dad-basisdateiname>.<warehouse-zielname>
- Die Programmschablone <servicename>.<dad-basisdateiname>

Wenn die Warehouse-Zielagenten-Site nicht die lokale Maschine ist, müssen Sie die Schrittparameter ändern:

1. Klicken Sie den Schritt mit Maustaste 2 an, und wählen Sie **Merkmale** aus. Klicken Sie im Notizbuch „Merkmale“ die Indexzunge **Parameter** an.
2. Ändern Sie den Namen des Parameters „DAD-Datei“ in den Namen derselben DAD-Datei auf der fernen Warehouse-Zielagenten-Site.

3. Stellen Sie sicher, dass die Agenten-Site in der Indexzunge „Verarbeitungsoptionen“ die gewünschte Agenten-Site enthält.

Verwenden von benutzerdefinierten Programmen für MQSeries

Die gespeicherte Prozedur für MQSeries und XML heißt **MQXMLXF** und ist in der DB2-Data Warehouse-Zentrale Version 7.2 für Windows NT und UNIX enthalten. Der Schritt, der erstellt wird, wenn Sie MQSeries- und XML-Metadaten importieren, führt die gespeicherte Prozedur aus. Seine Parameter werden in der folgenden Tabelle beschrieben:

Parameter	Werte
MQSeries ServiceName	Der Name des Servicepunkts, an den eine Nachricht gesendet wird oder von dem sie abgerufen wird
MQSeries PolicyName	Der Name der Richtlinie, den das Nachrichtenübertragungssystem verwenden wird, um die Operation auszuführen
DAD-Dateiname	Der Name der DAD-Datei von DB2 XML Extender
TargetTableList	Liste der Zieltabellen des Schritts, durch Kommas getrennt
Option	REPLACE oder APPEND
RUN ID	Schritteditionsnummer (für Protokollzwecke)
Anmerkung: Der Datentyp für alle Parameter in dieser Tabelle ist CHARACTER.	

Die gespeicherte Prozedur löscht alle Zeilen aus den Zieltabellen, wenn **Option** den Wert REPLACE hat. Die gespeicherte Prozedur ruft außerdem die gespeicherte DB2 XML Extender-Prozedur auf, um die Zieltabellen für alle vorhandenen MQSeries-Nachrichten zu füllen.

Fehlercode

Wenn der Schritt ausgeführt wird, gibt die gespeicherte Prozedur möglicherweise die Fehlercodes SQLCODE -443 und SQLSTATE 38600 zurück. In der folgenden Tabelle finden Sie mögliche Diagnosetexte, um den Fehler zu bestimmen:

Fehlernummer	Beschreibung
AMIRC=xxxx;<protokolldateiname>	xxxx ist der Rückkehrcode der AMI-Ebene. Weitere Einzelangaben finden Sie in der Dokumentation von MQSeries. <protokolldateiname> gibt die Adresse der Protokolldatei an.
XMLRC=xxxx;<protokolldateiname>	xxxx ist der Rückkehrcode von DB2 XML Extender. Eine Beschreibung der Rückkehrcodes finden Sie in der Dokumentation von DB2 XML Extender. <protokolldateiname> gibt die Adresse der Protokolldatei an.
SQLCODE=xxxx;<protokolldateiname>	xxxx ist der SQLCODE-Wert (nicht Null), der zurückgegeben wird, wenn eine SQL-Anfrage ausgeführt wird. <protokolldateiname> gibt die Adresse der Protokolldatei an.

In der Protokolldatei finden Sie weitere Informationen zu allen Fehlern.

Fehlerprotokolldatei

Die Data Warehouse-Zentrale speichert alle Diagnoseinformationen in einer Protokolldatei, wenn MQXMLXF ausgeführt wird. Der Name der Protokolldatei ist mqxf<nnnnnnnn>.log. Dabei ist <nnnnnnnn> die Ausführungs-ID, die an die gespeicherte Prozedur übermittelt wurde. Die Data Warehouse-Zentrale erstellt die Datei in dem Verzeichnis, das durch die Umgebungsvariable VWS_LOGGING angegeben wird. Wenn diese Umgebung nicht definiert ist, wird die Protokolldatei im temporären Verzeichnis erstellt.

Sie sollten VWS_LOGGING zur Umgebungsvariable DB2ENVLIST hinzufügen, indem Sie den Befehl **db2set** vor dem Befehl **db2start** ausführen, um die Umgebungsvariable VWS_LOGGING auf einer Unix-Plattform für die gespeicherte Prozedur sichtbar zu machen. Die folgende Abbildung zeigt einen Beispielumgebungsbehehl:

```
db2set DB2ENVLIST="AMT_DATA_PATH VWS_LOGGING"
```

Abbildung 6. Befehlsbeispiel für die Umgebungsvariable

Die Protokolldatei wird gelöscht, wenn der Schritt erfolgreich ausgeführt wird.

Unterstützung von Microsoft OLE DB und Data Transaction Services

Die Data Warehouse-Zentrale ermöglicht Ihnen jetzt den Zugriff auf Daten eines OLE DB-Anbieters als DB2-Datenbanksicht. Sie können mit dem in der Data Warehouse-Zentrale enthaltenen Assistenten zur Unterstützung von OLE DB verwenden eine DB2-OLE DB-Tabellenfunktion und die DB2-Sicht erstellen, über die Sie auf die Daten zugreifen können.

Die Datenkonvertierungsservices (Data Transformation Services, DTS) von Microsoft ermöglichen es Ihnen, Daten zwischen OLE DB-Quellen und -Zielen zu importieren, zu exportieren und zu konvertieren, um Data Warehouses und Datamarts aufzubauen. DTS wird mit dem Microsoft SQL Server installiert. Alle DTS-Tasks sind in DTS-Paketen gespeichert, die Sie mit Microsoft OLE DB Provider for DTS Packages ausführen können und auf die Sie mit dem Programm zugreifen können. Weil Sie auf Pakete aus DTS als OLE DB-Quellen zugreifen können, können Sie mit dem Assistenten zur Unterstützung von OLE DB in derselben Weise auch Sichten für DTS-Pakete erstellen wie für OLE DB-Datenquellen. Wenn Sie während der Laufzeit auf die Sicht zugreifen, wird das DTS-Paket ausgeführt und die Zieltabelle der Task in dem DTS-Paket wird die erstellte sicht.

Nachdem Sie eine Sicht in der Data Warehouse-Zentrale erstellt haben, können Sie sie wie jede andere Sicht verwenden. Sie können z. B. eine DB2-Tabelle mit einer OLE DB-Quelle in einem SQL-Schritt verknüpfen. Wenn Sie die erstellte Sicht in einem SQL-Schritt verwenden, wird der DTS-Anbieter aufgerufen und das DTS-Paket ausgeführt.

Softwarevoraussetzungen:

- DB2 Universal Database für Windows NT Version 7.2 als Warehouse-Zieldatenbank
- DB2 Warehouse Manager Version 7.2
- Wenn die Warehouse-Zieldatenbank vor Version 7.2 erstellt wurde, müssen Sie den Befehl db2updv7 ausführen, nachdem Sie DB2 UDB für Windows NT Version 7.2 installiert haben.

- Wenn Sie eine Warehouse-Quellendatenbank katalogisieren, wird der Aliasname der Datenbank auf der Warehouse-Agenten-Site katalogisiert. Wenn Sie allerdings den Assistenten starten, nimmt die Data Warehouse-Zentrale an, dass der Aliasname der Datenbank auch auf der Client-Workstation definiert ist und versucht, die Verbindung zur Client-Workstation herzustellen. Dabei werden die Benutzer-ID und das Kennwort der Warehouse-Quellendatenbank verwendet. Wenn die Verbindung erfolgreich ist, startet der Assistent, und Sie können die Sicht erstellen. Wenn die Verbindung fehlschlägt, wird eine Warnung angezeigt, und Sie müssen einen anderen Aliasnamen der Datenbank im Assistenten katalogisieren oder auswählen.
- Sie müssen das Markierungsfeld **DSO rowset provider** auf der Indexzunge „Options“ im Fenster „Workflow Properties“ der DataPumpTask auswählen, die die Zieltabelle erstellt. Wenn Sie mehrere DSO-Zeilengruppenanbieterattribute einschalten, wird nur das Ergebnis des ersten ausgewählten Schritts verwendet. Wenn eine Sicht ausgewählt wird, wird die Zeilengruppe ihrer Zieltabelle zurückgegeben, und alle anderen Zeilengruppen, die Sie in späteren Schritten erstellen, werden ignoriert.
- Wenn Sie den Tabellennamen für den Assistenten eingeben, verwenden Sie den Schrittnamen, die auf der Seite „Options“ im Notizbuch „Workflow Properties“ für die Task angezeigt wird.
- Die DTS-Paketverbindungszeichenfolge hat dieselbe Syntax wie der Befehl `dtstrun`.

Erstellen von Sichten für OLE DB-Tabellenfunktionen

Gehen Sie wie folgt vor, um eine Sicht für eine OLE DB-Tabellenfunktion zu erstellen:

1. Erweitern Sie die Baumstruktur **Warehouse-Quellen** im Fenster der Data Warehouse-Zentrale.
2. Erweitern Sie die Warehouse-Quelle, die die Sicht enthalten soll.
3. Klicken Sie den Ordner **Sichten** mit Maustaste 2 an, und klicken Sie **Für OLE DB-Tabellenfunktion erstellen...** an.

Der OLE DB-Assistent wird geöffnet. Der Assistent führt Sie durch die Task zum Erstellen einer neuen Sicht in der Warehouse-Quellendatenbank.

Erstellen von Sichten für DTS-Pakete

Gehen Sie wie folgt vor, um eine Sicht für ein DTS-Paket zu erstellen:

1. Erweitern Sie die Baumstruktur **Warehouse-Quellen** im Fenster der Data Warehouse-Zentrale.
2. Erweitern Sie die Warehouse-Quelle, die die Sicht enthalten soll.
3. Klicken Sie den Ordner **Sichten** mit Maustaste 2 an, und klicken Sie **Microsoft OLE DB Provider for DTS Packages** an.

Der Assistent zur Unterstützung von OLE DB wird geöffnet. Der Assistent führt Sie durch die Task zum Erstellen einer neuen Sicht in der Warehouse-Quellendatenbank.

Weitere Informationen finden Sie in der Dokumentation von Microsoft Platform SDK 2000, die eine genaue Beschreibung enthält, wie Sie die Anbieterzeichenfolge erstellen, die der Assistent benötigt, um sich mit dem DTS-Anbieter zu verbinden.

Verwenden von inkrementellen COMMIT-Operationen mit Ersetzen

In einem Schritt, in dem der Mengentyp „Ersetzen“ ist, wird eine inkrementelle COMMIT-Operation nur dann verwendet, wenn die neuen Daten eingefügt werden. Die alten Daten werden in einem einzigen COMMIT-Operationsbereich gelöscht. Wenn Sie die Daten löschen müssen, ohne Protokollsätze zu erzeugen, führen Sie einen Schritt aus, der eine leere Datei lädt, bevor Sie den SQL-Schritt mit Fülltyp „Anhängen“ ausführen.

Komponenten-Trace-Datendateinamen

Die Data Warehouse-Zentrale schreibt unter Windows NT die folgenden Dateien: **AGNT $nnnn$.Log** enthält Trace-Informationen. $nnnn$ ist die numerische Prozess-ID des Warehouse-Agenten, die je nach Betriebssystem 4 oder 5 Ziffern enthalten kann.

AGNT $nnnn$.Set enthält Umgebungseinstellungen für den Agenten. $nnnn$ ist die numerische Prozess-ID des Warehouse-Agenten, die je nach Betriebssystem 4 oder 5 Ziffern enthalten kann.

Das Standardverzeichnis ist `x:\program files\sqllib\logging`. Dabei ist x das Laufwerk, auf dem DB2 installiert ist.

Open Client für Sybase-Quellen unter AIX und der Solaris-Betriebsumgebung erforderlich

In „Kapitel 3. Definieren von Warehouse-Quellen“ sollte der Eintrag für Sybase für AIX in Tabelle 3, „Konnektivitätsanforderungen für unterstützte Datenquellen unter AIX“, und für die Solaris-Betriebsumgebung in Tabelle 4, „Konnektivitätsanforderungen für unterstützte Datenquellen in der Solaris-Betriebsumgebung“, einen zusätzlichen Schritt in der Spalte „Erforderliche Schritte zur Verbindungsherstellung:“ enthalten. Der zusätzliche Schritt wird unten als Schritt 3 angezeigt.

3. Installieren Sie Open Client

Beachten Sie, dass Open Client erforderlich ist, um eine Verbindung zu Sybase-Quellen auf Windows NT- oder Windows 2000-Plattformen herzustellen.

Korrigierte Beispieleinträge

Abbildungen 6, 8, 10 und 11 in „Kapitel 3. Definieren von Warehouse-Quellen“ im Handbuch *Data Warehouse-Zentrale Verwaltung* enthalten einen fehlerhaften Pfad für das Attribut „Driver“. Die folgenden Pfade sind korrekt:

Abbildung 6

Driver=/home/db2_07_01/3.6/odbc/lib/ivinf12.so

Abbildung 8

Driver=/home/db2_07_01/3.6/odbc/lib/ivsyb1112.so

Abbildung 10

Driver=/home/db2_07_01/3.6/lib/ivor814.so

Abbildung 11

Driver=/home/db2_07_01/3.6/odbc/lib/ivmsss14.so

Kapitel 3. Definieren von Warehouse-Quellen

Zuordnen des Memofelds in Microsoft Access zu einer Warehouse-Quelle

Das Memofeld einer Microsoft Access-Datenbank wird in einer Quelle der Data Warehouse-Zentrale als Datentyp LONG VARCHAR mit einer Spaltengröße von mehr als 1 GB dargestellt. Zur Unterstützung praktischer Systemkonfigurationen schneidet die Data Warehouse-Zentrale die Werte über 128 KB ab. Wenn Sie vermeiden wollen, dass Werte des Memofelds in der Warehouse-Quelle abgeschnitten werden, ändern Sie den Datentyp der Spalte, die die Memofelddaten empfängt, von LONG VARCHAR in CLOB, bevor Sie die Tabelle in einem Schritt verwenden. Wenn Sie dies nicht tun, werden Werte über 128 KB abgeschnitten. DRDA-Unterstützung für den Datentyp CLOB ist für OS/390 und OS/400 erforderlich. Der Datentyp CLOB wird für OS/390 ab DB2 Version 6 unterstützt. Für OS/400 wird der Datentyp CLOB ab Version 4, Release 4 mit DB FixPak 4 oder höher (PTF SF99104) unterstützt. Bei OS/400 enthält der Installationsdatenträger der Version 4, Release 4 vom Februar 1999 auch Unterstützung für den Datentyp CLOB.

Kapitel 10. Verwalten der Warehouse-Datenbank

Verknüpfen von Tabellen mit einem Schrittsubtyp für das Programm DB2 UDB RUNSTATS

Der Schrittsubtyp für ein RUNSTATS-Programm liest von einem Warehouse-Ziel und schreibt darauf. Verknüpfen Sie ein Ziel mit dem Schrittsubtyp im Fenster **Prozessmodell**, bevor Sie die Werte für den Schritt definieren.

Standard-Warehouse-Steuerungsdatenbank

Bei einer Standardinstallation von DB2 unter Windows NT oder Windows 2000 erstellt und initialisiert DB2 eine Standard-Warehouse-Steuerungsdatenbank für die Data Warehouse-Zentrale, sofern in der Registrierdatenbank von Windows NT keine aktive Warehouse-Steuerungsdatenbank angegeben wird. *Initialisierung* bezeichnet den Prozess, bei dem die Data Warehouse-Zentrale die Steuertabellen erstellt, die zum Speichern der Metadaten der Data Warehouse-Zentrale erforderlich sind. Der Name der Standard-Warehouse-Steuerungsdatenbank ist DWC-TRLDB. Beim Anmelden gibt die Data Warehouse-Zentrale standardmäßig DWC-TRLDB als Warehouse-Steuerungsdatenbank an. Wenn Sie den Namen der Warehouse-Steuerungsdatenbank anzeigen wollen, die zur Verwendung vorgesehen ist, klicken Sie den Knopf "Erweitert" im Fenster "Data Warehouse-Zentrale - Anmeldung" an.

Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung"

Das Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung" wird bei einer Standardinstallation von DB2 unter Windows NT oder Windows 2000 installiert. In diesem Fenster können Sie die aktive Warehouse-Steuerungsdatenbank ändern, neue Warehouse-Steuerungsdatenbanken erstellen und initialisieren und Warehouse-Steuerungsdatenbanken migrieren, die mit IBM Visual Warehouse verwendet wurden. Diese Aktivitäten werden in den folgenden Abschnitten erläutert. Stoppen Sie den Warehouse-Server, bevor Sie das Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung" verwenden.

Ändern der aktiven Warehouse-Steuerungsdatenbank

Wenn Sie eine andere als die aktive Warehouse-Steuerungsdatenbank verwenden möchten, registrieren Sie im Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung" die gewünschte Datenbank als aktive Steuerungsdatenbank. Wenn Sie beim Anmelden an die Data Warehouse-Zentrale einen anderen Namen als den der aktiven Warehouse-Steuerungsdatenbank angeben, erhalten Sie eine Fehlermeldung mit dem Inhalt, dass die von Ihnen angegebene Datenbank nicht mit der vom Warehouse-Server angegebenen Datenbank übereinstimmt.

Gehen Sie wie folgt vor, um die Datenbank zu registrieren:

1. Klicken Sie **Start** → **Programme** → **IBM DB2** → **Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung** an.
2. Geben Sie in das Feld **Neue Steuerungsdatenbank** den Namen der Steuerungsdatenbank ein, die Sie verwenden wollen.
3. Geben Sie in das Feld **Schema** den Namen des gewünschten Schemas für die Datenbank ein.
4. Geben Sie in das Feld **Benutzer-ID** den Namen der für den Zugriff auf die Datenbank erforderlichen Benutzer-ID ein.
5. Geben Sie in das Feld **Kennwort** den Namen des Kennworts für die Benutzer-ID ein.
6. Geben Sie in das Feld **Prüfkennwort** das Kennwort noch einmal ein.
7. Klicken Sie **OK** an.
Das Fenster bleibt offen. Im Feld **Nachrichten** werden Statusnachrichten zum Registrierungsprozess angezeigt.
8. Wenn der Prozess abgeschlossen ist, schließen Sie das Fenster.

Erstellen und Initialisieren einer Warehouse-Steuerungsdatenbank

Wenn Sie eine andere als die Standard-Warehouse-Steuerungsdatenbank erstellen wollen, können Sie dies während oder nach der Installation im Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung" tun. Sie können eine Datenbank bei der Installation auf derselben Workstation wie der Warehouse-Server oder auf einer anderen Workstation erstellen.

Zum Ändern des Namens der bei der Installation erstellten Warehouse-Steuerungsdatenbank müssen Sie eine angepasste Installation durchführen und den Namen im Fenster "Lokale Warehouse-Steuerungsdatenbank definieren" ändern. Der Installationsprozess erstellt die Datenbank mit dem angegebenen Namen, initialisiert sie zur Verwendung mit der Data Warehouse-Zentrale und registriert sie als aktive Warehouse-Steuerungsdatenbank.

Wenn Sie bei der Installation eine Warehouse-Steuerungsdatenbank auf einer anderen Workstation erstellen wollen als auf derjenigen, auf welcher der Warehouse-Server installiert ist, wählen Sie während einer angepassten Installation "Lokale Warehouse-Steuerungsdatenbank" aus. Die Datenbank wird durch den Installationsprozess erstellt. Nach der Installation müssen Sie das Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung" auf der Workstation mit dem Warehouse-Server verwenden, indem Sie die Schritte in „Ändern der aktiven Warehouse-Steuerungsdatenbank“, ausführen. Geben Sie den Datenbanknamen an, den Sie bei der Installation angegeben haben. Die Datenbank wird zur Verwendung mit der Data Warehouse-Zentrale initialisiert und als aktive Warehouse-Steuerungsdatenbank registriert.

Wenn Sie nach der Installation eine Warehouse-Steuerungsdatenbank erstellen und initialisieren möchten, können Sie dies im Fenster "Data Warehouse-Zentrale - Steuerungsdatenbankverwaltung" auf der Warehouse-Server-Workstation tun. Befindet sich die neue Warehouse-Steuerungsdatenbank nicht auf der Warehouse-Server-Workstation, müssen Sie die Datenbank zunächst erstellen und dann auf der Warehouse-Server-Workstation katalogisieren. Führen Sie dann die in „Ändern der aktiven Warehouse-Steuerungsdatenbank“ auf Seite 259, beschriebenen Schritte aus. Geben Sie den Datenbanknamen an, den Sie bei der Installation angegeben haben.

Wenn Sie sich an der Data Warehouse-Zentrale anmelden, klicken Sie den Knopf **Erweitert** an, und geben Sie den Namen der aktiven Warehouse-Steuerungsdatenbank ein.

Erstellen von Momentaufnahmen für SQL-Schritte

Beim Erstellen von Momentaufnahmen für SQL-Schritte auf Basis der Verwendung sollten Sie die Erstellung eines nicht eindeutigen Index in der Editionsspalte in Erwägung ziehen, um die Leistung beim Löschen von Editionen zu verbessern. Diese Überlegung ist nur für große Warehouse-Tabellen von Vorteil, da die Einfügeleistung beeinträchtigt werden kann, wenn eine geringe Anzahl an Zeilen eingefügt wird.

Ändern von Quellen und Zielen im Fenster "Prozessmodell"

Wenn Sie im Fenster "Prozessmodell" eine Quelle oder eine Ziel ändern, wird die Änderung sofort automatisch gespeichert. Jede andere Änderung, beispielsweise das Hinzufügen eines Schritts, muss ausdrücklich gespeichert werden, damit die Änderung von Dauer ist. Klicken Sie zum Speichern der Änderung "Prozess" → "Speichern" an.

Hinzufügen von Beschreibungen zu Objekten der Data Warehouse-Zentrale

Sie können im Beschreibungsfeld von Notizbüchern der Data Warehouse-Zentrale bis zu 254 Zeichen angeben. Dieser Maximalwert ersetzt die in der Onlinehilfefunktion angegebenen maximalen Längen.

Ausführen von Beispielinhalt

- Sie können eine Anforderung von Beispielinhalt für eine Flachdateiquelle mit dem AS/400-Agenten nicht erfolgreich ausführen. Sie können zwar eine Flachdateiquelle erstellen und versuchen, mit dem AS/400-Agenten eine Anforderung von Beispielinhalt abzusetzen, doch diese Anforderung schlägt fehl.
- Möglicherweise erhalten Sie eine Fehlermeldung, wenn Sie den Beispielinhalt auf einem Warehouse-Ziel im Prozesserstellungsprogramm ausführen. Dieser Fehler hängt mit der Verfügbarkeit einer allgemeinen Agenten-Site für eine Warehouse-Quelle, ein Warehouse-Ziel und einen Schritt in einem Prozess zusammen. Die Liste des verfügbaren Agenten-Sites für einen Schritt ergibt sich aus der Schnittmenge der Agenten-Sites der Warehouse-Quelle, der Agenten-Sites des Warehouse-Ziels und der für den entsprechenden Schritt verfügbaren Agenten-Sites (die Schritte werden auf der letzten Seite des Notizbuchs für die Merkmale der Agenten-Sites ausgewählt). Sie wollen beispielsweise den Beispielinhalt für einen Prozess anzeigen, der das FTP-Kopierprogramm (VWPRCPY) ausführt. Der in dem Prozess verwendete Schritt muss bei der Agenten-Site-Definition für die Agenten-Site ausgewählt werden. Wenn Sie den Beispielinhalt

für die Zieldatei ausführen, wird normalerweise die erste Agenten-Site in der ausgewählten Liste verwendet. Die Reihenfolge der Agenten-Sites kann jedoch durch Arbeitsgänge im Zuge der Datenbankpflege verändert werden. Der Beispieldinhalt schlägt fehl, wenn sich die ausgewählte Agenten-Site nicht auf demselben System befindet wie die Quellen- oder Zieldatei.

Bearbeiten einer SQL-Anweisung zur DDL-Erstellung

Wenn Sie versuchen, die SQL-Anweisung zur DDL-Erstellung für eine Zieltabelle oder einen Schritt im Entwicklungsmodus zu editieren, wird folgende irreführende Nachricht angezeigt: „Änderungen an der SQL-Anweisung zur DDL-Erstellung werden nicht in der Tabellendefinition oder der tatsächlichen physischen Tabelle widergespiegelt. Möchten Sie fortfahren?“

In Wirklichkeit wird die Änderung in der physischen Tabelle widergespiegelt. Ignorieren Sie diese Nachricht und fahren Sie mit der Änderung der Anweisung zur DDL-Erstellung fort.

Die korrekte Version dieser Nachricht lautet für Schritte im Entwicklungsmodus wie folgt: „Änderungen an der SQL-Anweisung zur DDL-Erstellung werden nicht in der Tabellendefinition widergespiegelt. Möchten Sie fortfahren?“

Für Schritte im Test- oder Produktionsmodus ist die Nachricht korrekt. Die Data Warehouse-Zentrale ändert die physische Zieltabelle nicht, die beim Hochstufen des Schritts in den Testmodus erstellt wurde.

Migrieren von Geschäftssichten von Visual Warehouse

Wenn Sie Geschäftssichten für Metadatensynchronisation von Visual Warehouse auf die entsprechenden Sichten der Data Warehouse-Zentrale migrieren wollen, stufen Sie vor der Migration der Warehouse-Steuerungsdatenbank die Geschäftssichten in den Produktionsstatus hoch. Sind die Geschäftssichten im Produktionsstatus, werden ihre Zeitpläne in die Data Warehouse-Zentrale migriert. Sind die Geschäftssichten nicht im Produktionsstatus, werden Sie im Teststatus ohne ihre Zeitpläne migriert. Sie können die migrierten Schritte nicht in den Produktionsstatus hochstufen. Sie müssen die Synchronisationsschritte in der Data Warehouse-Zentrale neu erstellen und die migrierten Schritte löschen.

Generieren von Zieltabellen und Primärschlüsseln

Beim Generieren der Zieltabelle für einen Schritt generiert die Data Warehouse-Zentrale keinen Primärschlüssel für die Zieltabelle. Einige Umsetzungsprogramme, darunter "Gleitender Durchschnitt", verwenden die generierte Tabelle als Quellentabelle und benötigen außerdem einen Primärschlüssel in der Quellentabelle. Vor der Verwendung der generierten Tabelle mit dem Umsetzungsprogramm müssen Sie den Primärschlüssel für die Tabelle definieren, indem Sie die Tabelle in der DB2-Steuerzentrale mit Maustaste 2 anklicken und **Ändern** auswählen.

Verwenden von Merant ODBC-Treibern

Stellen Sie für den Zugriff auf Microsoft SQL Server unter Windows NT mit den Merant ODBC-Treibern sicher, dass das Verzeichnis `sqllib\odbc32` im Systempfad enthalten ist.

Neuer ODBC-Treiber

Wenn Sie den AIX- oder Sun-Agenten der Data Warehouse-Zentrale nutzen wollen, der für den Zugriff auf Merant ODBC-Quellen verknüpft wurde, und auch auf DB2-Datenbanken zugreifen wollen, ändern Sie den Wert des Attributs "Driver=" im Abschnitt zur DB2-Quelle der Datei `.odbc.ini` wie folgt:

AIX: Der Treibername lautet „`/usr/lpp/db2_07_01/lib/db2_36.o`“.

Beispiel für einen ODBC-Quelleneintrag für AIX:

```
[SAMPLE] Driver=/usr/lpp/db2_07_01/lib/db2_36.o
Description=DB2 ODBC Database
Database=SAMPLE
```

Sun: Der Treibername lautet „`/opt/IBMd2/V7.1/lib/libdb2_36.so`“

Beispiel für einen ODBC-Quelleneintrag für Sun:

```
[SAMPLE] Driver=/opt/IBMd2/V7.1/lib/libdb2_36.so
Description=DB2 ODBC Database
Database=SAMPLE
```

Definieren einer Warehouse-Quelle oder eines Warehouse-Ziels in einer OS/2-Datenbank

Wenn Sie eine Warehouse-Quelle oder ein Warehouse-Ziel für eine OS/2-Datenbank definieren, geben Sie den Datenbanknamen in Großbuchstaben ein.

Überwachen des Status der Warehouse-Steuerungsdatenbank

Die DB2-Steuerzentrale oder der Befehlszeilenprozessor zeigt möglicherweise an, dass sich die Warehouse-Steuerungsdatenbank in einem inkonsistenten Status befindet. Dieser Status wird erwartet, denn er weist darauf hin, dass der Warehouse-Server seine ersten Nachrichten beim Systemstart nicht in der Warehouse-Protokollfunktion festgeschrieben hat.

Verwenden von SQL Assist mit der Beispieldatenbank TBC_MD

In dem Beispiel zu Data Warehouse in der Datenbank TBC_MD können Sie das SQL im Schritt **Select Scenario SQL** nicht mit SQL Assist ändern, weil das SQL nach dem Generieren durch SQL Assist editiert wurde.

Verwenden der Funktion FormatDate

Klicken Sie **SQL erstellen** auf der Seite **SQL-Anweisung** des Notizbuchs **Merkmale** für einen SQL-Schritt an, um die Funktion `FormatDate` zu verwenden.

Die Ausgabe von `FormatDate` hat den Datentyp `varchar(255)`. Sie können den Datentyp nicht durch die Auswahl von **Datum**, **Uhrzeit** oder **Datum/Uhrzeit** aus der Liste **Kategorie** auf der Seite **Funktionsparameter - FormatDate** ändern.

Ändern der Spracheinstellung

In AIX- und Solaris-Betriebsumgebungen legt der Installationsprozess die Sprache zur Veröffentlichung im Informationskatalog sowie zum Export nach OLAP Integration Server fest. Wenn Sie diese Funktionen in einer anderen als der bei der Installation gesetzten Sprache verwenden wollen, erstellen Sie folgenden Softlink durch die Eingabe des folgenden Befehls in einer Zeile:

Unter AIX

```
/usr/bin/ln -sf /usr/lpp/db2_07_01/msg/locale/flgnxolv.str  
/usr/lpp/db2_07_01/bin/flgnxolv.str
```

ländereinstellung

Der Name der Landessprache im Format xx_yy

In der Solaris-Betriebsumgebung

```
/usr/bin/ln -sf /opt/IBMDB2/V7.1/msg/ländereinstellung/flgnxolv.str  
/opt/IBMDB2/V7.1/bin/flgnxolv.str
```

ländereinstellung

Der Name der Landessprache im Format xx_yy

Verwenden des Umsetzungsprogramms "Schlüsseltabelle generieren"

Wenn Sie die Option **Wert in der Schlüsseltabelle aktualisieren** des Umsetzungsprogramms "Schlüsseltabelle generieren" verwenden, aktualisiert das Umsetzungsprogramm nur diejenigen Zeilen in der Tabelle, die keine Schlüsselwerte haben. (Das heißt, die Werte sind Null). Wenn zusätzliche Zeilen in die Tabelle eingefügt werden, sind die Schlüsselwerte Null, bis Sie das Umsetzungsprogramm erneut ausführen.

Sie haben folgende Möglichkeit, dieses Problem zu vermeiden:

- Verwenden Sie nach der ersten Ausführung des Umsetzungsprogramms die Option **Alle Schlüsselspaltenwerte ersetzen**, um die Schlüssel für alle Zeilen neu zu erstellen.

Aufrechterhalten der Verbindungen zu Datenbanken

Der Warehouse-Server erhält keine Verbindungen zu lokalen oder fernen Datenbanken aufrecht, wenn der DB2-Server, welcher die Datenbanken verwaltet, gestoppt und neu gestartet wird. Wenn Sie DB2 stoppen und neu starten, stoppen und starten Sie die Warehouse-Services ebenfalls neu.

Definieren eines fernen Clients der Data Warehouse-Zentrale

Wenn Sie DB2 Administration Client und die Data Warehousing-Tools installieren, um den Verwaltungs-Client der Data Warehouse-Zentrale auf einer anderen Workstation als der mit dem Warehouse-Server einzurichten, müssen Sie die Nummer des TCP/IP-Anschlusses hinzufügen, den die Warehouse-Server-Workstation zu Empfang der Servicedatei für die Client-Workstation verwendet. Fügen Sie wie folgt eine Eintrag in die Servicedatei hinzu:

```
vwkernel      11000/tcp
```

Definieren einer Warehouse-Quelle von DB2 für VM

Wenn Sie eine Warehouse-Quelle für eine DB2 für VM-Datenbank definieren, auf die über einen DRDA-Gateway zugegriffen wird, gibt es Einschränkungen bei der Verwendung der Datentypen CLOB und BLOB:

- Sie können Daten der Typen CLOB und BLOB nicht mit der Beispielinhaltsfunktion anzeigen.
- Sie können Spalten der Datentypen CLOB und BLOB nicht bei einem SQL-Schritt verwenden.

Dies ist eine bekannte Einschränkung für DB2 für VM-Server der Version 5.2, bei der LOB-Objekte nicht mit DRDA an einen Client von DB2 Version 7 übertragen werden können.

Definieren einer Zieltabelle von DB2 für VM oder DB2 für VSE

Wenn Sie in der Data Warehouse-Zentrale eine DB2 für VM- oder DB2 für VSE-Zieltabelle definieren, wählen Sie das Markierungsfeld **An PUBLIC erteilen** nicht aus. Die von der Data Warehouse-Zentrale generierte Befehlssyntax von GRANT wird von DB2 für VM und DB2 für VSE nicht unterstützt.

Aktivieren der Unterstützung für begrenzte Bezeichner

Wählen Sie das Markierungsfeld **Enable Quoted Identifiers** auf der Seite **Advanced** des Notizbuchs **ODBC Driver Setup** aus, um Unterstützung für begrenzte Bezeichner für Sybase und Microsoft SQL Server unter NT zu aktivieren.

Zum Aktivieren von Unterstützung für begrenzte Bezeichner für Sybase unter UNIX editieren Sie die Sybase-Datenquelle in der Datei `.odbc.ini`, indem Sie das Verbindungsattribut `EQI=1` einfügen.

DataJoiner-Fehler weist auf ein Bindeproblem hin

Kunden, die DataJoiner in Verbindung mit DB2 Version 7.1 und FixPak 2 oder höher verwenden, erhalten möglicherweise einen Fehler, der auf ein Bindeproblem hinweist.

Wenn Sie z. B. eine DataJoiner-Quelle mit einem Agenten der Data Warehouse-Zentrale der Version 7 verwenden, kann ein Fehler wie der folgende auftreten:

```
DWC07356E Die Verarbeitung eines Befehls vom Typ "importTableNames" für die Edition "0"
des Schritts "?" durch einen Agenten ist fehlgeschlagen.
SQL0001N Binden oder Vorkompilieren nicht erfolgreich abgeschlossen.
SQL0001N Paket "NULLID.SQLL6D05" nicht gefunden.  SQLSTATE=51002 RC = 7356  RC2 = 8600
```

Fügen Sie der Datei `db2cli.ini` die folgenden Zeilen hinzu, um das Problem zu beheben:

```
[COMMON]
DYNAMIC=1
```

Bei UNIX-Systemen befindet sich die Datei `db2cli.ini` im Verzeichnis `.../sql1lib/cfg`. Bei Windows NT befindet sich die Datei `db2cli.ini` im Verzeichnis `.../sql1lib`.

Einrichten und Ausführen von Replikation mit der Data Warehouse-Zentrale

1. Zum Einrichten und Ausführen von Replikation mit der Data Warehouse-Zentrale müssen die Replikationssteuertabellen sowohl in der Warehouse-Steuerungsdatenbank als auch in den Warehouse-Zieldatenbanken vorhanden sein.

Eine Replikation erfordert das Vorhandensein der Replikationssteuertabellen in der Steuerdatenbank und in den Zieldatenbanken. Die Replikationssteuertabellen befinden sich im ASN-Schema und beginnen mit der Zeichenfolge IBMSNAP. Die Replikationssteuertabellen werden in einer Datenbank automatisch für Sie erstellt, wenn Sie mit der Steuerzentrale eine Replikationsquelle definieren, sofern die Steuertabellen nicht bereits vorhanden sind. Beachten Sie, dass die Steuertabellen auch auf der Zieldatenbank vorhanden sein müssen. Zum Erstellen einer Gruppe von Steuertabellen in der Zieldatenbank können Sie entweder mit der Steuerzentrale eine Replikationsquelle erstellen und diese anschließend entfernen, so dass die Steuertabellen übrig bleiben. Oder Sie verwenden das DJRA-Tool (DataJoiner Replication Administration), um nur die Steuertabellen zu definieren.

2. Installieren und Verwenden von DJRA

Wenn Sie zum Definieren der Steuertabellen DJRA verwenden wollen oder müssen, müssen Sie dieses Tool zunächst installieren. DJRA ist im Lieferumfang von DB2 enthalten. Wechseln Sie zum Installieren von DJRA in das Verzeichnis `d:\sql11ib\djra` (Ihr DB2-Installationsverzeichnis), und klicken Sie das Paket `djra.exe` an. DJRA wird nun auf Ihrem System installiert. Für den Zugriff auf DJRA nach der Installation klicken Sie unter Windows NT im Menü "Start" die Auswahl "DB2 für Windows NT" an, wählen "Replication" und anschließend "Replication Administration Tools" aus. Die DJRA-Schnittstelle unterscheidet sich leicht von normalen NT-Anwendungen. Sie erstellt für jede ihrer Funktionen eine Gruppe von auszuführenden SQL-Anweisungen, führt diese jedoch nicht aus. Das Benutzer muss die generierten SQL-Anweisungen manuell sichern und dann die Funktion zum Ausführen der SQL-Anweisung auswählen.

3. Konfigurieren für die Ausführung von Capture und Apply

Anleitungen zum Konfigurieren Ihres Testsystems zum Ausführen des Capture- bzw. Apply-Programms finden Sie in "Replikation Referenzhandbuch". Sie müssen das Apply- und das Capture-Programm in allen Datenbanken binden, in denen Sie verwendet werden sollen. Beachten Sie, dass Sie KEINE Kennwortdatei erstellen müssen. Die Data Warehouse-Zentrale erstellt automatisch eine Kennwortdatei für die Replikationssubskription.

4. Definieren einer Replikationsquelle in der Steuerzentrale

Verwenden Sie die Steuerzentrale, um eine Replikationsquelle zu definieren: Die Data Warehouse-Zentrale unterstützt fünf Replikationsarten: "Benutzerkopie", "Mit Zeitangabe", "Basisergebnistabelle", "CA-Tabelle" und "Zwischenspeichertabelle" (CCD-Tabellen). Bei den Replikationsarten "Benutzerkopie", "Mit Zeitangabe" und "Komprimierte Zwischenspeichertabelle" muss die Replikationsquellentabelle einen Primärschlüssel besitzen. Bei den übrigen Replikationsarten ist dies nicht erforderlich. Beachten Sie dies beim Auswählen einer Eingabetabelle, die als Replikationsquelle definiert werden soll. Eine Replikationsquelle ist eigentlich die Definition einer ursprünglichen Quellentabelle und einer erstellten CD-Tabelle, welche die Datenänderungen enthält, bis diese in die Zieltabelle versetzt werden. Beim Definieren einer Replikationsquelle in der Steuerzentrale wird ein Eintrag in die Tabelle

ASN.IBMSNAP_REGISTER geschrieben, um die Quelle und ihre CD-Tabelle zu definieren. Gleichzeitig wird die CD-Tabelle erstellt, enthält anfangs jedoch keine Daten. Wenn Sie eine Replikationsquelle definieren, können Sie auswählen, ob nur die Nach-Image-Spalten oder sowohl Vor- als auch Nach-Image-Spalten enthalten sein sollen. Diese Auswahlen werden über Markierungsfelder in der Replikationsquellenschnittstelle der Steuerzentrale getroffen. Ihre Auswahl der Vor- bzw. Nach-Image-Spalten wird dann in Spalten umgesetzt, die in der neuen CD-Tabelle erstellt werden. Die Namen der Nach-Image-Spalten in der CD-Tabelle sind mit denen der ursprünglichen Spalte in der Quellentabelle identisch. Auch Vor-Image-Spalten haben dieselben Namen, beginnen jedoch mit dem Zeichen 'X', um sicherzustellen, dass die Namen eindeutig sind.

5. Importieren der Replikationsquelle in die Data Warehouse-Zentrale

Sobald Sie die Replikationsquelle in der Steuerzentrale erstellt haben, können Sie sie in die Data Warehouse-Zentrale importieren. Achten Sie beim Importieren der Quelle darauf, dass das Markierungsfeld "Nur replizierbare Tabellen abrufen" aktiviert ist. Damit wird die Data Warehouse-Zentrale angewiesen, in den Einträgen der Tabelle ASN.IBMSNAP_REGISTER zu prüfen, welche Tabellen als Replikationsquellen definiert worden sind.

6. Definieren eines Replikationsschritts in der Data Warehouse-Zentrale

Wählen Sie im Prozesserstellungsprogramm eine der fünf Replikationsarten aus: "Basisergebnistabelle", "CA-Tabelle", "Mit Zeitangabe", "Zwischenspeichertabelle" oder "Benutzerkopie". Wenn Sie eine Replikationsart "Basisergebnistabelle" oder "CA-Tabelle" erstellen wollen, lesen Sie weiter unten den Abschnitt "Einrichten einer Replikation mit Basisergebnistabelle oder CD-Tabelle in der Data Warehouse-Zentrale". Wählen Sie eine geeignete Replikationsquelle für die Replikationsart aus. Wie bereits weiter oben erwähnt, muss bei den Replikationsarten "Benutzerkopie", "Mit Zeitangabe" und "Komprimierte Zwischenspeichertabelle" die Eingabequelle einen Primärschlüssel besitzen. Verbinden Sie die Replikationsquelle mit dem Replikationsschritt. Öffnen Sie die Merkmale für den Replikationsschritt. Gehen Sie auf die Seite "Parameter". Wählen Sie die gewünschten Spalten aus. Wählen Sie das Markierungsfeld aus, damit die Zieltabelle erstellt wird. Wählen Sie ein Warehouse-Ziel aus. Gehen Sie zu den Verarbeitungsoptionen und geben Sie die Parameter an. Klicken Sie "OK" an.

7. Starten des Capture-Programms

Geben Sie in ein DOS-Fenster Folgendes ein: ASNCCP quellendatenbank COLD PRUNE

Das Parameter COLD gibt an, dass ein Kaltstart ausgeführt wird, wodurch sämtliche Daten in den CD-Tabellen gelöscht werden. Der Parameter PRUNE weist das Capture-Programm an, die Tabelle IBMSNAP_PRUNCNTL zu erhalten. Lassen Sie das Capture-Programm aktiv. Sie können es beim Beenden der Sitzung mit der Tastenkombination Strg-Untbr im DOS-Fenster stoppen. Achten Sie darauf, das Capture-Programm vor dem Apply-Programm zu starten.

8. Hochstufen eines Replikationsschritts in den Testmodus

Stufen Sie den definierten Replikationsschritt in der Data Warehouse-Zentrale in den Testmodus hoch. Dadurch werden die Replikationssubskriptionsinformationen in die Replikationssteuertabellen geschrieben. Den Tabellen IBMSNAP_SUBS_SET, IBMSNAP_SUBS_MEMBR, IBMSNAP_SUBS_COLS und IBMSNAP_SUBS_EVENT werden Einträge zur Unterstützung der Subskription hinzugefügt. Die Zieltabelle wird ebenfalls in der Zieldatenbank erstellt. Bei den Replikationsarten "Benutzerkopie", "Mit Zeitangabe" und "Komprimierte Zwischenspeichertabelle" ist ein Primärschlüssel in der Zieltabelle erforderlich. Gehen Sie in die Steuerzentrale, um den Primärschlüssel zu

erstellen. Beachten Sie, dass einige Replikationszieltabellen außerdem eindeutige Indizes für bestimmte Spalten verlangen. In der Data Warehouse-Zentrale ist Code enthalten, mit dessen Hilfe diese eindeutigen Indizes beim Erstellen der Tabelle erstellt werden können, so dass Sie sie NICHT selbst erstellen müssen. Beachten Sie jedoch, dass Ihnen beim Definieren eines Primärschlüssels in der Steuerzentrale für eine Spalte, die bereits einen eindeutigen Index enthält, eine Warnung angezeigt wird. Diese Warnung können Sie ignorieren.

9. Hochstufen eines Replikationsschritts in den Produktionsmodus

Während einer Hochstufung in den Produktionsmodus wird die Replikations-
subskription nicht geändert. Dies ist ein völlig interner Arbeitsgang der Data
Warehouse-Zentrale wie jeder andere Schritt.

10. Ausführen eines Replikationsschritts

Nach der Hochstufung eines Replikationsschritts in den Testmodus kann dieser ausgeführt werden. Führen Sie ihn zunächst aus, ohne Änderungen an der Quellentabelle vorzunehmen. Gehen Sie in den Abschnitt zu laufenden Prozessen und wählen Sie den Replikationsschritt aus. Führen Sie ihn aus. Beim Ausführen des Schritts wird der Ereigniseintrag in der Tabelle IBMSNAP_SUBS_EVENT aktualisiert und der Subskriptionseintrag in IBMSNAP_SUBS_SET auf aktiv gesetzt. Die Subskription sollte sofort ausgeführt werden. Beim Ausführen der Subskription ruft der Agent das Apply-Programm auf, um die aktiven Subskriptionen zu verarbeiten. Wenn Sie danach die ursprüngliche Quellentabelle aktualisieren, werden die geänderten Daten in die CD-Tabelle versetzt. Wenn Sie anschließend den Replikationsschritt so ausführen, dass das Apply-Programm noch einmal ausgeführt wird, werden die geänderten Daten aus der CD-Tabelle in die Zieltabelle versetzt.

11. Herabstufen eines Replikationsschritts in den Testmodus

Während einer Herabstufung in den Testmodus wird die Replikations-
subskription nicht geändert. Dies ist ein völlig interner Arbeitsgang der Data
Warehouse-Zentrale wie jeder andere Schritt.

12. Herabstufen eines Replikationsschritts in den Entwicklungsmodus

Wenn Sie einen Replikationsschritt in den Entwicklungsmodus herabstufen, werden die Subskriptionsinformationen aus den Replikationssteuertabellen entfernt. Nach Abschluss der Herabstufung in den Entwicklungsmodus bleiben keine Einträge für die entsprechende Subskription in den Replikationssteuertabellen. Die Zieltabelle wird dann ebenfalls gelöscht. Die CD-Tabelle bleibt erhalten, da sie zur Definition der Replikationsquelle gehört.

13. Einrichten einer Replikation mit Basisergbnistabelle oder CD-Tabelle in der Data Warehouse-Zentrale

- Eingabetabelle. Wählen Sie eine Eingabetabelle aus, die mit einer GROUP BY-Anweisung verwendet werden kann. In unserem Beispiel verwenden wir eine Eingabetabelle mit folgenden drei Spalten: SALES, REGION, DISTRICT.
- Replikationsschritt. Wählen Sie "Basisergbnistabelle" oder "CD-Tabelle" aus. Öffnen Sie die Merkmale für den Schritt.
 - Wenn das Apply-Programm aktiv ist, muss es eine SELECT-Anweisung ausführen, die wie folgt aussieht: SELECT SUM(SALES), REGION, DISTRICT GROUP BY REGION, DISTRICT. In den ausgewählten Ausgabespalten müssen Sie daher REGION, DISTRICT und eine berechnete Spalte SUM(SALES) auswählen. Verwenden Sie den Knopf "Berechnete Spalte hinzufügen". Geben Sie in unserem Beispiel SUM(SALES) in das Feld "SQL-Ausdruck" ein. Sichern Sie die Eingabe.

- WHERE-Klausel. Beim Einrichten eines Replikationsschritts, für den nur eine GROUP BY-Klausel erforderlich ist, müssen Sie zusätzlich eine Pseudo-WHERE-Klausel angeben, beispielsweise 1=1. Die WHERE-Klausel darf NICHT das Wort "WHERE" enthalten. Daher gibt es in der grafischen Benutzerschnittstelle der Data Warehouse-Zentrale für die Basisergebnistabelle nur ein Eingabefeld für die WHERE-Klausel. Geben Sie in dieses Feld beispielsweise 1=1 GROUP BY REGION, DISTRICT ein. Für die CA-Tabelle gibt es sowohl eine WHERE-Klausel als auch ein Eingabefeld für GROUP BY: Geben Sie 1=1 in das Feld für die WHERE-Klausel und GROUP BY REGION, DISTRICT in das Feld für GROUP BY ein.
- Legen Sie die übrigen Merkmale für den Schritt wie bei den anderen Replikationsarten fest. Drücken Sie "OK", um den Schritt zu sichern und das Zieltabellenobjekt zu erstellen.
- Öffnen Sie das Zieltabellenobjekt. Nun müssen Sie den Namen der Ausgabespalte für den berechneten Spaltenausdruck in einen gültigen Spaltennamen ändern und einen gültigen Datentyp für die Spalte angeben. Sichern Sie das Zieltabellenobjekt.
- Stufen Sie den Replikationsschritt in der Testmodus hoch. Die Zieltabelle wird erstellt. Sie benötigt KEINEN Primärschlüssel.
- Führen Sie den Schritt wie jeden anderen Replikationsschritt aus.

Tipps zur Fehlerbehebung

- Wenn Sie einen Trace für das Apply-Programm aktivieren wollen, setzen Sie den Wert für Agenten-Trace in der Anzeige mit den Warehouse-Merkmalen auf 4. Bei einem Wert von 4 aktiviert der Agent einen vollständigen Trace für das Apply-Programm.
Wenn in der CD-Tabelle keine Daten angezeigt werden, ist wahrscheinlich entweder das Capture-Programm nicht gestartet oder die ursprüngliche Quellentabelle nicht zur Erstellung geänderter Daten aktualisiert worden.
- In der Onlinehilfefunktion fehlt das Feld "Post-Server" auf der Seite "Hinweis" im Notizbuch "Zeitplan".
- Der Post-Server muss ESMTTP unterstützen, damit die Hinweisfunktion der Data Warehouse-Zentrale funktioniert. Klicken Sie "Warehouse —> Laufende Prozesse" in der Hilfe zum Öffnen des Fensters "Laufende Prozesse" an, nicht "Warehouse-Zentrale —> Laufende Prozesse".

Zugreifen auf Quellen und Ziele

In den folgenden Tabellen werden die Versions- und Release-Stufen aufgeführt, die von der Data Warehouse-Zentrale unterstützt werden.

Tabelle 7. Versions- und Release-Stufen der unterstützten IBM Warehouse-Quellen

Quelle	Version/Release
IMS	5.1
DB2 Universal Database für Windows NT	5.2 - 7.1
DB2 Universal Database Enterprise-Extended Edition	5.2 - 7.1
DB2 Universal Database für OS/2	5.2 - 7.1
DB2 Universal Database für AS/400	3.7 - 4.5
DB2 Universal Database für AIX	5.2 - 7.1

Tabelle 7. Versions- und Release-Stufen der unterstützten IBM Warehouse-Quellen (Forts.)

Quelle	Version/Release
DB2 Universal Database für Solaris-Betriebsumgebung	5.2 - 7.1
DB2 Universal Database für OS/390	4.1 - 7.1
DB2 DataJoiner	2.1.1
DB2 für VM	5.3.4 oder höher
DB2 für VSE	7.1

Quelle	Windows NT	AIX
Informix	7.2.2 - 8.2.1	7.2.4 - 9.2.0
Oracle	7.3.2 - 8.1.5	8.1.5
Microsoft SQL Server	7.0	
Microsoft Excel	97	
Microsoft Access	97	
Sybase	11.5	11.9.2

Tabelle 8. Versions- und Release-Stufen der unterstützten IBM Warehouse-Ziele

Ziel	Version/Release
DB2 Universal Database für Windows NT	6 - 7
DB2 Universal Database Enterprise-Extended Edition	6 - 7
DB2 Universal Database für OS/2	6 - 7
DB2 Universal Database für AS/400	3.1-4.5
DB2 Universal Database für AIX	6 -7
DB2 Universal Database für Solaris-Betriebsumgebung	6 -7
DB2 Universal Database für OS/390	4.1 - 7
DB2 DataJoiner	2.1.1
DB2 DataJoiner/Oracle	8
DB2 für VM	3.4 - 5.3.4
DB2 für VSE	3.2, 7.1
CA/400	3.1.2

Ergänzungen zu unterstützten Datenbankquellen von Fremdherstellern

Die folgende Tabelle enthält Ergänzungen zu den unterstützten Datenbankquellen von Fremdherstellern:

Datenbank	Betriebssystem	Anforderungen für den Datenbank-Client
Informix	AIX	Informix-Connect und ESQL/C Version 9.1.4 oder höher
Informix	Solaris-Betriebsumgebung	Informix-Connect und ESQL/C Version 9.1.3 oder höher
Informix	Windows NT	Informix-Connect für Windows Platforms 2.x oder Informix-Client Software Developer's Kit for Windows Platforms 2.x
Oracle 7	AIX	Oracle7 SQL*Net und die gemeinsam benutzte Bibliothek von Oracle7 SQL*Net (von der Prozedur gencIntsh erzeugt)
Oracle 7	Solaris-Betriebsumgebung	Oracle7 SQL*Net und die gemeinsam benutzte Bibliothek von Oracle7 SQL*Net (von der Prozedur gencIntsh erzeugt)
Oracle 7	Windows NT	Die entsprechenden DLLs für die aktuelle Version von SQL*Net, plus OCIW32.DLL. SQL*Net 2.3 benötigt z. B. ORA73.DLL, CORE35.DLL, NLSRTL32.DLL, CORE350.DLL und OCIW32.DLL.
Oracle 8	AIX	Oracle8 Net8 und die gemeinsam benutzte Bibliothek von Oracle8 SQL*Net (von der Prozedur gencIntsh8 erzeugt)
Oracle 8	Solaris-Betriebsumgebung	Oracle8 Net8 und die gemeinsam benutzte Bibliothek von Oracle8 SQL*Net (von der Prozedur gencIntsh8 erzeugt)

Datenbank	Betriebssystem	Anforderungen für den Datenbank-Client
Oracle 8	Windows NT	<p>Für den Zugriff auf ferne Oracle8-Datenbank-Server ab Version 8.0.3 oder höher installieren Sie Oracle Net8 Client Version 7.3.4.x, 8.0.4 oder höher.</p> <p>Installieren Sie auf Intel-Systemen die entsprechenden DLLs für den Oracle Net8 Client (z. B. Ora804.DLL, PLS804.DLL und OCI.DLL) in Ihrem Pfad.</p>
Sybase	AIX	<p>In einer Nicht-DCE-Umgebung (ODBC-Treiber ibsyb15): Bibliothek libct</p> <p>In einer DCE-Umgebung (ODBC-Treiber ibsyb1115): Sybase 11.1 Client-Bibliothek libct_r</p>
Sybase	Solaris-Betriebsumgebung	<p>In einer Nicht-DCE-Umgebung (ODBC-Treiber ibsyb15): Bibliothek libct</p> <p>In einer DCE-Umgebung (ODBC-Treiber ibsyb1115): Sybase 11.1 Client-Bibliothek libct_r</p>
Sybase	Windows NT	Sybase Open Client-Library 10.0.4 oder höher und die entsprechende Sybase Net-Library

Manuelles Erstellen einer Datenquelle in der Data Warehouse-Zentrale

Wenn eine Datenquelle mit Relational Connect und der Anweisung „Create Nick-name“ erstellt wird, ist die Datenquelle in den Funktionen nicht verfügbar, die sich auf das Importieren von Tabellen in die Data Warehouse-Zentrale beziehen. Führen Sie die folgenden Schritte aus, um die Datenquelle als Quellen- oder Zieltabelle zu verwenden:

1. Definieren Sie das Ziel/die Quelle, ohne Tabellen zu importieren.
2. Erweitern Sie die Baumstruktur für die Warehouse-Ziele/Quellen im Hauptfenster der Data Warehouse-Zentrale, und klicken Sie „Tabellen“ für das gewünschte Ziel bzw. die gewünschte Quelle an.
3. Klicken Sie **Definieren an**.
4. Definieren Sie die Datenquelle über das angezeigte Notizbuch, und stellen Sie sicher, dass die Spalten für jede Datenquelle definiert sind.

Weitere Informationen finden Sie in den Abschnitten zum Definieren einer Warehouse-Quellentabelle bzw. einer Warehouse-Zieltabelle in „Information - Unterstützung“.

Importieren und Exportieren von Metadaten mit CWMI (Common Warehouse Metadata Interchange)

Einführung

Zusätzlich zur bereits vorhandenen Unterstützung für Befehlssprachendateien kann die Data Warehouse-Zentrale jetzt Metadaten aus XML-Dateien, die dem CWM-Standard (Common Warehouse Metamodel) entsprechen, importieren und Daten in diese Dateien exportieren. Das Importieren und Exportieren dieser CWM-kompatiblen XML-Dateien wird als *CWMI (Common Warehouse Metadata Interchange)* bezeichnet.

Sie können Metadaten aus den folgenden Objekten der Data Warehouse-Zentrale importieren bzw. in diese exportieren:

- Warehouse-Quellen
- Warehouse-Ziele
- Themenbereiche, einschließlich Prozessen, Quellen, Zielen und Schritten
- Benutzerdefinierte Programme

Das CWMI-Dienstprogramm für den Import und Export unterstützt zurzeit bestimmte Arten von Metadaten nicht. Dazu gehören Zeitpläne, Warehouse-Schemata, Direktaufschritte, verbundene Abhängigkeiten, Benutzer und Gruppen.

Die Data Warehouse-Zentrale erstellt eine Protokolldatei mit den Ergebnissen der Im- und Exportprozesse. In der Regel wird die Protokolldatei im Verzeichnis `x:\programmdateien\sql11ib\logging` (dabei ist `x`: das Laufwerk, auf dem DB2 installiert wurde) oder in dem Verzeichnis erstellt, das über die Umgebungsvariable `VWS_LOGGING` angegeben wurde. Die Protokolldatei ist eine Textdatei und kann mit einem beliebigen Texteditor angezeigt werden.

Importieren von Metadaten

Sie können Metadaten über die Data Warehouse-Zentrale oder über die Befehlszeile importieren.

Neue Objekte, die durch den Importprozess erstellt werden, werden der Standard-sicherheitsgruppe der Data Warehouse-Zentrale zugeordnet. Weitere Informationen finden Sie in den vorliegenden Release-Informationen im Abschnitt zum Aktualisieren der Sicherheit nach dem Import.

Wenn Sie Metadaten ohne einen Schritt importieren, können dem Schritt mehrere Dateien zugeordnet werden. Metadaten zum Schritt werden in einer XML-Datei gespeichert; gelegentlich werden dem Schritt zugeordnete Daten aber auch in BLOBs gespeichert. Die BLOB-Metadaten haben denselben Namen wie die XML-Datei, befinden sich aber in einzelnen Dateien, deren Erweiterungen durchnummeriert werden. Alle zugehörigen Schrittdateien müssen sich für den Import im selben Verzeichnis befinden.

Aktualisieren von Schritten, die sich im Test- oder Produktionsmodus befinden

Ein Schritt muss sich im Entwicklungsmodus befinden, damit die Data Warehouse-Zentrale die Metadaten des Schritts aktualisieren kann. Befindet sich der Schritt im Test- oder Produktionsmodus, muss er vor dem Import der Metadaten auf den Entwicklungsmodus herabgestuft werden:

1. Melden Sie sich an der Data Warehouse-Zentrale an.
2. Klicken Sie den Schritt, der herabgestuft werden soll, mit Maustaste 2 an, und klicken Sie dann **Modus** an.
3. Klicken Sie **Entwicklung** an.

Der Schritt befindet sich jetzt im Entwicklungsmodus. Setzen Sie den Schritt nach dem Import der Metadaten auf den Test- oder den Produktionsmodus zurück.

Importieren von Daten über die Data Warehouse-Zentrale

Sie können Metadaten über die Data Warehouse-Zentrale importieren:

1. Melden Sie sich an der Data Warehouse-Zentrale an.
2. Klicken Sie im linken Teilfenster **Warehouse** an.
3. Klicken Sie **Ausgewählt** —> **Metadaten importieren** —> **Datei im Datenaustauschformat** an.
4. Geben Sie im Fenster **Metadaten importieren** den Namen der Datei mit den Metadaten an, die Sie importieren wollen. Sie können den Dateinamen eingeben oder nach der Datei suchen.
 - Wenn Sie die Speicherposition der Datei kennen, geben Sie den vollständig qualifiziertem Pfad und den Namen der Datei ein, die Sie importieren wollen. Achten Sie darauf, die Dateierweiterung `.xml` anzugeben, um festzulegen, dass Sie Metadaten im XML-Format importieren wollen.

- Gehen Sie wie folgt vor, um die Dateien zu durchsuchen:
 - a. Klicken Sie den Druckknopf mit dem Auslassungszeichen (...) an.
 - b. Ändern Sie den Dateityp im Dateifenster in XML.
 - c. Wechseln Sie in das entsprechende Verzeichnis, und wählen Sie die Datei aus, die Sie importieren wollen.

Anmerkung: Die Datei muss die Erweiterung `.xml` haben.

- d. Klicken Sie **OK** an.
5. Klicken Sie **OK** im Fenster **Metadaten importieren** an, um den Vorgang zu beenden. Ein Statusfenster wird angezeigt, während die Data Warehouse-Zentrale die Datei importiert.

Importieren von Metadaten über die Befehlszeile

Sie können Metadaten auch über die Befehlszeile importieren. Die Befehlssyntax für den Import lautet wie folgt:

```
CWMImport XML_datei dwcsteuerDB dwcBenutzerID dwcKW [PREFIX = DWctbschema]
```

<i>XML_datei</i>	Der vollständig qualifizierte Pfad (einschließlich Laufwerk und Verzeichnis) und der Name der XML-Datei, die Sie importieren wollen. Dieser Parameter ist erforderlich.
<i>dwcControlDB</i>	Der Name der Warehouse-Steuerungsdatenbank, in die Sie Ihre Metadaten importieren wollen. Dieser Parameter ist erforderlich.
<i>dwcBenutzerID</i>	Die Benutzer-ID, mit der Sie sich an der Warehouse-Steuerungsdatenbank anmelden. Dieser Parameter ist erforderlich.
<i>dwcKW</i>	Das Benutzerkennwort, mit dem Sie sich an der Warehouse-Steuerungsdatenbank anmelden. Dieser Parameter ist erforderlich.
<i>[PREFIX=DWctbschema]</i>	Der Name des Datenbankschemas für die Systemtabellen der Data Warehouse-Zentrale. Dieser wird gelegentlich auch als <i>Tabellenpräfix</i> bezeichnet. Wird kein Wert für PREFIX= angegeben, wird der Standardschemename IWH verwendet. Dieser Parameter ist optional.

Aktualisieren der Metadaten nach dem Ausführen des Importdienstprogramms

Aktualisieren der Sicherheit nach dem Import

Eine der Sicherheitsmaßnahmen der Data Warehouse-Zentrale besteht darin, Kennwörter weder zu importieren noch zu exportieren. Sie müssen die Kennwörter für neue Objekte ggf. aktualisieren. Ausführlichere Informationen zu Überlegungen beim Import von Metadaten finden Sie in Kapitel 12 des Handbuchs *Data Warehouse-Zentrale Verwaltung*.

Beim Importieren von Metadaten werden alle Objekte der *Standardsicherheitsgruppe* zugeordnet. Sie können die Gruppen ändern, die Zugriff auf dieses Objekt haben:

1. Melden Sie sich an der Data Warehouse-Zentrale an.
2. Klicken Sie den Ordner, der das zu ändernde Objekt enthält, mit Maustaste 2 an.
3. Klicken Sie **Merkmale** und anschließend die Indexzunge **Sicherheit** an.
4. Entfernen Sie Gruppen aus der Liste der ausgewählten Warehouse-Gruppen, oder fügen Sie der Liste der verfügbaren Warehouse-Gruppen Gruppen hinzu.
5. Klicken Sie **OK** an.

Exportieren von Metadaten

Sie können Metadaten über die Data Warehouse-Zentrale oder über die Befehlszeile exportieren.

Einige Schritte verfügen über Metadaten, die als BLOB gespeichert sind. Die BLOB-Metadaten werden in eine separate Datei exportiert, die denselben Namen wie die XML-Datei des Schritts hat, deren Erweiterung aber durchnummeriert wird (.1, .2 usw.).

Exportieren von Daten über die Data Warehouse-Zentrale

Sie können Metadaten über die Data Warehouse-Zentrale exportieren:

1. Melden Sie sich an der Data Warehouse-Zentrale an.
2. Klicken Sie im linken Teilfenster **Warehouse** an.
3. Klicken Sie **Ausgewählt** —> **Metadaten exportieren** —> **Datei im Datenaustauschformat** an.
4. Geben Sie im Fenster **Metadaten exportieren** den Namen der Datei mit den exportierten Metadaten an. Sie können den Dateinamen eingeben oder nach der Datei suchen:
 - Wenn Sie die vollständig qualifizierten Pfad und den Namen der Datei kennen, geben Sie diese Angaben in das Feld für den Dateinamen ein. Achten Sie darauf, die Dateierweiterung `.xml` anzugeben, um festzulegen, dass Sie Metadaten im XML-Format exportieren wollen.

- Gehen Sie wie folgt vor, um die Dateien zu durchsuchen:
 - a. Klicken Sie den Druckknopf mit dem Auslassungszeichen (...) an.
 - b. Ändern Sie den Dateityp im Dateifenster in XML.
 - c. Wechseln Sie in das entsprechende Verzeichnis, und wählen Sie die Datei aus, die Sie exportieren wollen.

Anmerkung: Wenn Sie eine vorhandene Datei auswählen, wird diese mit den exportierten Metadaten überschrieben.

- d. Klicken Sie **OK** an.
5. Wenn im Fenster **Metadaten exportieren** der korrekte Dateiname angezeigt wird, klicken Sie das Objekt, dessen Metadaten Sie exportieren wollen, in der Liste der verfügbaren Objekte an.
6. Klicken Sie das Zeichen > an, um das ausgewählte Objekt aus der Liste der verfügbaren Objekte in die Liste der ausgewählten Objekte zu verschieben. Wiederholen Sie diese Schritte, bis alle Objekte, die Sie exportieren wollen, in der Liste der ausgewählten Objekte aufgeführt werden.
7. Klicken Sie **OK** an.

Die Data Warehouse-Zentrale erstellt eine *Eingabedatei* mit Informationen zu den Objekten der Data Warehouse-Zentrale, die Sie für den Export ausgewählt haben, und exportiert die Metadaten zu diesen Objekten anschließend. Ein Statusfenster wird angezeigt, während die Data Warehouse-Zentrale die Metadaten exportiert.

Nach Beendigung des Exportprozesses erhalten Sie eine entsprechende Informationsnachricht. Der Rückkehrcode 0 gibt an, dass der Export erfolgreich war. Sie können der Protokolldatei genauere Informationen entnehmen.

Exportieren von Metadaten über die Befehlszeile

Bevor Sie Metadaten über die Befehlszeile exportieren können, müssen Sie eine *Eingabedatei* erstellen. Die Eingabedatei ist eine Textdatei mit der Erweiterung .INP und listet alle Objekte, die Sie exportieren wollen, nach dem Objekttyp auf. Bei einem Export über die Data Warehouse-Zentrale wird die Eingabedatei automatisch erstellt, bei einem Export über die Befehlszeile muss diese Datei jedoch zuerst erstellt werden. Sie können die Eingabedatei mit einem beliebigen Texteditor erstellen. Geben Sie die Objektnamen so ein, wie sie in der Data Warehouse-Zentrale angezeigt werden. Stellen Sie sicher, dass die Datei sich in einem Verzeichnis mit Schreib-/Lesezugriff befindet. Beim Ausführen des Exportdienstprogramms schreibt die Data Warehouse-Zentrale die XML-Dateien in das Verzeichnis, in dem sich die Eingabedatei befindet.

Eine Beispieleingabedatei:

```
<PROC>
Prozess für Lernprogramm fakttabelle
<IR>
Dateiquelle für Lernprogramm
Ziel für Lernprogramm
<UDP>
Neue Programmgruppe
```

Listen Sie im Abschnitt <PROC> (Prozess) alle Prozesse auf, die Sie exportieren wollen. Listen Sie im Abschnitt <IR> (Informationsquellen) alle Warehouse-Quellen und -Ziele auf, die Sie exportieren wollen.

Die Data Warehouse-Zentrale schließt die Tabellen und Spalten, die diesen Zielen und Quellen zugeordnet sind, automatisch ein. Listen Sie im Abschnitt <UDP> (benutzerdefinierte Programme) die Programmgruppen auf, die Sie exportieren wollen.

Geben Sie zum Exportieren der Metadaten den folgenden Befehl in eine DOS-Befehlszeile ein:

```
CWMEExport INPsteuerdatei dwcsteuerDB dwcBenutzerID dwcKW [PREFIX=DWCtbschema]
```

INPsteuerdatei

Der vollständig qualifizierte Pfad (einschließlich Laufwerk und Verzeichnis) und der Name der .INP-Datei mit den Objekten, die Sie exportieren wollen. Dieser Parameter ist erforderlich.

dwcControlDB

Der Name der Warehouse-Steuerungsdatenbank, aus der Sie exportieren wollen. Dieser Parameter ist erforderlich.

dwcBenutzerID

Die Benutzer-ID, mit der Sie sich an der Warehouse-Steuerungsdatenbank anmelden. Dieser Parameter ist erforderlich.

dwcKW

Das Kennwort, mit dem Sie sich an der Warehouse-Steuerungsdatenbank anmelden. Dieser Parameter ist erforderlich.

[PREFIX=DWCtbschema]

Der Name des Datenbankschemas für die Systemtabellen der Data Warehouse-Zentrale. Dieser wird gelegentlich auch als *Tabellenpräfix* bezeichnet. Wird kein Wert für PREFIX= angegeben, wird standardmäßig IWH verwendet. Dieser Parameter ist optional.

Schritt des OS/390-Dienstprogramms Runstats

Beim Definieren eines Schritts des OS/390-Dienstprogramms Runstats in der Data Warehouse-Zentrale sollte der Benutzer folgende Punkte auf der Indexzunge **Parameter** des Dialogs für die Schrittmerkmale beachten. Im Feld **Tabellenbereich** sollte der Benutzer den Namen in Großbuchstaben eingeben. Wenn der Tabellenbereich sich nicht in der Datenbank DSADB04 befindet, muss der Tabellenbereichsname von der Datenbank qualifiziert werden, in der er enthalten ist. Geben Sie z. B. SAMPLE.EMPLOYEE ein. In der Hilfe wird dieses Feld zurzeit nicht erwähnt.

Schritt des OS/390-Dienstprogramms Load

Beim Definieren eines Schritts des OS/390-Dienstprogramms Load in der Data Warehouse-Zentrale sollte der Benutzer folgende Punkte auf der Indexzunge **Parameter** des Dialogs für die Schrittmerkmale beachten. Damit der Ladevorgang funktioniert, muss der Benutzer immer den Knopf **Erweitert** auswählen. Andernfalls wird die Klausel INTO der Ladeanweisung nicht generiert, und der Ladevorgang schlägt fehl.

Darüber hinaus enthält FixPak 3 einen Fix, der die doppelten Anführungszeichen um den Namen der Ladedatei entfernt. Ohne diesen Fix funktioniert der Ladevorgang nicht.

XML-Unterstützung für Common Warehouse Metamodel (CWM)

Das CWM-Toolkit Version 7.2 arbeitet mit Java Development Kit (JDK) 1.2.2 oder 1.3.

Sie können jetzt die folgenden CWM-XML-Objekte importieren und exportieren:

Direktaufrufschritte von anderen Prozessen

Wenn Sie einen Prozess exportieren, der einen Schritt enthält, der mit einem Schritt in einem anderen Prozess verbunden ist (ein Direktaufwurf), werden beide Prozesse exportiert, und die Beziehung wird beibehalten.

Bedingt überlappend angeordnete Beziehungen

Sie können jetzt unterschiedliche überlappend angeordnete Beziehungen zwischen Schritten importieren und exportieren, einschließlich CHILD, SUCCESS, FAILURE und UNCONDITIONAL.

Warehouse-Quellen als Sichtobjekte

Wenn Sie exportieren, können Sie jetzt eine Warehouse-Quelle als Sichtobjekt definieren. Sichtobjekte werden in derselben Weise verarbeitet wie Tabellenobjekte.

SQLDataType für Spalten und Felder

Sie können jetzt den Datentyp SQLDataType für Spalten und Felder verwenden.

Mehrere Korrelationsnamen für dieselbe Tabelle

Während des Imports oder Exports können Sie mehrere Korrelationsnamen haben, von denen jeder seine eigene Spaltenzuordnung für dieselbe Tabelle hat.

Neue Unterstützung für SAP- und WebSphere Site Analyzer-Quellen

Durch Hinzufügen von neuen Quellenunterstützungsbefehlen, können Sie jetzt SAP- und WSA-Informationen (WebSphere Site Analyzer) exportieren, die in Ihrem Warehouse gespeichert sind.

Prozesserstellungsprogramm

Sie können die Größe der Prozesserstellungsprogrammpalette an Ihren Bildschirm anpassen. Die Symbole in der Palette werden automatisch in einer mehrspaltigen Palette neu angeordnet.

Wenn Sie ein Palettensymbol anklicken, sehen Sie eine Kopfzeile in Palettenobjekten.

Sie können jetzt Tabellen- und Dateiobjekte mit ihrem Geschäftsnamen anzeigen, indem Sie die Option **Geschäftsnamen anzeigen** im Menü **Anzeigen** anklicken. Sie können auch die Einstellung des Prozentsatzes ändern und Ihre Prozesssichten kleiner oder größer anzeigen, indem Sie die Option **Zoomen** auswählen.

Wenn Objekte sich in der Palette überlappen, können Sie die Objekte anklicken, um Sie in den Bildschirmvordergrund zu holen. Außerdem brechen Objektnamen jetzt in mehrere Zeilen um, um Palettenplatz zu sparen.

Sie können Objekte jetzt mit der Taste „Entf“ löschen. Änderungen an der Tabelle werden gespeichert, wenn Sie einen Prozess speichern. Das Auswahlverhalten ist nicht automatisch. Wenn Sie eine Tabelle, Datei oder Sicht aus einer Warehouse-Quelle, einem -Ziel oder einem -Prozess löschen wollen, können Sie das Objekt mit Maustaste 2 anklicken und die Aktion **Aus Quelle entfernen** auswählen, wenn sich das Objekt in einer Warehouse-Quelle befindet oder die Aktion **Aus Ziel entfernen**, wenn das Objekt sich in einem Warehouse-Ziel befindet.

Der Cursor zeigt jetzt den Palettenauswahlstatus an. Außerdem zeigt die Statuszeile den Namen des Objekts an, auf dem sich der Cursor befindet.

Schemamodellierungsprogramm

Sie können die Tabellen in dem Schemamodellierungsprogramm jetzt minimieren und maximieren. Wenn Sie eine Tabelle minimieren, wird sie zu einem Symbol. Sie können nun ein Sternschema-Layout für höhere visuelle Genauigkeit erstellen.

Musseingabefelder

Die Data Warehouse-Zentrale zeigt jetzt einen roten Rahmen um erforderliche Felder an. Der rote Rahmen erinnert Sie an verbindliche Informationen wie z. B. Datenbanknamen, Benutzer-IDs oder Kennwörter, die notwendig sind, um Data Warehouse-Zentralenobjekte zu definieren. Wenn Sie die erforderlichen Informationen eingegeben haben, wird der Rahmen nicht mehr angezeigt.

Erweiterungen des Assistenten für die Data Warehouse-Zentrale

Wenn Sie ein Data Warehouse-Zentralenobjekt mit dem Assistenten erstellen, wird die Navigationsbaumstruktur erweitert, um die Adresse anzuzueigen, an der sich das neue Objekt befindet.

Ausgeben von Schrittinformationen in eine Datei

Sie können jetzt Informationen zu einem Schritt (wie z. B. Themenbereich, Quellen- und Zieltabellennamen) in eine Textdatei ausgeben. Gehen Sie wie folgt vor, um Schrittinformationen in eine Datei auszugeben: Klicken Sie das Schrittsymbol im Prozesserstellungsprogramm mit Maustaste 2 an, klicken Sie **Drucken** → **Ausgabe in eine Datei** an, und geben Sie den Namen der Datei an, in die Sie die Informationen ausgeben wollen.

Data Warehouse Center Application Integration Guide

In Chapter 5. Metadata Templates, Table 16 describes Column tag tokens. The information in the manual should state that `*ColumnPositionNumber` should start with "1". The manual incorrectly gives "0" as the starting character.

Later in Chapter 5, in Table 42, the `TableTypeIfFile` token is required if the type specified for the `DatabaseType` token in the corresponding `SourceDataBase.tag` is `ISV_IR_FFLan`. If it is not specified, an error will be detected.

In Chapter 6. Data Warehouse Center metadata, the description of the `POSNO` column object property should be changed to:

An index, starting with 1, of the column
or field in the row of the table or file.

In Chapter 8. Information Catalog Manager object types, the directory where you can find the `.TYP` files, which include the tag language for defining an object type, has been changed to `\SQLLIB\DGWIN\TYPES`.

Additional metadata templates

In Chapter 5, Metadata Templates, the following metadata templates should be included.

Tabelle 9. New Metadata templates supplied with the Data Warehouse Center

Template	Description	See:
<code>commit.tag</code>		„Commit.tag“
<code>Foreignkey.tag</code>		„ForeignKey.tag“ auf Seite 282
<code>Foreignkeyadditional.tag</code>		„ForeignKeyAdditional.tag“ auf Seite 284
<code>Primarykey.tag</code>		„PrimaryKey.tag“ auf Seite 286
<code>Primarykeyadditional.tag</code>		„PrimaryKeyAdditional.tag“ auf Seite 287

Commit.tag

Use this template to improve performance when you are using large tag language files. A commit template can be inserted between any of the groups of templates described here. A commit template cannot be inserted between templates within a group. For example, it is valid to insert a commit template between `AgentSite.tag` and `VWPGGroup.tag` but invalid to insert a commit tag between `VWPPProgramTemplate.tag` and `VWPPProgramTemplateParameter.tag`. If commit templates are used incorrectly, import may report an error.

- `AgentSite.tag`
- `VWPGGroup.tag`
- `VWPPProgramTemplate.tag`, `VWPPProgramTemplateParameter.tag`
- `SourceDatabase.tag`
- `WarehouseDatabase.tag`
- `Table.tag`, `Column.tag`
- `SubjectArea.tag`
- `Process.tag`

- Step.tag, StepInputTable.tag, StepOutputTable.tag, StepVWPOutputTable.tag, StepVWPPProgramInstance.tag, VWPPProgramInstanceParameter.tag
- StepCascade.tag
- StarSchema.tag, StarSchemaInputTable.tag
- PrimaryKey.tag, PrimaryKeyAdditional.tag
- ForeignKey.tag, ForeignKeyAdditional.tag

The use of the commit template is optional.

Tokens

Tabelle 10 provides information about each token in the template.

Tabelle 10. Commit.tag tokens

Token	Description	Allowed values
Relationship parameters		
*CurrentCheckPointID++	An index, starting with 0, that increases each time it is substituted in a token.	A numeric value.
This token is required.		

Examples of values

Tabelle 11 provides example values for each token to illustrate the kind of metadata you might provide for each token.

Tabelle 11. Example values for Commit.tag tokens

Token	Example value
*CurrentCheckPointID++	1

ForeignKey.tag

Use this template to define foreign key constraints on tables. The ForeignKey.tag template defines the relationships to the table and the column on which the constraint is being defined. This template also defines the relationships to the table and column of the primary key that is being referred to. Before you use the ForeignKey.tag template, you must define the primary key constraint (using the PrimaryKey.tag template) and the tables and columns (using the Table.tag and Column.tag templates) on which you want to define the foreign key constraint.

Tokens

Tabelle 12 provides information about each token in the template.

Tabelle 12. ForeignKey.tag tokens

Token	Description	Allowed values
Entity parameters		
*ConstraintName	The name of the constraint. The name must be unique within a table or field. This token is required.	A text string, up to 80 bytes in length.
*ForeignColumnKeyName	Name of the column on which the foreign key constraint is being defined.	A text string, up to 254 bytes in length.

Table 12. ForeignKey.tag tokens (Forts.)

Token	Description	Allowed values
*ForeignKeyID	<p>The key that uniquely identifies the foreign key. The key must be unique from all other keys in the tag language file.</p> <p>Tip: Finish processing the ForeignKey.tag template before increasing the value of the key.</p> <p>This token is required.</p>	A numeric value.
*MapID	<p>An arbitrary number that is unique from all other keys in the interchange file.</p> <p>Tip: Finish processing the ForeignKey.tag template before increasing the value of this token.</p> <p>This token is required.</p>	A numeric value.
*PrimaryColumnName	The column name of the referenced column.	A text string, up to 80 bytes in length.
*ReferencedPrimaryKeyID	<p>The key that uniquely identifies the primary key. The key must be unique from all other keys in the tag language file.</p> <p>Tip: Finish processing the ForeignKey.tag template before increasing the value of the key.</p> <p>This token is required.</p>	A numeric value.
Relationship parameters		
*DatabaseName	<p>The business name of the warehouse source or warehouse target.</p> <p>This token is required.</p>	A text string, up to 40 bytes in length.
*ForeignTablePhysicalName	The database-defined name of the physical table containing the foreign keys that reference the keys in other tables.	A text string, up to 254 bytes in length.
*PrimaryTablePhysicalName	The database-defined name of the physical table containing the keys that are referenced by the foreign keys.	A text string, up to 80 bytes in length.
*PrimaryTableOwner	<p>The owner, high-level qualifier, collection, or schema of the table that contains the primary key column that is being referenced.</p> <p>This token is required.</p>	A text string, up to 128 bytes in length.

Tabelle 12. ForeignKey.tag tokens (Forts.)

Token	Description	Allowed values
*ForeignTableOwner	The owner, high-level qualifier, collection, or schema of the table that contains the foreign key constraint column.	A text string, up to 128 bytes in length.
	This token is required.	

Examples of values

Tabelle 13 provides example values for each token to illustrate the kind of meta-data that you might provide for each token.

Tabelle 13. Example values for ForeignKey.tag tokens

Token	Example value
*ConstraintName	Department
*DatabaseName	Finance Warehouse
*ForeignColumnName	Country_code
*ForeignKeyID	07011
*ForeignTablePhysicalName	Geography
*MapID	02568
*PrimaryColumnName	State_code
*Referenced PrimaryKeyID	Name
*PrimaryTablePhysicalName	City
*PrimaryTableOwner	DB2ADMIN
*ForeignTableOwner	IWH

ForeignKeyAdditional.tag

Use this template to define a composite foreign key. Before you use the ForeignKeyAdditional.tag template, you must define a constraint (using the ForeignKey.tag template) on the first column. You can then add columns by using this template for each column that you want to add.

Tokens

Tabelle 14 provides information about each token in the template.

Tabelle 14. ForeignKeyAdditional.tag tokens

Token	Description	Allowed values
Entity parameters		
*ForeignColumnName	The name of the column on which the foreign key constraint is being defined.	A text string, up to 80 bytes in length.

Table 14. ForeignKeyAdditional.tag tokens (Forts.)

Token	Description	Allowed values
*ForeignKeyID	<p>The key that uniquely identifies the foreign key. The key must be unique from all other keys in the tag language file.</p> <p>Tip: Finish processing the ForeignKey.tag template before increasing the value of the key.</p> <p>This token is required.</p>	A numeric value.
*MapID	<p>An arbitrary number that is unique from all other keys in the interchange file.</p> <p>Tip: Finish processing the ForeignKeyAdditional.tag template before increasing the value of this token.</p> <p>This token is required.</p>	A numeric value.
*MapSeqNo	<p>A number signifying each additional column added as part of a composite key to the foreign key constraint.</p>	Unique, increasing, consecutive number starting at 2.
*PrimaryColumnKeyName	<p>The column name of the referenced column.</p>	A text string, up to 80 bytes in length.
Relationship parameters		
*DatabaseName	<p>The business name of the warehouse source or warehouse target.</p> <p>This token is required.</p>	A text string, up to 40 bytes in length.
*ForeignTablePhysicalName	<p>The database-defined name of the physical table containing the foreign keys that reference the keys in other tables.</p>	A text string, up to 80 bytes in length.
*PrimaryTablePhysicalName	<p>The database-defined name of the physical table containing the keys that are referenced by the foreign keys.</p>	A text string, up to 80 bytes in length.
*PrimaryTableOwner	<p>The owner, high-level qualifier, collection, or schema of the table that contains the primary key column that is being referenced.</p> <p>This token is required.</p>	A text string, up to 128 bytes in length.
*ForeignTableOwner	<p>The owner, high-level qualifier, collection, or schema of the table that contains the foreign key constraint column.</p> <p>This token is required.</p>	A text string, up to 128 bytes in length.

Examples of values

Table 15 provides example values for each token to illustrate the kind of meta-data that you might provide for each token.

Table 15. Example values for *ForeignKeyAdditional.tag* tokens

Token	Example value
*DatabaseName	Finance Warehouse
*ForeignKeyColumnName	Country_code
*ForeignKeyID	07011
*ForeignKeyTablePhysicalName	Geography
*MapID	22578
*MapSeqNo	2
*PrimaryColumnName	State_code
*PrimaryTablePhysicalName	City
*PrimaryTableOwner	DB2ADMIN
*ForeignKeyTableOwner	IWH

PrimaryKey.tag

Use this template to define primary key constraints on tables. The template also defines the relationships to the table and the column on which the constraint is being defined. Before you use the *PrimaryKey.tag* template, you must define the tables and columns (using the *Table.tag* and *Column.tag* templates) on which you want to define the primary key constraint.

Tokens

Table 16 provides information about each token in the template.

Table 16. *PrimaryKey.tag* tokens

Token	Description	Allowed values
Entity parameters		
*ColumnName	The name of the column or field. The name must be unique within a table or field. This token is required.	A text string, up to 80 bytes in length.
*MapID	An arbitrary number that is unique from all other keys in the interchange file. Tip: Finish processing the <i>PrimaryKey.tag</i> template before increasing the value of this token. This token is required.	A numeric value.

Tabelle 16. PrimaryKey.tag tokens (Forts.)

Token	Description	Allowed values
*PrimaryKeyID	The key that uniquely identifies the primary key. The key must be unique from all other keys in the tag language file. Tip: Finish processing the ForeignKey.tag template before increasing the value of the key. This token is required.	A numeric value.
Relationship parameters		
*DatabaseName	The business name of the warehouse source or warehouse target. This token is required.	A text string, up to 40 bytes in length.
*TableOwner	The owner, high-level qualifier, collection, or schema of the table that contains the column. This token is required.	A text string, up to 128 bytes in length.
*TablePhysicalName	The physical name of the table or file that contains the column as defined to the database manager or file system. This token is required.	A text string, up to 80 bytes in length.

Examples of values

Tabelle 17 provides example values for each token to illustrate the kind of meta-data that you might provide for each token.

Tabelle 17. Example values for PrimaryKey.tag tokens

Token	Example value
*ColumnName	Country_code
*DatabaseName	Finance Warehouse
*MapID	54627
*PrimaryKeyID	74622
*TableOwner	DB2ADMIN
*TablePhysicalName	GEOGRAPHY

PrimaryKeyAdditional.tag

Use this template to define a composite primary key. Before you use the PrimaryKeyAdditional.tag template, you must define a constraint on the first column by using the PrimaryKey.tag template. Any additional columns can then be added using this template. The template also relates the additional primary keys to the first primary key which is defined using PrimaryKey.tag.

Tokens

Tabelle 18 provides information about each token in the template.

Tabelle 18. PrimaryKeyAdditional.tag tokens

Token	Description	Allowed values
Entity parameters		
*ColumnName	<p>The name of the column or field.</p> <p>The name must be unique within a table or field.</p> <p>This token is required.</p>	A text string, up to 80 bytes in length.
*FirstPrimaryKeyID	<p>The key that uniquely identifies the primary key. The key must be unique from all other keys in the tag language file.</p> <p>Tip: Finish processing the ForeignKey.tag template before increasing the value of the key.</p> <p>This token is required.</p>	A numeric value.
*MapID	<p>An arbitrary number that is unique from all other keys in the interchange file.</p> <p>Tip: Finish processing the PrimaryKeyAdditional.tag template before increasing the value of this token.</p> <p>This token is required.</p>	A numeric value.
*MapSeqNo	<p>A number signifying each additional column added as part of a composite key to the primary key constraint.</p>	Unique, increasing, consecutive number starting at 2.
Relationship parameters		
*DatabaseName	<p>The business name of the warehouse source or warehouse target.</p> <p>This token is required.</p>	A text string, up to 40 bytes in length.
*TableOwner	<p>The owner, high-level qualifier, collection, or schema of the table that contains the column.</p> <p>This token is required.</p>	A text string, up to 15 bytes in length.
*TablePhysicalName	<p>The physical name of the table or file that contains the column as defined to the database manager or file system.</p> <p>This token is required.</p>	A text string, up to 80 bytes in length.

Examples of values

Tabelle 19 provides example values for each token to illustrate the kind of meta-data that you might provide for each token.

Tabelle 19. Example values for PrimaryKeyAdditional.tag tokens

Token	Example value
<i>*ColumnName</i>	Country_code
<i>*DatabaseName</i>	Finance Warehouse
<i>*MapID</i>	99542
<i>*MapSeqNo</i>	2
<i>*FirstPrimaryKeyID</i>	07801
<i>*TableOwner</i>	DB2ADMIN
<i>*TablePhysicalName</i>	GEOGRAPHY

Data Warehouse-Zentrale - Onlinehilfefunktion

Definieren von Tabellen oder Sichten zur Replikation

Eine Tabelle oder Sicht muss mit der DB2-Steuerzentrale für die Replikation definiert werden, bevor sie in einer Data Warehouse-Zentrale als Replikationsquelle verwendet werden kann.

Ausführen von Essbase VWP's mit dem AS/400-Agenten

Bevor die Essbase-VWP's mit dem AS/400-Agenten ausgeführt werden, müssen ARBORLIB und ARBORPATH als *sys-Umgebungsvariablen gesetzt werden. Die Benutzer-ID muss über die Berechtigung *jobctl verfügen, um diese Variablen setzen zu dürfen. Diese Umgebungsvariablen müssen auf die Bibliothek zeigen, auf der Essbase installiert ist.

Verwenden der Fenster "Veröffentlichen von Metadaten der Data Warehouse-Zentrale" und dem zugehörigen Merkmalfenster

In Schritt 10 der Hilfe zur Task wird angegeben, dass bei der Angabe des Grenzwerts 1 (zur Begrenzung der Objekte in einer Baumstruktur) nur 1 Schritt dieses Prozesses veröffentlicht und angezeigt wird. Dieses Beispiel trifft nicht in allen Situationen zu. In Schritt 8 ist die erste Anweisung des zweiten Listenpunkts nicht korrekt. Sie muss wie folgt lauten: "Klicken Sie die Spaltenebene an, um ein Objekt zur Umsetzung zwischen einer Quellenspalte und einer Zielspalte des Informationskatalogs zu generieren."

Fremdschlüssel

Alle Verweise in der Onlinehilfefunktion auf "Fremdschlüssel" sollten "Warehouse-Fremdschlüssel" lauten.

Replikationsnotizbücher

Alle Verweise in der Onlinehilfefunktion auf das Notizbuch "Replikation definieren" sollten "Notizbuch 'Replikationsschritt'" lauten.

Importieren einer Befehlssprachendatei

Im Abschnitt "Importieren einer Befehlssprachendatei" der Onlinehilfefunktion gibt es eine Liste mit allgemeinen Fehlern beim Importieren. Einer davon lautet: "Import einer Befehlssprachendatei, die nicht richtig exportiert worden ist". Dieser Punkt gehört nicht in die Liste der allgemeinen Eingabefehler.

Links zum Hinzufügen von Daten

Im Abschnitt „Daten hinzufügen“ der Onlinehilfefunktion sind die Links zu den Themen „Hinzufügen von Quellentabellen zu einem Prozess“ und „Hinzufügen von Zieltabellen zu einem Prozess“ defekt. Sie finden diese Stichworte im Hilfeindex.

Importieren von Tabellen

Die Hilfethemen "Importieren von Quellentabellen und -sichten in eine Warehouse-Quelle" und "Importieren von Zieltabellen in ein Warehouse-Ziel" enthalten falsche Informationen zu Platzhalterzeichen. Der folgende Satz:

Wenn Sie zum Beispiel XYZ* eingeben, würden Tabellen und Sichten mit Schemata zurückgegeben werden, die mit diesen Zeichen beginnen.

sollte folgendermaßen lauten:

Wenn Sie zum Beispiel XYZ% eingeben, würden Tabellen und Sichten mit Schemata zurückgegeben werden, die mit diesen Zeichen beginnen.

Korrektur der Onlinehilfefunktion zu RUNSTATS und REORGANIZE TABLE

In der Onlinehilfefunktion für diese Dienstprogramme steht, dass die Tabelle, für die Statistik ausgeführt bzw. die reorganisiert werden soll, als Quelle UND als Ziel verbunden werden muss. Da der Schritt jedoch in die Quelle schreibt, müssen Sie nur die Quelle mit dem Schritt verbinden.

Seite "Hinweis" (Anzeige der Warehouse-Merkmale und Notizbuch "Zeitplan")

Auf der Seite **Hinweis** der Anzeige der Warehouse-Merkmale ist bisher folgende Aussage enthalten:

Das Eingabefeld **Absender** wird mit der Zeichenfolge <Anmelde-ID des aktuellen Benutzers> initialisiert.

Dies sollte geändert werden in:

Das Eingabefeld **Absender** wird mit der Zeichenfolge <E-Mail-Adresse des angemeldeten Benutzers> initialisiert.

Auf der Seite **Hinweis** des Notizbuchs **Zeitplan** wird der Absender mit der Einstellung aus der Anzeige der Warehouse-Merkmale initialisiert: Wenn keine Einstellung vorgenommen wurde, wird er mit der E-Mail-Adresse des aktuell angemeldeten Benutzers initialisiert. Wenn dem angemeldeten Benutzer keine E-Mail-Adresse zugeordnet ist, wird der Absender auf die Anmelde-Benutzer-ID eingestellt.

Das Agentenmodulfeld im Notizbuch "Agenten-Site"

Das Agentenmodulfeld im Notizbuch "Agenten-Site" enthält den Namen des Programms, das beim Erstellen des Warehouse-Agenten durch den Warehouse-Agentendämon ausgeführt wird. Ändern Sie den Namen in diesem Feld nicht, sofern IBM Sie nicht dazu auffordert.

DB2 OLAP Starter Kit

IBM DB2 OLAP Starter Kit 7.2 fügt die Unterstützung für Oracle, MS-SQL, Sybase und auf bestimmten Betriebssystemplattformen für das Verwaltungssystem für relationale Datenbanken (Relational Database Management System, RDBMS) von Informix. Version 7.2 enthält Scripts und Tools für alle unterstützten Verwaltungssysteme für relationale Datenbanken, einschließlich DB2. Es gibt einige Einschränkungen. Weitere Informationen finden Sie in „Bekannte Probleme und Einschränkungen“ auf Seite 302.

Die Servicestufe von DB2 OLAP Starter Kit for DB2 Universal Database Version 7.2 ist gleichbedeutend mit Programmkorrektur 2 für Hyperion Essbase 6.1 und Programmkorrektur 2 für Hyperion Integration Server 2.0.

OLAP Server-Website

Die aktuellen Tipps für die Installation und den Einsatz von DB2 OLAP Starter Kit finden Sie auf der Bibliotheksseite der Website für DB2 OLAP Server:

<http://www.ibm.com/software/data/db2/db2olap/library.html>

Unterstützte Betriebssystemservicestufen

Die Serverkomponenten von OLAP Starter Kit for Version 7.2 unterstützen die folgenden Betriebssysteme und Servicestufen:

- Windows NT 4.0-Server mit SP 5 und Windows 2000
- AIX Version 4.3.3 oder höher
- Solaris-Betriebssystem Version 2.6, 7 und 8 (Sun OS 5.6, 5.7 oder 5.8)

Die Client-Komponenten laufen unter Windows 95, Windows 98, Windows NT 4.0 SP5 und Windows 2000.

Abschließen der Installation von DB2 OLAP Starter Kit unter UNIX

Die Installation von DB2 OLAP Starter Kit verwendet im wesentlichen die Prozeduren der Installation von DB2 Universal Database für UNIX. Die Produktdateien werden vom Installationsprogramm in einem Systemverzeichnis gespeichert (für AIX: /usr/lpp/db2_07_01; für Solaris: /opt/IBMdb2/V7.1).

In der Exemplarerstellungsphase werden zwei DB2 OLAP-Verzeichnisse (essbase und is) im Ausgangsverzeichnis des Exemplarbenutzers unter sqllib erstellt. Nur ein Exemplar eines OLAP-Servers kann jeweils auf einer Maschine ausgeführt werden. Zum Abschluss dieser Konfiguration muss der Benutzer manuell das Verzeichnis is/bin so einstellen, dass es keine Programmverbindung zum Verzeichnis is/bin im System bildet. Es sollte eine Verbindung zu einem nicht schreibgeschützten Verzeichnis im Ausgangsverzeichnis des Exemplars herstellen.

Zum Abschluss der Installation für Solaris melden Sie sich mit der Exemplar-ID an, wechseln Sie in das Verzeichnis sqllib/is, und geben Sie folgende Befehle ein:

```
rm bin
mkdir bin
cd bin
```

```
ln -s /opt/IBMdb2/V7.1/is/bin/ismesg.mdb ismesg.mdb
ln -s /opt/IBMdb2/V7.1/is/bin/olapicmd olapicmd
ln -s /opt/IBMdb2/V7.1/is/bin/olapisvr olapisvr
ln -s /opt/IBMdb2/V7.1/is/bin/essbase.mdb essbase.mdb
ln -s /opt/IBMdb2/V7.1/is/bin/libolapams.so libolapams.so
```

Konfigurieren von ODBC für OLAP Starter Kit

IBM DB2 OLAP Starter Kit 7.2 erfordert eine Datei ODBC.ini zur Verarbeitung von ODBC-Verbindungen (Open Database Connectivity) von OLAP Integration Server zu der relationalen Datenquelle und zum OLAP-Metadatenkatalog.

- Auf Windows-Systemen befindet sich diese Datei in der Registrierdatenbank unter HKEY_LOCAL_MACHINE/SOFTWARE/ODBC. Verwenden Sie ODBC Data Source Administrator, um Informationen zur Art zu speichern, mit der Sie die Verbindung zu einer relationalen Datenquelle herstellen.
- Auf UNIX-Systemen erstellt das Installationsprogramm ein Modell für die Datei `odbc.ini`. Editieren Sie die Datei mit Ihrem bevorzugten Editor, um Informationen zur Art zu speichern, mit der Sie die Verbindung zu einer relationalen Datenquelle herstellen.

Die Datei ODBC.ini ist in ODBC-Softwarepaketen verfügbar und ist in der Microsoft Office-Software enthalten. Besuchen Sie die folgende Website, wenn Sie weitere Informationen zu Anwendungen benötigen, die ODBC-Treiber oder den ODBC-Administrator installieren:

<http://support.microsoft.com/support/kb/articles/Q113/1/08.asp>.

Anweisung für Oracle-Benutzer auf AIX-Maschinen: Für die Konfiguration von ODBC für Oracle müssen Sie die Datei `ODBC.ini` aktualisieren, damit sie auf die MERANT 3.6-Treiber zeigt.

In Version 7.2 verwaltet OLAP Starter Kit ODBC-Verbindungen zu der relationalen Datenquelle und zum OLAP-Metadatenkatalog. OLAP Starter Kit verwendet ODBC-Treiber unter Windows NT 4.0, Windows 2000, AIX und Solaris, um diese ODBC-Verbindungen bieten zu können.

- Datenbank-Client unter DB2 Universal Database Version 6: ODBC-Treiber für DB2 Version 6 unter Windows NT 4.0 SP5 oder Windows 2000, AIX 4.3.3 und Solaris Operating System 2.6, 7 oder 8 (Sun OS 5.6, 5.7 oder 5.8)
- Datenbank-Client unter DB2 Universal Database 7.1: ODBC-Treiber für DB2 Version 7 unter Windows NT 4.0 SP5 oder Windows 2000, AIX 4.3.3 and Solaris Operating System 2.6, 7 oder 8 (Sun OS 5.6, 5.7 oder 5.8)
- Datenbank-Client unter Oracle 8.04 und 8i SQL*Net 8.0: ODBC-Treiber für MERANT 3.6 unter Windows NT 4.0 SP5 oder Windows 2000, AIX 4.3.3 und Solaris Operating System 2.6, 7 oder 8 (Sun OS 5.6, 5.7 oder 5.8)
- MS SQL Server 6.5.201 (kein Datenbank-Client erforderlich): ODBC-Treiber für MS SQL Server 6.5 unter Windows NT 4.0 SP5 oder Windows 2000
- MS SQL Server 7.0 (kein Datenbank-Client erforderlich): ODBC-Treiber für MS SQL Server 7.0 unter Windows NT 4.0 SP5 oder Windows 2000

Konfigurieren von Datenquellen auf UNIX-Systemen

Unter AIX und Solaris müssen Sie die Umgebungsvariablen für ODBC manuell festlegen und die Datei `odbc.ini` editieren, um die relationalen Datenquellen und den OLAP-Metadatenkatalog zu konfigurieren. Stellen Sie sicher, dass Sie die Datei `odbc.ini` editieren, wenn Sie einen neuen Treiber oder eine neue Datenquelle hinzufügen oder wenn Sie den Treiber oder die Datenquelle ändern.

Konfigurieren der ODBC-Umgebungsvariablen

Auf UNIX-Systemen müssen Sie die Umgebungsvariablen festlegen, um den Zugriff auf ODBC-Kernkomponenten zu aktivieren. Die Shell-Skripts `is.sh` und `is.csh`, die die erforderlichen Variablen festlegen, befinden sich im Starter Kit-Ausgangsverzeichnis. Sie müssen eines der Skripts ausführen, bevor Sie ODBC verwenden, um sich mit Datenquellen zu verbinden. Sie sollten diese Skripts in das Anmelde-Skript des Benutzernamens integrieren, mit dem Sie OLAP Starter Kit ausführen.

Editieren der Datei `odbc.ini`

Zum Konfigurieren einer Datenquelle in der Datei `odbc.ini` müssen Sie einen Namen und eine Beschreibung für die ODBC-Datenquelle hinzufügen und den ODBC-Treiberpfad, -Dateinamen und andere Treibereinstellungen in einem anderen Abschnitt angeben, den Sie für den Datenquellennamen erstellen. Das Installationsprogramm installiert eine Beispieldatei `odbc.ini` in dem Verzeichnis `ISHOME`. Die Datei enthält generische ODBC-Verbindungen und Konfigurationsinformationen für unterstützte ODBC-Treiber. Verwenden Sie diese Datei als Ausgangspunkt, um die ODBC-Treiber, die Sie verwenden, den relationalen Datenquellen und dem OLAP-Metadatenkatalog zuzuordnen.

Wenn Sie eine andere Datei als `odbc.ini` verwenden, stellen Sie sicher, dass die Umgebungsvariable `ODBCINI` auf den Namen der Datei gesetzt ist, die Sie verwenden.

Hinzufügen einer Datenquelle zu der Datei `odbc.ini`

1. Öffnen Sie auf dem System, das die OLAP Starter Kit-Server ausführt, die Datei `odbc.ini`, indem Sie einen Texteditor wie z. B. `vi` verwenden.
2. Suchen Sie den Abschnitt, der mit `[ODBC Data Sources]` beginnt, und fügen Sie eine neue Zeile mit dem Namen und der Beschreibung der Datenquelle hinzu. Zum Beispiel: `meindat=Datenquelle für Analyse`. Der Name der Datenquelle sollte mit dem Namen der Datenbank in dem RDBMS übereinstimmen, um Verwechslungen zu vermeiden.
3. Fügen Sie der Datei einen neuen Abschnitt hinzu, indem Sie eine neue Zeile mit dem Namen der neuen Datenquelle in eckigen Klammern erstellen. Zum Beispiel: `[meindat]`.
4. Fügen Sie in den Zeilen nach dem Datenquellennamen den vollständigen Pfad und Dateinamen des ODBC-Treibers, der für diese Datenquelle erforderlich ist, und alle anderen erforderlichen ODBC-Treiberinformationen hinzu. Verwenden Sie die Beispiele, die in den folgenden Abschnitten gezeigt werden, als Richtlinie für die Zuordnung der Datenquelle in Ihrem Verwaltungssystem für relationale Datenbanken. Stellen Sie sicher, dass die ODBC-Treiberdatei tatsächlich an der Adresse vorhanden ist, die Sie in der Einstellung `Driver=` angegeben haben.
5. Wenn Sie das Editieren der Datei `odbc.ini` beendet haben, speichern Sie die Datei, und verlassen Sie den Texteditor.

Beispiel für ODBC-Einstellungen für DB2

Das folgende Beispiel zeigt, wie Sie die Datei `odbc.ini` editieren können, um mit DB2 Universal Database Version 6.1 unter AIX eine Verbindung zu der relationalen Datenquelle `db2data` herzustellen, wobei Sie einen nativen IBM DB2-ODBC-Treiber verwenden. Verwenden Sie in dem Editor `vi` den Befehl `$ODBCINI`, um die Datei `odbc.ini` zu editieren, und fügen Sie die folgenden Anweisungen ein:

```
[ODBC Data Sources]
db2data=DB2 Source Data on AIX
...
```

```
[db2data]
Driver=/home/db2inst1/sqllib/lib/db2.o
Description=DB2-Datenquelle - AIX, nativ
```

Beispiel für ODBC-Einstellungen für Oracle

Es folgt ein Beispiel, wie Sie die Datei `odbc.ini` editieren können, um mit Oracle Version 8 (unter Solaris) eine Verbindung zu der relationalen Datenquelle `oradata` herzustellen, wobei Sie einen MERANT Version 3.6-ODBC-Treiber verwenden. In diesem Beispiel werden die Parameter `anmelde-id` und `kennwort` mit den tatsächlichen Werten überschrieben, die als Benutzername und Kennwort in OLAP Starter Kit verwendet werden.

```
[ODBC Data Sources]
oradata=Oracle8 Source Data on Solaris
...
[mein-oracle] Driver=
/export/home/users/dkendric/is200/odbc/lib/ARor815.so
Description=meine Oracle-Quelle
```

Konfigurieren des OLAP-Metadatenkatalogs auf UNIX-Systemen

Das Konfigurieren eines OLAP-Metadatenkatalogs unter AIX und Solaris ist ähnlich wie das Konfigurieren einer Datenquelle. Fügen Sie der Datei `odbc.ini` für die OLAP-Metadatenkatalogdatenbank einen Datenquellennamen und einen Abschnitt hinzu, wie unter „Editieren der Datei `odbc.ini`“ auf Seite 295 beschrieben wird. Es sind keine anderen Änderungen erforderlich.

Sie müssen eine OLAP-Metadatenkatalogdatenbank in einem unterstützten Verwaltungssystem für relationale Datenbanken erstellen, bevor Sie sie als ODBC-Datenquelle konfigurieren.

Es folgt ein Beispiel, wie Sie die Datei `odbc.ini` editieren können, um mit DB2 Version 6.1 (unter Solaris) eine Verbindung zum OLAP-Metadatenkatalog „TBC_MD“ herzustellen, wobei Sie einen nativen ODBC-Treiber verwenden:

```
[ODBC Data Sources]
ocd6a5a=db2 v6
...
[ocd6a5a]
Driver=/home/db2inst1/sqllib/lib/db2.o
Description=DB2
```

Konfigurieren von Datenquellen auf Windows-Systemen

Zum Konfigurieren einer relationalen Datenquelle auf Windows NT- oder Windows 2000-Systemen müssen Sie den ODBC-Administrator starten und danach eine Verbindung zu der Datenquelle erstellen, die Sie für die Erstellung der OLAP-Modelle und OLAP-Metamodellstrukturen verwenden wollen. Führen Sie den ODBC-Administrator über die Windows-Systemsteuerung aus. Das folgende Beispiel erstellt eine DB2-Datenquelle. Die Dialogfenster für andere Verwaltungssysteme für relationale Datenbanken können anders aussehen.

Führen Sie die folgenden Schritte aus, um eine relationale Datenquelle mit dem ODBC-Administrator zu konfigurieren:

1. Öffnen Sie das Fenster „Systemsteuerung“ auf dem Windows-Desktop.
2. Führen Sie in dem Fenster „Systemsteuerung“ einen der folgenden Schritte aus:
 - a. Klicken Sie unter Windows NT das Symbol **ODBC** doppelt an, um das Dialogfenster „ODBC Data Source Administrator“ zu öffnen.

- b. Klicken Sie unter Windows 2000 das Symbol **Verwaltung** doppelt an, klicken Sie danach das Symbol **Datenquellen (ODBC)** an, um das Dialogfenster „ODBC-Datenquellenadministrator“ zu öffnen.
3. Klicken Sie die Indexzunge **System-DSN** im Dialogfenster „ODBC-Datenquellenadministrator“ an.
4. Klicken Sie **Hinzufügen** an, um das Dialogfenster „Neue Datenquelle erstellen“ zu öffnen.
5. Wählen Sie einen passenden Treiber aus dem Listenfenster mit Treibern im Dialogfenster „Neue Datenquelle erstellen“ des ODBC-Administrators aus, wie z. B. IBM DB2 ODBC Driver, und klicken Sie **Fertig stellen** an, um das Dialogfenster „IBM DB2-ODBC-Treiber - Hinzufügen“ zu öffnen.
6. Wählen Sie im Dialogfenster „IBM DB2-ODBC-Treiber - Hinzufügen“ aus der Dropdown-Liste **Aliasname der Datenbank** den Namen der Datenbank für Ihre relationalen Quelldaten (z. B. TBC in derselben Anwendung) aus.
7. Geben Sie eine optionale Beschreibung in das Textfenster **Beschreibung** ein, die angibt, wie Sie diesen Treiber verwenden, und klicken Sie **Hinzufügen** an. Geben sie z. B. die folgenden Wörter ein, um die Datenbank „Mein Geschäft“ zu beschreiben:
Kunden, Produkte, Märkte

Sie können Folgendes eingeben, um die Beispielanwendungsdatenbank zu beschreiben:

Relationale Beispieldatenquelle

Die Beschreibungen helfen dabei, die verfügbaren Datenquellen für Ihre Auswahl anzugeben, wenn Sie eine Verbindung vom OLAP Starter Kit-Desktop herstellen.

8. Klicken Sie **OK** an, um zum Dialogfenster „ODBC-Datenquellenadministrator“ zurückzukehren. Der von Ihnen eingegebene Datenquellename und der zugeordnete Treiber werden in dem Listenfenster **Systemdatenquellen** auf der Indexzunge **System-DSN** angezeigt.

Gehen Sie wie folgt vor, um Konfigurationsinformationen zu einer Datenquelle zu editieren:

1. Wählen Sie den Datenquellennamen aus, und klicken Sie **Konfigurieren** an, um das Dialogfenster „IBM DB2-ODBC-Treiber - Hinzufügen“ zu öffnen.
2. Korrigieren Sie alle Informationen, die Sie ändern wollen.
3. Klicken Sie zweimal **OK** an, um das Fenster zu verlassen.

Konfigurieren des OLAP-Metadatenkatalog auf Windows-Systemen

Starten Sie den ODBC-Administrator, und erstellen Sie danach eine Verbindung zu der Datenquelle, die die OLAP-Metadatenkatalogdatenbank enthält, um einen OLAP-Metadatenkatalog unter Windows NT oder Windows 2000 zu konfigurieren.

Das folgende Beispiel erstellt eine DB2-Datenquelle. Dialogfenster für andere Verwaltungssysteme für relationale Datenbanken können anders aussehen. Führen Sie die folgenden Schritte aus, um eine Datenquelle für den OLAP-Metadatenkatalog zu erstellen:

1. Öffnen Sie das Fenster „Systemsteuerung“ auf dem Desktop.
2. Führen Sie in dem Fenster „Systemsteuerung“ einen der folgenden Schritte aus:

- a. Klicken Sie unter Windows NT das Symbol **ODBC** doppelt an, um das Dialogfenster „ODBC-Datenquellenadministrator“ zu öffnen.
- b. Klicken Sie unter Windows 2000 das Symbol **Verwaltung** doppelt an, klicken Sie danach das Symbol **Datenquellen (ODBC)** an, um das Dialogfenster „ODBC-Datenquellenadministrator“ zu öffnen.
3. Klicken Sie die Indexzunge **System-DSN** im Dialogfenster „ODBC-Datenquellenadministrator“ an.
4. Klicken Sie **Hinzufügen** an, um das Dialogfenster „Neue Datenquelle erstellen“ zu öffnen.
5. Wählen Sie einen passenden Treiber aus dem Listenfenster mit Treibern im Dialogfenster „Neue Datenquelle erstellen“ des ODBC-Administrators aus, wie z. B. IBM DB2 ODBC Driver, und klicken Sie **Fertig stellen** an, um das Dialogfenster „IBM DB2-ODBC-Treiber - Hinzufügen“ zu öffnen.
6. Wählen Sie im Dialogfenster „ODBC IBM DB2 Driver - Hinzufügen“ aus der Dropdown-Liste **Aliasname der Datenbank** den Namen der Datenbank für Ihren OLAP-Metadatenkatalog (z. B. TBC_MD in derselben Anwendung) aus. Der Name der ausgewählten Datenbank wird automatisch in dem Textfenster **Name der Datenquelle** angezeigt.
7. Wenn Sie den Namen einer Datenquelle ändern wollen, wählen Sie den Namen aus, der im Textfenster **Datenquellennamen** angezeigt wird, geben Sie einen neuen Namen ein, der angibt, wie Sie diesen Treiber verwenden und klicken Sie **Hinzufügen** an. Sie können z. B. den folgenden Namen eingeben, um anzugeben, dass Sie den Treiber verwenden, um eine Verbindung zum ersten OLAP-Metadatenkatalog herzustellen:

Erster OLAP-Katalog

Sie würden den folgenden Namen eingeben, um anzugeben, dass Sie eine Verbindung zu der OLAP-Metadatenkatalogdatenbank der Beispielanwendung herstellen:

TBC_MD

8. Geben Sie eine Beschreibung in das Textfenster **Beschreibung** ein, die angibt, wie Sie diesen Treiber verwenden. Sie können z. B. Folgendes eingeben, um den OLAP-Metadatenkatalog zu beschreiben:

Meine ersten Modelle und
Metamodellstrukturen

Sie können Folgendes eingeben, um die OLAP-Metadatenkatalogdatenbank der Beispielanwendung zu beschreiben:

Beispielmodelle und
-Metamodellstrukturen

Die Beschreibungen helfen Ihnen, den Katalog anzugeben, den Sie auswählen wollen, wenn Sie eine Verbindung vom OLAP Starter Kit-Desktop zum OLAP-Metadatenkatalog herstellen.

9. Klicken Sie **OK** an, um zum Dialogfenster „ODBC-Datenquellenadministrator“ zurückzukehren. Der von Ihnen eingegebene Datenquellennamen und der zugeordnete Treiber werden in dem Listenfenster **Systemdatenquellen** auf der Indexzunge **System-DSN** angezeigt.

Gehen Sie wie folgt vor, um Konfigurationsinformationen zu einer Datenquelle zu editieren:

1. Wählen Sie den Datenquellennamen aus, und klicken Sie **Konfigurieren** an, um das Dialogfenster „IBM DB2-ODBC-Treiber - Hinzufügen“ zu öffnen.

2. Korrigieren Sie alle Informationen, die Sie ändern wollen.
3. Klicken Sie zweimal **OK** an, um das Fenster zu verlassen.

Vorgehensweise nach dem Konfigurieren einer Datenquelle

Nachdem Sie die relationale Datenquelle und den OLAP-Metadatenkatalog konfiguriert haben, können Sie über OLAP Starter Kit eine Verbindung zu ihnen herstellen. Sie können dann OLAP-Modelle und -Metamodellstrukturen erstellen, ändern und speichern.

Der SQL-Server-ODBC-Treiber kann während eines Aufrufs einer SQL-Serverdatenbank das Zeitlimit überschreiten. Starten Sie den Aufruf erneut, wenn die Datenbank nicht ausgelastet ist. Erhöhen Sie den Zeitraum für das Zeitlimit, um dieses Problem zu vermeiden. Weitere Informationen zu dem Treiber, den Sie verwenden, finden Sie in der ODBC-Dokumentation.

Weitere Informationen zu ODBC-Verbindungsproblemen und Lösungen finden Sie im Handbuch *OLAP Integration Server System Administrator's Guide*.

Anmelden über OLAP Starter Kit Desktop

Wenn Sie OLAP Starter Kit Desktop für die Erstellung von OLAP-Modellen und -Metamodellstrukturen verwenden wollen, müssen Sie von der Client-Software eine Verbindung zu zwei Serverkomponenten herstellen: DB2 OLAP Integration Server und DB2 OLAP Server. Im Anmeldedialog werden Sie nach den Informationen gefragt, die Desktop für die Herstellung von Verbindungen zu diesen beiden Servern benötigt. Geben Sie auf der linken Seite des Dialogs Informationen zu DB2 OLAP Integration Server ein. Auf der rechten Seite geben Sie Informationen zu DB2 OLAP Server ein.

Gehen Sie wie folgt vor, um eine Verbindung zu DB2 OLAP Integration Server herzustellen:

- **Server:** Geben Sie den Host-Namen oder die IP-Adresse Ihres Integration Server ein. Wenn der Integration Server auf derselben Workstation installiert ist wie Desktop, sind typische Werte "localhost" oder "127.0.0.1".
- **OLAP Metadata Catalog:** Wenn Sie eine Verbindung zu OLAP Integration Server herstellen, müssen Sie auch einen Metadatenkatalog angeben. OLAP Integration Server speichert Informationen zu den von Ihnen erstellten OLAP-Modellen und -Metamodellstrukturen in einer relationalen Datenbank, dem so genannten Metadatenkatalog. Diese relationale Datenbank muss für ODBC registriert sein. Die Katalogdatenbank enthält eine spezielle Gruppe relationaler Tabellen, die OLAP Integration Server erkennt. Im Anmeldedialog können Sie einen Integration Server angeben und dann das Feld **OLAP Metadata Catalog** erweitern, um eine Liste der ODBC-Datenquellennamen anzuzeigen, die OLAP Integration Server bekannt sind. Wählen Sie eine ODBC-Datenbank aus, die die Metadaten-Katalogtabellen enthält.
- **User Name and Password:** OLAP Integration Server stellt mit dem in diesem Feld angegebenen Benutzernamen und Kennwort eine Verbindung zum Metadatenkatalog her. Dies ist ein Anmeldekonto auf dem Server (nicht auf dem Client, sofern Server und Client nicht auf derselben Maschine aktiv sind). Der Benutzername muss der Benutzer sein, der den OLAP-Metadatenkatalog erstellt hat. Andernfalls kann OLAP Integration Server die relationalen Tabellen in der Katalogdatenbank nicht finden, da die Tabellenschemennamen anders sind.

Die Informationen zu DB2 OLAP Server sind optional. Die Eingabefelder auf der rechten Seite des Anmeldedialogs können deshalb leer bleiben. Einige Operationen in Desktop und in Administration Manager setzen eine Verbindung zu DB2 OLAP Server voraus. Wenn Sie diese Felder leer lassen, zeigt Desktop den Anmeldedialog erneut an, wenn Integration Server eine Verbindung zu DB2 OLAP Server herstellen muss, um eine angeforderte Operation abzuschließen. Es empfiehlt sich daher, die Felder zu DB2 OLAP Server im Anmeldedialog immer auszufüllen.

Gehen Sie wie folgt vor, um eine Verbindung zu DB2 OLAP Server herzustellen:

- **Server:** Geben Sie den Host-Namen oder die IP-Adresse Ihres DB2 OLAP Server ein. Wenn Sie OLAP Starter Kit ausführen, entspricht OLAP Server dem Integration Server. Wenn Integration Server und OLAP Server auf unterschiedlichen Hosts installiert sind, geben Sie den Host-Namen oder eine IP-Adresse ein, die auf OLAP Integration Server definiert ist.
- **User Name and Password:** OLAP Integration Server stellt mit dem in diesem Feld angegebenen Benutzernamen und Kennwort eine Verbindung zu DB2 OLAP Server her. Benutzername und Kennwort müssen bereits in DB2 OLAP Server definiert sein. OLAP Server verwaltet seine eigenen Benutzernamen und Kennwörter separat vom Betriebssystem des Hosts.

Beispiel für Starter Kit-Anmeldung

Im folgenden Beispiel wird davon ausgegangen, dass Sie bei der Installation von OLAP Starter Kit die OLAP-Beispieldatenbank erstellt, *db2admin* als Administrator-Benutzer-ID und *kennwort* als Administratorkennwort ausgewählt haben.

- Für OLAP Integration Server: Der Server heißt *localhost*, der OLAP-Metadatenkatalog ist *TBC_MD*, der Benutzername lautet *db2admin* und das Kennwort ist *kennwort*.
- Für DB2 OLAP Server: Der Server heißt *localhost* und der Benutzername lautet *db2admin*.

Manuelles Erstellen und Konfigurieren der Beispieldatenbanken für OLAP Starter Kit

Die Beispieldatenbanken werden bei der Installation von OLAP Starter Kit automatisch erstellt. Die folgenden Anweisungen beschreiben die manuelle Einrichtung der Katalog- und Beispieldatenbanken, sofern dies notwendig sein sollte.

1. Öffnen Sie in Windows die Befehlszentrale mit **Start→Programme→DB2 für Windows NT→Befehlsfenster**.
2. Erstellen Sie die Produktionskatalogdatenbank:
 - a. Geben Sie `db2 create db OLAP_CAT` ein.
 - b. Geben Sie `db2 connect to OLAP_CAT` ein.
3. Erstellen Sie Tabellen in der Datenbank:
 - a. Wechseln Sie zu `\SQLLIB\IS\ocscript\ocdb2.sql`.
 - b. Geben Sie `db2 -tf ocdb2.sql` ein.
4. Erstellen Sie die Beispielquellendatenbank:
 - a. Geben Sie `db2 connect reset` ein.
 - b. Geben Sie `db2 create db TBC` ein.
 - c. Geben Sie `db2 connect to TBC` ein.
5. Erstellen Sie Tabellen in der Datenbank:
 - a. Wechseln Sie in das Verzeichnis `\SQLLIB\IS\samples\`.

- b. Kopieren Sie `tbddb2.sql` in `\SQLLIB\samples\db2samp1\tbc`.
 - c. Kopieren Sie `lddb2.sql` in `\SQLLIB\samples\db2samp1\tbc`.
 - d. Wechseln Sie in das Verzeichnis `\SQLLIB\samples\db2samp1\tbc`.
 - e. Geben Sie `db2 -tf tbddb2.sql` ein.
 - f. Geben Sie `db2 -vf lddb2.sql` ein, um die Beispielquelldaten in die Tabellen zu laden.
6. Erstellen Sie die Beispielkatalogdatenbank:
- a. Geben Sie `db2 connect reset` ein.
 - b. Geben Sie `db2 create db TBC_MD` ein.
 - c. Geben Sie `db2 connect to TBC_MD` ein.
7. Erstellen Sie Tabellen in der Datenbank:
- a. Wechseln Sie in das Verzeichnis `\SQLLIB\IS\samples\tbc_md`.
 - b. Kopieren Sie `ocdb2.sql` in `\SQLLIB\samples\db2samp1\tbcmd`.
 - c. Kopieren Sie `lcdb2.sql` in `\SQLLIB\samples\db2samp1\tbcmd`.
 - d. Wechseln Sie in das Verzeichnis `\SQLLIB\samples\db2samp1\tbcmd`.
 - e. Geben Sie `db2 -tf ocdb2.sql` ein.
 - f. Geben Sie `db2 -vf lcdb2.sql` ein, um die Beispielmetadaten in die Tabellen zu laden.
8. Konfigurieren Sie ODBC für TBC_MD, TBC und OLAP_CAT:
- a. Öffnen Sie die Systemsteuerung von Windows NT mit **Start—>Einstellungen—>Systemsteuerung**.
 - b. Wählen Sie **ODBC** (oder ODBC-Datenquellen) aus der Liste aus.
 - c. Wählen Sie die Indexzunge **System-DSN** aus.
 - d. Klicken Sie **Hinzufügen** an. Das Fenster **Neue Datenquelle erstellen** wird geöffnet.
 - e. Wählen Sie **IBMDB2ODBC Driver** aus der Liste aus.
 - f. Klicken Sie **Fertig stellen** an. Das Fenster **IBM DB2-ODBC-Treiber - Hinzufügen** wird geöffnet.
 - g. Geben Sie den Namen der Datenquelle (OLAP_CAT) in das Feld **Name der Datenquelle** ein.
 - h. Geben Sie den Aliasnamen in das Feld **Aliasname der Datenbank** ein oder klicken Sie den Pfeil nach unten an und wählen Sie **OLAP_CAT** aus der Liste aus.
 - i. Klicken Sie **OK** an.
 - j. Wiederholen Sie diese Schritte für die Datenbanken **TBC_MD** und **TBC**.

Migrieren von Anwendungen nach OLAP Starter Kit Version 7.2

Das Installationsprogramm installiert die Beispielanwendungen, -datenbanken und -datendateien von OLAP Starter Kit nicht erneut. Ihre vorhandenen Anwendungen und Datenbanken werden in keiner Weise beeinträchtigt. Es ist trotzdem immer empfehlenswert, Ihre Anwendungen und Datenbanken vor einer Installation zu sichern.

Ihre Anwendungen werden automatisch nach Version 7.2 migriert, wenn Sie sie öffnen.

Bekannte Probleme und Einschränkungen

In diesem Abschnitt werden bekannte Einschränkungen für DB2 OLAP Starter Kit aufgelistet.

Kompatibilität von Informix RDBMS mit Merant-Treibern für Windows-Plattformen

Damit die Merant-Treiber für Windows-Plattformen mit Informix RDBMS funktionieren, müssen die folgenden zwei Einträge zu der Anweisung PATH hinzugefügt werden:

- C:\Informix
- C:\Informix\bin

Beide Einträge müssen sich am Anfang von PATH befinden.

Mögliche Inkonsistenz zwischen Dimensionen in OLAP-Modellen und zugeordneten Metamodellstrukturen

Unter bestimmten Bedingungen können Sie in einer Metamodellstruktur eine Dimension erstellen, die keine entsprechende Dimension in dem OLAP-Modell hat. Dies kann in dem folgenden Szenario passieren:

1. Erstellen Sie ein neues OLAP-Modell, und speichern Sie es.
2. Erstellen Sie eine Modellstruktur, die auf dem Modell basiert, speichern Sie die Modellstruktur aber nicht.
3. Kehren Sie zu dem OLAP-Modell zurück, und löschen Sie eine Dimension, auf der eine der Dimensionen der Metamodellstruktur basiert.
4. Kehren Sie zu der Metamodellstruktur zurück, speichern und schließen Sie sie, und öffnen Sie sie erneut. Die Metamodellstruktur enthält eine Dimension, die keine entsprechende Dimension in dem OLAP-Modell hat.

OLAP Starter Kit kann nicht zwischen einer inkonsistenten Dimension, die auf diese Weise erstellt worden ist, und einer benutzerdefinierten Dimension in einer Metamodellstruktur unterscheiden. Deshalb wird die inkonsistente Dimension in der Metamodellstruktur angezeigt, aber die Metamodellstruktur behandelt sie als benutzerdefinierte Dimension, weil keine entsprechende Dimension in dem OLAP-Modell vorhanden ist.

Auf Windows 2000-Plattformen verursacht die Einstellung der Umgebungsvariable für TMP Fehlschläge beim Laden von Elementen und Daten

Wegen unterschiedlichen Einstellungen für die Standardsystem- und Benutzerumgebungsvariable TMP zwischen Windows 2000 und Windows NT schlagen Ladevorgänge von Elementen und Daten fehl, wenn OLAP Starter Kit auf Windows 2000-Plattformen ausgeführt wird. Die resultierende Fehlermeldung teilt den Benutzern mit, dass die temporäre Datei nicht erstellt werden konnte. Sie können diese Einschränkung unter Windows 2000 umgehen, indem Sie die folgenden Schritte ausführen:

1. Erstellen Sie ein Verzeichnis mit dem Namen C:\TEMP
2. Setzen Sie die Umgebungsvariable für das System und den Benutzer auf TMP=C:\TEMP

Die Installation von ODBC ersetzt vorhandene Merant-Treiber nicht

Die vorhandenen Merant 3.6-ODBC-Treiber werden mit dieser Installation nicht aktualisiert. Wenn Sie von OLAP Starter Kit Version 7.1, FixPak 2 oder früher aktualisieren, sollten Sie weiterhin die vorher installierten ODBC-Treiber verwenden.

Verwenden von Merant-Informix-ODBC-Treibern auf UNIX-Plattformen

Sie müssen einen der folgenden Schritte ausführen, um die Merant-Informix-ODBC-Treiber auf UNIX-Plattformen zu verwenden:

- Bevor Sie OLAP Starter Kit starten, setzen Sie die Umgebungsvariable LANG auf "en_US". Geben Sie für Korn-Shell z. B. Folgendes ein:

```
export LANG='en_US'
```

Setzen Sie diese Variable jedesmal, wenn Sie OLAP Starter Kit starten.

- Wenn Ihre Umgebungsvariable LANG bereits auf einen anderen Wert gesetzt ist, erstellen Sie nach der Installation die folgende symbolische Verbindung:

```
ln -s $ISHOME/locale/en_US $ISHOME/locale/$LANG
```

Mischen von Servicestufen von OLAP-Clients und -Servern

IBM empfiehlt, dass Sie sowohl die Client- wie auch die Serverkomponenten von DB2 OLAP Starter Kit auf derselben Version und FixPak-Stufe halten. In einigen Situationen können Sie allerdings unterschiedliche Servicestufen von Client- und Serverkomponenten mischen:

Verwenden von Clients und Servern mit unterschiedlichen Servicestufen innerhalb einer Version

IBM unterstützt die Verwendung von neueren Clients mit älteren Servern nicht und rät davon ab. Sie könnten allerdings ältere Clients mit neueren Servern verwenden, obwohl IBM das nicht unterstützt. Sie könnten einige Probleme bekommen. Zum Beispiel:

- Nachrichten vom Server könnten fehlerhaft sein. Sie können dieses Problem umgehen, indem Sie die Datei message.MDB auf dem Client aktualisieren, damit sie mit der Stufe des Servers übereinstimmt.
- Neue Serverfunktionen arbeiten nicht. Der Client, der Server oder beide könnten fehlschlagen, wenn Sie versuchen, eine neue Funktion zu verwenden.
- Der Client könnte sich nicht richtig mit dem Server verbinden.

Verwenden von mehreren Servern mit einem einzelnen Client innerhalb einer Version

Wenn Sie einen Client mit mehreren OLAP-Servern auf unterschiedlichen Maschinen oder Betriebssystemen verbinden müssen, empfiehlt IBM, dass alle dieselbe Version und Servicestufe haben sollten. Ihr Client sollte mindestens dieselbe Stufe wie die niedrigste Serverstufe haben. Wenn Sie auf Probleme stoßen, müssen Sie möglicherweise unterschiedliche Client-Maschinen verwenden, um mit dem jeweiligen Host übereinzustimmen oder alle Clients und Server auf dieselbe Servicestufe aktualisieren.

Mischen von Clients und Servern von unterschiedlichen Versionen

IBM unterstützt die Verwendung von OLAP Starter Kit-Clients und -Servern von Version 7.1 mit Clients und Servern von Version 7.2 nicht. Wenn IBM OLAP-Produkte auf einen neueren Versionsstand aktualisiert werden, gibt es häufig Netzwerkaktualisierungen und Änderungen des Datenformats, die erfordern, dass Client und Server denselben Versionsstand haben.

Mischen von IBM-Produkten (DB2 OLAP Starter Kit) mit Hyperion-Produkten (Hyperion Essbase und Hyperion Integration Server)

IBM unterstützt das Mischen von OLAP Starter Kit-Clients und -Servern von IBM mit Clients und Servern von Hyperion Solutions nicht. Es gibt einige unterschiedliche Funktionen, die Probleme verursachen könnten, obwohl das Mischen dieser Komponenten in einigen Situationen funktionieren könnte.

Fehlende EQD-Dateien des OLAP-Tabellenkalkulations-Add-In

Im DB2 OLAP Starter Kit verfügt das Tabellenkalkulations-Add-In über eine Komponente namens Query Designer (EQD). Das Onlinehilfemenü für EQD enthält einen Druckknopf für das Lernprogramm, der nichts anzeigt. Das Material, das in den EQD-Lernprogrammen angezeigt werden sollte, ist ein Teil von Kapitel 2 aus dem *OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Excel* und dem *OLAP Tabellenkalkulations-Add-In Benutzerhandbuch für Lotus 1-2-3*. Alle Informationen im EQD-Lernprogramm sind als HTML-Versionen dieser Handbücher in **Information - Unterstützung** und als PDF-Versionen verfügbar.

Information Catalog Manager Systemverwaltung

Initialisierungsprogramm von Information Catalog Manager

Das Initialisierungsdienstprogramm von Information Catalog Manager ermöglicht Ihnen jetzt das Anhängen einer SQL-Anweisung an das Ende der Anweisung CREATE TABLE mit Hilfe des folgenden Befehls:

```
CREATEIC \DBTYPE dbtyp \DGNAME  
icm-name \USERID benutzer-id  
\PASSWORD kennwort  
\KA1 benutzer-id \TABOPT  
"verzeichnis:\tabopt.file"
```

Sie können das Schlüsselwort TABOPT im Dienstprogramm CREATEIC von dem Verzeichnis aus angeben, in dem DB2 installiert ist. Der Wert nach dem Schlüsselwort TABOPT ist der Dateiname *tabopt.file* mit vollständigem Pfad. Wenn der Verzeichnisname Leerzeichen enthält, schließen Sie den Namen in Anführungszeichen ein. Der Inhalt der Datei *tabopt.file* muss Informationen enthalten, die an die Anweisung CREATE TABLE angehängt werden. Sie können beliebige der unten aufgeführten SQL-Anweisungen zum Schreiben in die Datei *tabopt.file* verwenden. Das ICM-Dienstprogramm liest diese Datei und hängt sie dann an die Anweisung CREATE TABLE an.

Tabelle 20. SQL-Anweisungen

IN MYTABLESPACE	Erstellt eine Tabelle mit Daten in MYTABLESPACE
DATA CAPTURE CHANGES	Erstellt eine Tabelle und protokolliert SQL-Änderungen in einem erweiterten Format
IN ACCOUNTING INDEX IN ACCOUNT_IDX	Erstellt eine Tabelle mit Daten in ACCOUNTING und Index in ACCOUNT_IDX

Die Maximalgröße der Nachrichtenkomponente beträgt 1000 Single-Byte-Zeichen.

Diese neue Funktionalität ist nur bei Windows- und UNIX-Systemen verfügbar.

Lizenzfragen

Es kann vorkommen, dass die folgende Nachricht angezeigt wird:

FLG0083E:

Sie haben keine gültige Lizenz für das Dienstprogramm zur Initialisierung von IBM Information Catalog Manager. Wenden Sie sich an Ihren Softwarehändler vor Ort oder an einen IBM Vertriebsbeauftragten.

Sie müssen DB2 Warehouse Manager oder IBM DB2 OLAP Server erwerben und Information Catalog Manager installieren, das das Informationskatalog-Initialisierungsprogramm enthält.

Installationsfragen

Wenn Sie DB2 Warehouse Manager oder IBM DB2 OLAP Server und danach eine andere Komponente von Information Catalog Manager Administrator (über die CD-ROM von DB2 Universal Database) auf derselben Workstation installiert haben, wurde das Informationskatalog-Initialisierungsprogramm möglicherweise über-

schrieben. Suchen Sie in diesem Fall im Verzeichnis `\sql11ib\bin` die Dateien `createic.bak` und `flgnmwcr.bak`, und benennen Sie sie in `createic.exe` bzw. `flgnmwcr.exe` um.

Wenn Sie weitere Information Catalog Manager-Komponenten von DB2 Universal Database aus installieren, müssen sich die Komponenten auf einer anderen Workstation befinden als der, auf der DB2 Warehouse Manager installiert ist. Weitere Informationen finden Sie in Kapitel 3 des DB2-Handbuchs *Data Warehouse Manager Installation*.

Zugreifen auf Informationskataloge von DB2 Version 5 mit Information Catalog Manager von DB2 Version 7

Die Subkomponenten von Information Catalog Manager von DB2 Version 7 werden bei der Installation von DB2 Version 7 so konfiguriert, dass der Zugriff auf Informationskataloge unterstützt wird, die in Datenbanken von DB2 Version 6 und Version 7 gespeichert sind. Sie können die Konfiguration der Subkomponenten so modifizieren, dass Sie auch auf Informationskataloge in Datenbanken von DB2 Version 5 zugreifen können. Die Subkomponenten von Information Catalog Manager von DB2 Version 7 bieten keine Unterstützung für den Zugriff auf Daten von DB2 Version 2 oder sonstigen früheren Versionen.

Gehen Sie wie folgt vor, um "Informationskatalog - Administrator", "Informationskatalog - Benutzer" und das Dienstprogramm zur Initialisierung des Informationskatalogs einzurichten, damit Sie auf Informationskataloge zugreifen können, die in Datenbanken von DB2 Version 5 gespeichert sind:

1. Installieren Sie DB2 Connect Enterprise Edition Version 6 auf einer Workstation, auf welcher DB2 Version 7 Information Catalog Manager nicht installiert ist.

DB2 Connect Enterprise Edition ist im Lieferumfang von DB2 Universal Database Enterprise Edition und DB2 Universal Database Enterprise - Extended Edition enthalten. Wenn Version 6 eines dieser beiden Produkte installiert ist, brauchen Sie DB2 Connect nicht separat zu installieren.

Einschränkung: Sie können auf einer Windows NT- oder OS/2-Workstation nicht mehrere DB2-Versionen installieren. Sie können DB2 Connect auf einer anderen Windows NT-, OS/2- oder UNIX-Workstation installieren.

2. Konfigurieren Sie Information Catalog Manager und DB2 Connect Version 6 für den Zugriff auf Daten von DB2 Version 5. Weitere Informationen finden Sie im *DB2 Connect Benutzerhandbuch*. Die folgenden Schritte geben eine Übersicht über die erforderlichen Schritte:
 - a. Verwenden Sie auf einem System mit DB2 Version 5 den DB2- Befehlszeilenprozessor, um die Datenbank der Version 5 zu katalogisieren, auf die Information Catalog Manager zugreifen soll.
 - b. Katalogisieren Sie auf dem DB2 Connect-System folgende Elemente mit dem DB2-Befehlszeilenprozessor:
 - Den TCP/IP-Knoten für das System mit DB2 Version 5
 - Die Datenbank für das System mit DB2 Version 5
 - Den DCS-Eintrag für das System mit DB2 Version 5
 - c. Katalogisieren Sie auf der Information Catalog Manager-Workstation folgende Elemente mit dem DB2-Befehlszeilenprozessor:
 - Den TCP/IP-Knoten für das DB2 Connect-System
 - Die Datenbank für das DB2 Connect-System

Informationen zur Katalogisierung von Datenbanken finden Sie in *DB2 Universal Database Installation und Konfiguration Ergänzung*.

3. Binden Sie im Warehouse von Information Catalog Manager das DB2-CLI-Paket an alle Datenbanken, auf die mit DB2 Connect zugegriffen werden soll.

Die folgenden DB2-Befehle sind Beispiele für das Binden von v5database, einer gedachten Datenbank von DB2 Version 5. Setzen Sie die folgenden Befehle mit dem DB2-Befehlszeilenprozessor ab. db2cli.lst und db2ajgrt befinden sich im Verzeichnis \sqllib\bnd.

```
db2 connect to v5database user benutzer-id
using kennwort
db2 bind db2ajgrt.bnd db2 bind @db2cli.lst blocking all grant public
```

Dabei gilt: *benutzer-id* ist die Benutzer-ID für v5database und *kennwort* ist das Kennwort für die Benutzer-ID.

Beim Binden von db2cli.lst an die Datenbank von DB2 Version 5 tritt ein Fehler auf. Dies liegt daran, dass große Objekte (LOBs) in dieser Konfiguration nicht unterstützt werden. Dieser Fehler hat keine Auswirkungen auf den Zugriff des Warehouse-Agenten auf die Datenbank von DB2 Version.

Für den Zugriff auf Daten von DB2 Version 5 mit DB2 Connect ist das FixPak 14 für DB2 Universal Database Version 5 erforderlich, das seit Juni 2000 verfügbar ist. Informationen dazu finden Sie unter der APAR-Nummer JR14507 in diesem FixPak.

Einrichten eines Informationskatalogs

Schritt 2 im ersten Abschnitt von Kapitel 1, "Informationskatalog konfigurieren", lautet:

```
Wenn Sie DB2 Warehouse Manager oder DB2 OLAP Server installieren, wird ein Standardinformationskatalog in DB2 Universal Database für Windows NT erstellt.
```

Diese Angabe ist nicht richtig. Sie müssen einen neuen Informationskatalog definieren. Weitere Informationen finden Sie im Abschnitt "Informationskatalog erstellen".

Austauschen von Metadaten mit anderen Produkten

In Kapitel 6, "Metadaten mit anderen Produkten austauschen", im Abschnitt "OLAP-Objekte zur Veröffentlichung angeben" gibt es eine Anweisung im zweiten Abschnitt, die folgendermaßen lautet:

```
Bei der Veröffentlichung von DB2 OLAP Integration Server-Metadaten wird eine Beziehung in Form einer Verknüpfung zwischen einer Objektart mit "Dimensionen innerhalb einer mehrdimensionalen Datenbank" im Informationskatalog und einem Tabellenobjekt in OLAP Integration Server erstellt.
```

Dieser Text muss wie folgt lauten:

```
Bei der Veröffentlichung von DB2 OLAP Integration Server-Metadaten wird eine Beziehung in Form einer Verknüpfung zwischen einer Objektart mit "Dimensionen innerhalb eines mehrdimensionalen Datenbankobjekts und eines Tabellenobjekts erstellt.
```

Dieser Abschnitt erscheint auch in Anhang C, "Zuordnung von Metadaten", im Abschnitt "Zuordnung von Metadaten zwischen Information Catalog Manager und einem OLAP-Server".

Austauschen von Metadaten mit dem Befehl "flgnxoln"

In Kapitel 6, "Metadaten mit anderen Produkten austauschen" gibt es einen Abschnitt "OLAP-Objekte zur Veröffentlichung angeben". Am Ende dieses Abschnitts ist ein Beispiel dazu, wie Sie mit dem Befehl flgnxoln OLAP Server-Metadaten in einem Informationskatalog veröffentlichen. In diesem Beispiel wird fälschlicherweise das Verzeichnis x:\Programmdateien\sqllib\logging für die Dateien db2olap.ctl und db2olap.ff angezeigt. Der Verzeichnisname muss x:\Programmdateien\sqllib\exchange lauten, wie weiter oben im Abschnitt zum Editieren der Steuerdatei beschrieben.

Austauschen von Metadata mit dem Befehl "MDISDGC"

Kapitel 6, "Metadaten mit anderen Produkten austauschen": "MDIS-konforme Metadaten in eine Befehlssprachendatei umsetzen". Sie können den Befehl MDIS-DGC nicht über die MS-DOS-Eingabeaufforderung absetzen. Sie müssen den Befehl MDISDGC über ein DB2-Befehlsfenster absetzen. Im ersten Satz des Abschnitts "Befehlssprachendatei in MDIS-konforme Metadaten umsetzen" wird auch gesagt, dass Sie den Befehl DGMDISC über die MS-DOS-Eingabeaufforderung absetzen müssen. Sie müssen den Befehl DGMDISC über ein DB2-Befehlsfenster absetzen.

Aufrufen von Programmen

Einige Beispiele im Handbuch *Information Catalog Manager Systemverwaltung* zeigen Befehle, die den Verzeichnisnamen Programmdateien enthalten. Wenn Sie ein Programm aufrufen, das Programmdateien als Teil seines Pfadnamens enthält, müssen Sie den Programmaufruf in doppelte Anführungszeichen einschließen. Anhang B, "Vordefinierte Objektarten von Information Catalog Manager", z. B. enthält ein Beispiel im Abschnitt "Informationskatalog mit vordefinierten Objektarten initialisieren". Wenn Sie das Beispiel in diesem Abschnitt verwenden, erhalten Sie einen Fehler, wenn Sie es von einer DOS-Eingabeaufforderung ausführen.

Das folgende Beispiel ist korrekt:

```
"X:Programmdateien\SQLLIB\SAMPLES\SAMPDATA\DGWDEMO"  
/T benutzer-id kennwort icm-name
```

Information Catalog Manager Programming Guide and Reference

Information Catalog Manager Reason Codes

In Appendix D: Information Catalog Manager reason codes, some text might be truncated at the far right column for the following reason codes: 31014, 32727, 32728, 32729, 32730, 32735, 32736, 32737, 33000, 37507, 37511, and 39206. If the text is truncated, please see the HTML version of the book to view the complete column.

Information Catalog Manager Benutzerhandbuch

In Kapitel 2 gibt es einen Abschnitt "Registrieren eines Server-Knotens und eines fernen Informationskatalogs". In diesem Abschnitt werden Schritte aufgelistet, die Sie von der DB2-Steuerzentrale ausführen können, bevor Sie einen fernen Informationskatalog mit Information Catalog Manager registrieren. Im letzten Absatz des Abschnitts heißt es, dass Sie nach der Durchführung einer Reihe von Schritten von der DB2-Steuerzentrale aus (Hinzufügen eines Systems, Hinzufügen eines Exemplars und Hinzufügen einer Datenbank) die Steuerzentrale schließen müssen, bevor Sie Information Catalog Manager öffnen. Diese Angabe ist nicht richtig. Es ist nicht notwendig, die Steuerzentrale vor dem Öffnen von Information Catalog Manager zu schließen.

The same correction also applies to the online help task "Registering a server node and remote information catalog", and the online help for the Register Server Node and Information Catalog window.

Information Catalog Manager: Onlinenachrichten

Nachricht FLG0260E

Der zweite Satz in der Erläuterung der Nachricht muss wie folgt lauten:

Der Fehler hat eine ROLLBACK-Operation für den Informationskatalog ausgelöst, bei dem der Fehler aufgetreten ist. Der Informationskatalog hat keinen stabilen Status, es wurden aber keine Änderungen vorgenommen.

Nachricht FLG0051E

Der zweite Listenpunkt in der Erläuterung der Nachricht muss wie folgt lauten:

Der Informationskatalog enthält zu viele Objekte oder Objektarten.

Die Antwort des Administrators muss wie folgt lauten:

Löschen Sie einige Objekte oder Objektarten aus dem aktuellen Informationskatalog mit Hilfe der Importfunktion.

Nachricht FLG0003E

Die Erläuterung der Nachricht muss wie folgt lauten:

Der Informationskatalog muss registriert werden, bevor Sie ihn verwenden können. Dieser Informationskatalog wurde möglicherweise nicht korrekt registriert.

Nachricht FLG0372E

Der erste Satz in der Erläuterung der Nachricht sollte wie folgt lauten:

Der Wert ATTACHMENT-IND wurde bei einem Objekt ignoriert, da es sich um ein Attachment-Objekt handelt.

Nachricht FLG0615E

Der zweite Satz der Nachricht muss wie folgt lauten:

Bei Information Catalog Manager ist ein unerwarteter Datenbankfehler aufgetreten oder die Bindedatei kann im aktuellen Verzeichnis oder Pfad nicht gefunden werden.

Information Catalog Manager: Onlinehilfefunktion

Fenster "Informationskatalog": In der Onlinehilfefunktion wird bei der Option "Öffnen" im Menü "Ausgewählt" fälschlicherweise der Text "Öffnet das ausgewählte Objekt" angezeigt. Der Text muss lauten: "Öffnet das Fenster 'Suche definieren'".

Information Catalog Manager für das Web

Wenn Sie einen Informationskatalog auf einem DB2 UDB für OS/390-System verwenden, ist die Suchfunktion ohne Beachtung von Groß-/Kleinschreibung nicht verfügbar. Dies gilt gleichermaßen für eine einfache und für eine erweiterte Suche. In der Onlinehilfefunktion wird nicht erklärt, dass bei allen Suchvorgängen in einem Informationskatalog von DB2 UDB für OS/390 für eine einfache Suche die Groß-/Kleinschreibung beachtet werden muss. Außerdem sind alle Objekte einer Gruppierungskategorie erweiterbar, selbst wenn keine zugrundeliegenden Objekte vorhanden ist.

DB2 Warehouse Manager Installation

Softwarevoraussetzungen für Warehouse-Umsetzungsprogramme

Java Developer's Kit (JDK) Version 1.1.8 oder später muss auf dem Datenbanksystem installiert sein, in dem Sie die Warehouse-Umsetzungsprogramme verwenden wollen.

Connector für SAP R/3

Bei der Zuordnung von Spalten aus Feldern eines SAP R/3-Geschäftsobjekts könnten einige der generierten Spaltennamen länger als 30 Zeichen sein. In diesem Fall gibt der generierte Spaltenname nur die ersten 30 Zeichen des SAP-Feldnamens wieder. Wenn der generierte Name nicht Ihren Wünschen entspricht, können Sie ihn im Notizbuch „Merkmale“ für die Tabelle ändern.

Vorbedingungen zur Installation

Legen Sie die Umgebungsvariable RFC_INI fest. Geben Sie z. B. `RFC_INI=c:\rfcapl.ini` an. Nachdem Sie diese Variable festgelegt haben, müssen Sie einen Warmstart der Maschine durchführen.

Connector für das Web

Wenn bei der Ausführung des Connector für das Web Probleme auftreten, fordert der IBM Kundendienst möglicherweise an, dass Sie einen Trace für den Connector einsenden.

Setzen Sie den Agenten-Trace in der Warehouse-Zentrale auf eine Stufe größer als 0, um den Trace für den Connector für das Web zu aktivieren. Die Trace-Datei hat den Namen `WSApid.log`. Dabei gilt Folgendes: `pid` ist die Windows-Prozess-ID für den Agenten. Die Trace-Datei wird in dem Verzeichnis `\sql11b\logging` erstellt.

Vorbedingungen zur Installation

Installieren Sie die Java-Laufzeitumgebung (run-time environment, JRE) oder die virtuelle Java-Maschine (Java virtual machine, JVM) Version 1.2.2 oder später, und legen Sie sie als Standardeinstellung fest. Fügen Sie Ihrer Systemvariable `PATH` den Pfad für Version 1.2.2 von JRE hinzu (z. B. `C:\JDKs\IBM\java12\bin;`), um eine Version von JRE als Ihre Standardeinstellung festzulegen. Nachdem Sie Ihre Standard-JRE festgelegt haben, müssen Sie einen Warmstart der Maschine durchführen. Wenn Sie kein Java installiert haben, können Sie es von der Installations-CD-ROM von Data Warehouse Connectors installieren.

Query Patroller Administration Guide

DB2 Query Patroller Client is a Separate Component

The DB2 Query Patroller client is a separate component that is not part of the DB2 Administration client. This means that it is not installed during the installation of the DB2 Administration Client, as indicated in the Query Patroller Installation Guide. Instead, the Query Patroller client must be installed separately.

The version and level of the Query Patroller client and the Query Patroller server must be the same.

Migrating from Version 6 of DB2 Query Patroller Using `dqpmigrate`

The `dqpmigrate` command must be used if the Version 7 Query Patroller Server was installed over the Version 6 Query Patroller Server. For FixPak 2 or later, you do not have to run `dqpmigrate` manually as the installation of the FixPak runs this command for you. Without using this command, the existing users defined in v6 have no EXECUTE privileges on several new stored procedures added in Version 7.

Anmerkung: `dqpmigrate.bnd` is found in the `sqllib/bnd` directory and `dqpmigrate.exe` is found in the `sqllib/bin` directory.

To use **`dqpmigrate`** manually to grant the EXECUTE privileges, perform the following after installing the FixPak:

1. Bind the `/sqllib/bnd/dqpmigrate.bnd` package file to the database where the Query Patroller server has been installed by entering the following command:
`db2 bind dqpmigrate.bnd`
2. Execute **`dqpmigrate`** by entering the following:
`dqpmigrate dbalias userid passwd`

Enabling Query Management

In the "Getting Started" chapter under "Enabling Query Management", the text should read:

You must be the owner of the data base, or you must have SYSADM, SYSCTRL, or SYSMAINT authority to set database configuration parameters.

Location of Table Space for Control Tables

In Chapter 1, System Overview, under DB2 Query Patroller Control Tables, the following text is to be added at the end of the section's first paragraph:

The table space for the DB2 Query Patroller control tables must reside in a single-node nodegroup, or DB2 Query Patroller will not function properly.

New Parameters for `dqpstart` Command

In Chapter 2, Getting Started, under Starting and Stopping DB2 Query Patroller, the following text is to be added following the last paragraph:

New Parameters for the `dqpstart` command:

RESTART parameter:

Allows the user to replace the host name and/or the node type of the specified node in the `dqnodes.cfg` file. DB2 Query Patroller will be started on this node.

Anmerkung: Before running the `DQPSTART` command with the `RESTART` parameter, ensure the following:

1. DB2 Query Patroller is already stopped on the host that is going to be replaced.
2. DB2 Query Patroller is not already running on the new host.

The syntax is as follows:

```
dqpstart nodenum node_num restart hostname server | agent | none
```

ADDNODE parameter:

Allows the user to add a new node to the `dqnodes.cfg` file. DB2 Query Patroller will be started on this node after the new node entry is added to the `dqnodes.cfg` file. The syntax is as follows:

```
dqpstart nodenum node_num addnode hostname server | agent | none
```

DROPNODE parameter:

Allows the user to drop a node from the `dqnodes.cfg` file. DB2 Query Patroller will be stopped on this node before the node entry is dropped from the `dqnodes.cfg` file. The syntax is as follows:

```
dqpstop nodenum node_num dropnode
```

New Parameter for `iwm_cmd` Command

A new `-v` parameter has been added to the `iwm_cmd` command to allow the user to recover the status of the jobs that were running on the node specified. Only jobs on an inactive node are allowed to be recovered. This command should be issued when there is a node failure and there are some jobs running on that node or being cancelled at the time. Jobs that were in "Running" state will be resubmitted and set back to "Queued" state. Jobs that were in "Cancelling" state will be set to "Cancelled" state.

The partial syntax is as follows:

```
▶▶ iwm_cmd [ -u user_id [ -p password ] ] -v node_id_to_recover ▶▶
```

`node_id_to_recover`

Specifies the node on which the jobs are to be recovered.

New Registry Variable: DQP_RECOVERY_INTERVAL

There is a new registry variable called DQP_RECOVERY_INTERVAL which is used to set the interval of time in minutes that the iwm_scheduler searches for recovery files. The default is 60 minutes.

Starting Query Administrator

In the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, instructions are provided for starting QueryAdministrator from the Start menu on Windows. The first step provides the following text:

If you are using Windows, you can select DB2 Query Patroller -> QueryAdministrator from the IBM DB2 program group.

The text should read:

DB2 Query Patroller -> QueryAdmin.

User Administration

In the "User Administration" section of the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, the definition for the Maximum Elapsed Time parameter indicates that if the value is set to 0 or -1, the query will always run to completion. This parameter cannot be set to a negative value. The text should indicate that if the value is set to 0, the query will always run to completion.

The Max Queries parameter specifies the maximum number of jobs that the DB2 Query Patroller will run simultaneously. Max Queries must be an integer within the range of 0 to 32767.

Creating a Job Queue

In the "Job Queue Administration" section of the "Using QueryAdministrator to Administer DB2 Query Patroller" chapter, the screen capture in the steps for "Creating a Job Queue" should be displayed after the second step. The Information about new Job Queue window opens once you click **New** on the Job Queue Administration page of the QueryAdministrator tool.

References to the Job Queues page or the Job Queues tab should read Job Queue Administration page and Job Queue Administration tab, respectively.

Using the Command Line Interface

For a user with User authority on the DB2 Query Patroller system to submit a query and have a result table created, the user may require CREATETAB authority on the database. The user does not require CREATETAB authority on the database if the DQP_RES_TBLSPC profile variable is left unset, or if the DQP_RES_TBLSPC profile variable is set to the name of the default table space. The creation of the result tables will succeed in this case because users have the authority to create tables in the default table space.

Query Enabler Notes

- When using third-party query tools that use a keyset cursor, queries will not be intercepted. In order for Query Enabler to intercept these queries, you must modify the `db2cli.ini` file to include:

```
[common]
DisableKeySetCursor=1
```

- For AIX clients, please ensure that the environment variable `LIBPATH` is not set. Library `libXext.a`, shipped with the JDK, is not compatible with the library in the `/usr/lib/X11` subdirectory. This will cause problems with the Query Enabler GUI.

DB2 Query Patroller Tracker may Return a Blank Column Page

FixPak 3 includes a fix for the DB2 Query Patroller Tracker. The Tracker will now correctly report queries which hit no columns. An example of such a query is "SELECT COUNT(*) FROM ...". Since this kind of query does not hit any column in the table, the Tracker will present a blank page for the column page. This blank column page is not a defect.

Query Patroller and Replication Tools

Query Patroller Version 7 will intercept the queries of the replication tools (`asnapply`, `asnccp`, `djra` and `analyze`) and cause these tools to malfunction. A workaround is to disable dynamic query management when running these tools.

Appendix B. Troubleshooting DB2 Query Patroller Clients

In Appendix B, Troubleshooting DB2 Query Patroller Clients, section: Common Query Enabler Problems, problem #2, the text of the first bullet is replaced with:

Ensure that the path setting includes `jre`.

Teil 6. Anwendungsentwicklung

Administrative API Reference	327	Chapter 12. Building Solaris Applications	354
db2ArchiveLog (new API)	327	SPARCompiler C++	354
db2ArchiveLog	327	Chapter 13. Building Applications for Windows	
db2ConvMonStream	329	32-bit Operating Systems	354
db2DatabasePing (new API)	330	VisualAge C++ Version 4.0	354
db2DatabasePing - Ping Database	330	Application Development Guide	355
db2HistData	331	Chapter 2. Coding a DB2 Application	355
db2HistoryOpenScan	333	Activating the IBM DB2 Universal Database	
db2XaGetInfo (new API)	333	Project and Tool Add-ins for Microsoft Visual	
db2XaGetInfo - Get Information for Resource		C++	355
Manager	333	Chapter 6. Common DB2 Application Techniques	355
db2XaListIndTrans (new API that supercedes sqlx-		Generating Sequential Values	355
phqr)	334	Controlling Sequence Behavior	356
db2XaListIndTrans - List Indoubt Transactions	334	Improving Performance with Sequence	
db2GetSnapshot - Get Snapshot	337	Objects	357
Forget Log Record	337	Comparing Sequence Objects and Identity	
sqlaintp - Get Error Message	338	Columns	357
sqlbctcq - Close Tablespace Container Query	338	Chapter 7. Stored Procedures	358
sqlubkp - Backup Database	338	DECIMAL Type Fails in Linux Java Routines	358
sqlureot - Reorganize Table	338	Using Cursors in Recursive Stored Procedures	358
sqlurestore - Restore Database.	338	Writing OLE Automation Stored Procedures	358
Documentation Error Regarding AIX Extended		Chapter 12. Working with Complex Objects: User-	
Shared Memory Support (EXTSHM).	339	Defined Structured Types	359
SQLFUPD	339	Inserting Structured Type Attributes Into	
locklist	339	Columns	359
SQLEDBDESC	340	Chapter 13. Using Large Objects (LOBs)	359
SQLFUPD Documentation Error	340	Large object (LOBs) support in federated data-	
Application Building Guide	341	base systems	359
Chapter 1. Introduction	341	How DB2 retrieves LOBs	359
Supported Software	341	How applications can use LOB locators	360
Sample Programs	343	Restrictions on LOBs	360
Chapter 3. General Information for Building DB2		Mappings between LOB and non-LOB data	
Applications.	343	types	360
Build Files, Makefiles, and Error-checking Utili-		Tuning the system.	361
ties	343	Part 5. DB2 Programming Considerations	361
Chapter 4. Building Java Applets and Applications	344	IBM DB2 OLE DB Provider.	361
Setting the Environment.	344	Chapter 20. Programming in C and C++	361
JDK Level on OS/2	345	C/C++ Types for Stored Procedures, Functions,	
Java2 on HP-UX	345	and Methods	361
Chapter 5. Building SQL Procedures.	346	Chapter 21. Programming in Java	363
Setting the SQL Procedures Environment	346	Java Method Signature in PARAMETER STYLE	
Setting the Compiler Environment Variables	346	JAVA Procedures and Functions	363
Customizing the Compilation Command	347	Connecting to the JDBC Applet Server	363
Retaining Intermediate Files	351	Appendix B. Sample Programs	363
Backup and Restore	351	CLI Guide and Reference.	365
Creating SQL Procedures	351	Binding Database Utilities Using the Run-Time Cli-	
Calling Stored Procedures	352	ent	365
Distributing Compiled SQL Procedures.	352	Using Static SQL in CLI Applications	365
Chapter 7. Building HP-UX Applications.	353	Limitations of JDBC/ODBC/CLI Static Profiling	365
HP-UX C.	353	ADT Transforms	366
HP-UX C++	353	Chapter 3. Using Advanced Features	366
Chapter 9. Building OS/2 Applications	353	Writing Multi-Threaded Applications	366
VisualAge C++ for OS/2 Version 4.0	353	Scrollable Cursors	367
Chapter 10. Building PTX Applications	353		
ptx/C++	353		

Server-side Scrollable Cursor Support for OS/390	367	MQRECEIVE	397
Using ompound SQL	368	MQSEND	398
Using Stored Procedures	369	MQSUBSCRIBE	400
Writing a Stored Procedure in CLI	369	MQUNSUBSCRIBE	401
CLI Stored Procedures and Autobinding	369	MULTIPLY_ALT	402
Chapter 4. Configuring CLI/ODBC and Running Sample Applications	369	REC2XML	403
Configuration Keywords	369	ROUND	408
Chapter 5. DB2 CLI Functions	369	WEEK_ISO	409
SQLBindFileToParam - Bind LOB File Reference to LOB Parameter	369	Table Functions	410
SQLNextResult - Associate Next Result Set with Another Statement Handle	370	MQREADALL	410
Purpose	370	MQRECEIVEALL	412
Syntax	370	Procedures	414
Function Arguments	370	GET_ROUTINE_SAR	414
Usage	370	PUT_ROUTINE_SAR	415
Return Codes	371	Chapter 5. Queries	416
Diagnostics	371	select-statement/syntax diagram	416
Restrictions	371	select-statement/fetch-first-clause	417
References	371	Chapter 6. SQL Statements	417
Appendix D. Extended Scalar Functions	372	Update of the Partitioning Key Now Supported	417
Date and Time Functions	372	Statement: ALTER TABLE	417
Appendix K. Using the DB2 CLI/ODBC/JDBC Trace Facility	372	Statement: CREATE TABLE	417
		Statement: DECLARE GLOBAL TEMPORARY TABLE PARTITIONING KEY (column-name,...)	417
		Statement: UPDATE	417
		Larger Index Keys for Unicode Databases	418
Fehlernachrichten	373	ALTER TABLE	418
Abrufen der Hilfe zu Nachrichten und SQLSTATE-Werten	373	CREATE INDEX	418
Änderung der Zuordnung von SQLCODE-Werten in DB2 Connect	373	CREATE TABLE	418
Neue und geänderte Nachrichten	373	ALTER SEQUENCE	418
CLI-Nachrichten (Call Level Interface)	373	ALTER SEQUENCE	418
DB2-Nachrichten	373	ALTER TABLE	422
DBI-Nachrichten	374	Compound SQL (Embedded)	425
Nachrichten der Data Warehouse-Zentrale	374	Compound Statement (Dynamic)	425
SQL-Nachrichten	375	Compound Statement (Dynamic)	425
Korrigierte SQLSTATE-Werte	377	CREATE FUNCTION (Source or Template)	429
		CREATE FUNCTION (SQL Scalar, Table or Row)	430
SQL Reference	379	CREATE METHOD	432
SQL-Referenz in einer PDF-Datei verfügbar	379	CREATE METHOD	432
Chapter 3. Language Elements	379	CREATE SEQUENCE	434
Naming conventions and Implicit Object Name Qualifications	379	CREATE SEQUENCE	434
DATALINK Assignments	379	CREATE TRIGGER	438
Expressions	379	CREATE TRIGGER	438
Syntax Diagram	379	CREATE WRAPPER	440
OLAP Functions	380	DECLARE CURSOR	440
Sequence Reference	382	DELETE	441
Chapter 4. Functions	385	DROP	442
Enabling the New Functions and Procedures	385	GRANT (Sequence Privileges)	443
Scalar Functions	386	GRANT (Sequence Privileges)	443
ABS or ABSVAL	386	INSERT	444
DECRYPT_BIN and DECRYPT_CHAR	386	SELECT INTO	445
ENCRYPT	387	SET ENCRYPTION PASSWORD	446
GETHINT	389	SET ENCRYPTION PASSWORD	446
IDENTITY_VAL_LOCAL	390	SET transition-variable	447
LCASE and UCASE (Unicode)	394	SET Variable	447
MQPUBLISH	394	UPDATE	449
MQREAD	396	Chapter 7. SQL Procedures now called Chapter 7. SQL Control Statements	450
		SQL Procedure Statement	450
		SQL Procedure Statement	450

FOR	452
FOR	452
Compound Statement changes to Compound Statement (Procedure)	454
RETURN	456
RETURN	456
SIGNAL	458
SIGNAL	458
Appendix A. SQL Limits	461
Appendix D. Catalog Views	461
SYSCAT.SEQUENCES	461

Versetzen von Daten Dienstprogramme und Referenz	482
Anhang C, Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD	482

DB2 Stored Procedure Builder	463
Java 1.2-Unterstützung für DB2 Stored Procedure Builder	463
Fernes Debug von gespeicherten DB2-Prozeduren	464
Erstellen von SQL-Prozeduren auf Windows-, OS/2- oder UNIX-Plattformen.	464
Verwenden von DB2 Stored Procedure Builder auf der Solaris-Plattform	464
Bekannte Probleme und Einschränkungen	464
Using DB2 Stored Procedure Builder with Traditional Chinese Locale	465
UNIX-Installationen (AIX, Sun Solaris, Linux) und Stored Procedure Builder	465
Erstellen von gespeicherten SQL-Prozeduren unter OS/390	465
Debug von gespeicherten SQL-Prozeduren . . .	465
Exportieren gespeicherter Java-Prozeduren . . .	466
Einfügen gespeicherter Prozeduren unter OS/390	466
Einstellen von Erstellungsoptionen für gespeicherte SQL-Prozeduren auf einem Workstation-Server . .	466
Automatisches Aktualisieren des WLM- Adressraums für gespeicherte Prozeduren unter OS/390	467
Entwicklung von gespeicherten Java-Prozeduren unter OS/390	467
Aufbau einer benutzerdefinierten Funktion in einer DB2-Tabelle für MQ Series und OLE DB	467

Unicode-Aktualisierungen	469
Einführung	469
DB2-Unicode-Datenbanken und -Anwendungen	469
Aktualisierung der Dokumentation	471
SQL Reference	471
Chapter 3 Language Elements	471
Promotion of Data Types	471
Casting Between Data Types	472
Assignments and Comparisons	473
Rules for Result Data Types	474
Rules for String Conversions	475
Expressions	475
Predicates	475
Chapter 4 Functions	475
Scalar Functions	475
CLI Guide and Reference	476
Chapter 3. Using Advanced Features	476
Writing a DB2 CLI Unicode Application . . .	476
Appendix C. DB2 CLI and ODBC	481
ODBC Unicode Applications	481

Administrative API Reference

db2ArchiveLog (new API)

db2ArchiveLog

Closes and truncates the active log file for a recoverable database. If user exit is enabled, issues an archive request.

Berechtigung

One of the following:

- *sysadm*
- *sysctrl*
- *sysmaint*
- *dbadm*

Erforderliche Verbindung

This API automatically establishes a connection to the specified database. If a connection to the specified database already exists, the API will return an error.

Anmerkungen

db2ApiDf.h

Syntax für C-API

```
/* File: db2ApiDf.h */
/* API: Archive Active Log */
SQL_API_RC SQL_API_FN
db2ArchiveLog (
    db2UInt32 version,
    void *pDB2ArchiveLogStruct,
    struct sqlca * pSqlca);

typedef struct
{
    char          *piDatabaseAlias;
    char          *piUserName;
    char          *piPassword;
    db2UInt16     iAllNodeFlag;
    db2UInt16     iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2UInt32     iOptions;
}
```

db2ArchiveLog - Archive Active Log

Syntax für generische API

```
/* File: db2ApiDf.h */
/* API: Archive Active Log */
SQL_API_RC SQL_API_FN
db2gArchiveLog (
    db2UInt32 version,
    void *pDB2ArchiveLogStruct,
    struct sqlca * pSqlca);

typedef struct
{
    db2UInt32          iAliasLen;
    db2UInt32          iUserNameLen;
    db2UInt32          iPasswordLen;
    char               *piDatabaseAlias;
    char               *piUserName;
    char               *piPassword;
    db2UInt16          iAllNodeFlag;
    db2UInt16          iNumNodes;
    SQL_PDB_NODE_TYPE *piNodeList;
    db2UInt32          iOptions;
}
```

API-Parameter

version

Input. Specifies the version and release level of the variable passed in as the second parameter, *pDB2ArchiveLogStruct*.

pDB2ArchiveLogStruct

Input. A pointer to the *db2ArchiveLogStruct* structure.

pSqlca

Output. A pointer to the *sqlca* structure.

iAliasLen

Input. A 4-byte unsigned integer representing the length in bytes of the database alias.

iUserNameLen

A 4-byte unsigned integer representing the length in bytes of the user name. Set to zero if no user name is used.

iPasswordLen

Input. A 4-byte unsigned integer representing the length in bytes of the password. Set to zero if no password is used.

piDatabaseAlias

Input. A string containing the database alias (as cataloged in the system database directory) of the database for which the active log is to be archived.

piUserName

Input. A string containing the user name to be used when attempting a connection.

piPassword

Input. A string containing the password to be used when attempting a connection.

iAllNodeFlag

MPP only. Input. Flag indicating whether the operation should apply to all nodes listed in the *db2nodes.cfg* file. Valid values are:

db2ArchiveLog - Archive Active Log

DB2ARCHIVELOG_NODE_LIST

Apply to nodes in a node list that is passed in *piNodeList*.

DB2ARCHIVELOG_ALL_NODES

Apply to all nodes. *piNodeList* should be NULL. This is the default value.

DB2ARCHIVELOG_ALL_EXCEPT

Apply to all nodes except those in the node list passed in *piNodeList*.

iNumNodes

MPP only. Input. Specifies the number of nodes in the *piNodeList* array.

piNodeList

MPP only. Input. A pointer to an array of node numbers against which to apply the archive log operation.

iOptions

Input. Reserved for future use.

db2ConvMonStream

In the Usage Notes, the structure for the snapshot variable datastream type SQLM_ELM_SUBSECTION should be *sqlm_subsection*.

db2DatabasePing (new API)

db2DatabasePing - Ping Database

Tests the network response time of the underlying connectivity between a client and a database server. This API can be used by an application when a host database server is accessed via DB2 Connect either directly or through a gateway.

Berechtigung

None

Erforderliche Verbindung

Database

Anmerkungen

db2ApiDf.h

Syntax für C-API

```
/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */

SQL_API_RC SQL_API_FN
db2DatabasePing (
    db2Uint32 versionNumber,
    void      *pParmStruct,
    struct sqlca * pSqlca);
/* ... */

typedef SQL_STRUCTURE db2DatabasePingStruct
{
    char          iDbAlias[SQL_ALIAS_SZ + 1];
    db2Uint16     iNumIterations;
    db2Uint32     *poElapsedTime;
}
```

Syntax für generische API

```
/* File: db2ApiDf.h */
/* API: Ping Database */
/* ... */

SQL_API_RC SQL_API_FN
db2gDatabasePing (
    db2Uint32 versionNumber,
    void      *pParmStruct,
    struct sqlca * pSqlca);
/* ... */

typedef SQL_STRUCTURE db2gDatabasePingStruct
{
    db2Uint16     iDbAliasLength;
    char          iDbAlias[SQL_ALIAS_SZ];
    db2Uint16     iNumIterations;
    db2Uint32     *poElapsedTime;
}
```

API-Parameter**versionNumber**

Input. Version and release of the DB2 Universal Database or DB2 Connect product that the application is using.

Anmerkung: Constant db2Version710 or higher should be used for DB2 Version 7.1 or higher.

iDbAliasLength

Input. Length of the database alias name.

Anmerkung: This parameter is not currently used. It is reserved for future use.

iDbAlias

Input. Database alias name.

Anmerkung: This parameter is not currently used. It is reserved for future use.

iNumIterations

Input. Number of test request iterations. The value must be between 1 and 32767 inclusive.

poElapsedTime

Output. A pointer to an array of 32-bit integers where the number of elements is equal to iNumIterations. Each element in the array will contain the elapsed time in microseconds for one test request iteration.

Anmerkung: The application is responsible for allocating the memory for this array prior to calling this API.

pSqlca

Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

Hinweise

A database connection must exist before invoking this API, otherwise an error will result.

This function can also be invoked using the PING command. For a description of this command, see the *Command Reference*.

db2HistData

The following entries should be added to Table 11. Fields in the db2HistData Structure:

Field Name	Data Type	Description
oOperation	char	See Table 12.
oOptype	char	See Table 13.

The following table will be added following Table 11.

db2ArchiveLog - Archive Active Log

Table 12. Valid event values for oOperation in the db2HistData Structure

Value	Description	C Definition	COBOL/FORTRAN Definition
A	add tablespace	DB2HISTORY_OP_ADD_TABLESPACE	DB2HIST_OP_ADD_TABLESPACE
B	backup	DB2HISTORY_OP_BACKUP	DB2HIST_OP_BACKUP
C	load-copy	DB2HISTORY_OP_LOAD_COPY	DB2HIST_OP_LOAD_COPY
D	dropped table	DB2HISTORY_OP_DROPPED_TABLE	DB2HIST_OP_DROPPED_TABLE
F	roll forward	DB2HISTORY_OP_ROLLFWD	DB2HIST_OP_ROLLFWD
G	reorganize table	DB2HISTORY_OP_REORG	DB2HIST_OP_REORG
L	load	DB2HISTORY_OP_LOAD	DB2HIST_OP_LOAD
N	rename tablespace	DB2HISTORY_OP_REN_TABLESPACE	DB2HIST_OP_REN_TABLESPACE
O	drop tablespace	DB2HISTORY_OP_DROP_TABLESPACE	DB2HIST_OP_DROP_TABLESPACE
Q	quiesce	DB2HISTORY_OP_QUIESCE	DB2HIST_OP_QUIESCE
R	restore	DB2HISTORY_OP_RESTORE	DB2HIST_OP_RESTORE
S	run statistics	DB2HISTORY_OP_RUNSTATS	DB2HIST_OP_RUNSTATS
T	alter tablespace	DB2HISTORY_OP_ALT_TABLESPACE	DB2HIST_OP_ALT_TBS
U	unload	DB2HISTORY_OP_UNLOAD	DB2HIST_OP_UNLOAD

The following table will also be added.

Table 13. Valid oOptype values db2HistData Structure

oOperation	oOptype	Description	C/COBOL/FORTRAN Definition
B	F	Offline	DB2HISTORY_OPTYPE_OFFLINE
	N	Online	DB2HISTORY_OPTYPE_ONLINE
	I	Incremental offline	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	Incremental online	DB2HISTORY_OPTYPE_INCR_ONLINE
	D	Delta offline	DB2HISTORY_OPTYPE_DELTA_OFFLINE
	E	Delta online	DB2HISTORY_OPTYPE_DELTA_ONLIN
F	E	End of log	DB2HISTORY_OPTYPE_EOL
	P	Point in time	DB2HISTORY_OPTYPE_PIT
L	I	Insert	DB2HISTORY_OPTYPE_INSERT
	R	Replace	DB2HISTORY_OPTYPE_REPLACE
Q	S	Quiesce share	DB2HISTORY_OPTYPE_SHARE
	U	Quiesce update	DB2HISTORY_OPTYPE_UPDATE
	X	Quiesce exclusive	DB2HISTORY_OPTYPE_EXCL
	Z	Quiesce reset	DB2HISTORY_OPTYPE_RESET
R	F	Offline	DB2HISTORY_OPTYPE_OFFLINE
	N	Online	DB2HISTORY_OPTYPE_ONLINE
	I	Incremental offline	DB2HISTORY_OPTYPE_INCR_OFFLINE
	O	Incremental online	DB2HISTORY_OPTYPE_INCR_ONLINE
T	C	Add containers	DB2HISTORY_OPTYPE_ADD_CONT
	R	Rebalance	DB2HISTORY_OPTYPE_REB

db2HistoryOpenScan

The following value will be added to the **iCallerAction** parameter.

DB2HISTORY_LIST_CRT_TABLESPACE

Select only the CREATE TABLESPACE and DROP TABLESPACE records that pass the other filters.

db2XaGetInfo (new API)

db2XaGetInfo - Get Information for Resource Manager

Extracts information for a particular resource manager once an xa_open call has been made.

Berechtigung

None

Erforderliche Verbindung

Database

Anmerkungen

sqlxa.h

Syntax für C-API

```

/* File: sqlxa.h */
/* API: Get Information for Resource Manager */
/* ... */
SQL_API_RC SQL_API_FN
db2XaGetInfo (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaGetInfoStruct
{
    db2int32 iRmid;
    struct sqlca oLastSqlca;
} db2XaGetInfoStruct;

```

API-Parameter

versionNumber

Input. Specifies the version and release level of the structure passed in as the second parameter, *pParmStruct*.

pParmStruct

Input. A pointer to the *db2XaGetInfoStruct* structure.

pSqlca

Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

iRmid Input. Specifies the resource manager for which information is required.

oLastSqlca

Output. Contains the *sqlca* for the last XA API call.

Anmerkung: Only the *sqlca* that resulted from the last failing XA API can be retrieved.

db2XaListIndTrans (new API that supercedes sqlxphqr)

db2XaListIndTrans - List Indoubt Transactions

Provides a list of all indoubt transactions for the currently connected database.

Bereich

This API affects only the node on which it is issued.

Berechtigung

One of the following:

- *sysadm*
- *dbadm*

Erforderliche Verbindung

Database

Anmerkungen

db2ApiDf.h

Syntax für C-API

```
/* File: db2ApiDf.h */
/* API: List Indoubt Transactions */
/* ... */
SQL_API_RC SQL_API_FN
db2XaListIndTrans (
    db2Uint32 versionNumber,
    void * pParmStruct,
    struct sqlca * pSqlca);

typedef SQL_STRUCTURE db2XaListIndTransStruct
{
    db2XaRecoverStruct * piIndoubtData;
    db2Uint32             iIndoubtDataLen;
    db2Uint32             oNumIndoubtsReturned;
    db2Uint32             oNumIndoubtsTotal;
    db2Uint32             oReqBufferLen;
} db2XaListIndTransStruct;

typedef SQL_STRUCTURE db2XaRecoverStruct
{
    sqluint32             timestamp;
    SQLXA_XID             xid;
    char                  dbalias[SQLXA_DBNAME_SZ];
    char                  applid[SQLXA_APPLID_SZ];
    char                  sequence_no[SQLXA_SEQ_SZ];
    char                  auth_id[SQL_USERID_SZ];
    char                  log_full;
    char                  connected;
    char                  indoubt_status;
    char                  originator;
    char                  reserved[8];
} db2XaRecoverStruct;
```

API-Parameter**versionNumber**

Input. Specifies the version and release level of the structure passed in as the second parameter, *pParmStruct*.

pParmStruct

Input. A pointer to the *db2XaListIndTransStruct* structure.

pSqlca

Output. A pointer to the *sqlca* structure. For more information about this structure, see the *Administrative API Reference*.

piIndoubtData

Input. A pointer to the application supplied buffer where indoubt data will be returned. The indoubt data is in *db2XaRecoverStruct* format. The application can traverse the list of indoubt transactions by using the size of the *db2XaRecoverStruct* structure, starting at the address provided by this parameter.

If the value is NULL, DB2 will calculate the size of the buffer required and return this value in *oReqBufferLen*. *oNumIndoubtsTotal* will contain the total number of indoubt transactions. The application may allocate the required buffer size and issue the API again.

oNumIndoubtsReturned

Output. The number of indoubt transaction records returned in the buffer specified by *pIndoubtData*.

oNumIndoubtsTotal

Output. The Total number of indoubt transaction records available at the time of API invocation. If the *piIndoubtData* buffer is too small to contain all the records, *oNumIndoubtsTotal* will be greater than the total for *oNumIndoubtsReturned*. The application may reissue the API in order to obtain all records.

Anmerkung: This number may change between API invocations as a result of automatic or heuristic indoubt transaction resynchronisation, or as a result of other transactions entering the indoubt state.

oReqBufferLen

Output. Required buffer length to hold all indoubt transaction records at the time of API invocation. The application can use this value to determine the required buffer size by calling the API with *pIndoubtData* set to NULL. This value can then be used to allocate the required buffer, and the API can be issued with *pIndoubtData* set to the address of the allocated buffer.

Anmerkung: The required buffer size may change between API invocations as a result of automatic or heuristic indoubt transaction resynchronisation, or as a result of other transactions entering the indoubt state. The application may allocate a larger buffer to account for this.

db2ArchiveLog - Archive Active Log

timestamp

Output. Specifies the time when the transaction entered the indoubt state.

xid

Output. Specifies the XA identifier assigned by the transaction manager to uniquely identify a global transaction.

dbalias

Output. Specifies the alias of the database where the indoubt transaction is found.

applid

Output. Specifies the application identifier assigned by the database manager for this transaction.

sequence_no

Output. Specifies the sequence number assigned by the database manager as an extension to the *applid*.

auth_id

Output. Specifies the authorization ID of the user who ran the transaction.

log_full

Output. Indicates whether or not this transaction caused a log full condition. Valid values are:

SQLXA_TRUE

This indoubt transaction caused a log full condition.

SQLXA_FALSE

This indoubt transaction did not cause a log full condition.

connected

Output. Indicates whether or not the application is connected. Valid values are:

SQLXA_TRUE

The transaction is undergoing normal syncpoint processing, and is waiting for the second phase of the two-phase commit.

SQLXA_FALSE

The transaction was left indoubt by an earlier failure, and is now waiting for resynchronisation from the transaction manager.

indoubt_status

Output. Indicates the status of this indoubt transaction. Valid values are:

SQLXA_TS_PREP

The transaction is prepared. The *connected* parameter can be used to determine whether the transaction is waiting for the second phase of normal commit processing or whether an error occurred and resynchronisation with the transaction manager is required.

SQLXA_TS_HCOM

The transaction has been heuristically committed.

SQLXA_TS_HROL

The transaction has been heuristically rolled back.

SQLXA_TS_MACK

The transaction is missing commit acknowledgement from a node in a partitioned database.

SQLXA_TS_END

The transaction has ended at this database. This transaction may be re-activated, committed, or rolled back at a later time. It is also possible that the transaction manager encountered an error and the transaction will not be completed. If this is the case, this transaction requires heuristic actions, because it may be holding locks and preventing other applications from accessing data.

Hinweise

A typical application will perform the following steps after setting the current connection to the database or to the partitioned database coordinator node:

1. Call **db2XaListIndTrans** with *piIndoubtData* set to NULL. This will return values in *oReqBufferLen* and *oNumIndoubtsTotal*.
2. Use the returned value in *oReqBufferLen* to allocate a buffer. This buffer may not be large enough if there are additional indoubt transactions because the initial invocation of this API to obtain *oReqBufferLen*. The application may provide a buffer larger than *oReqBufferLen*.
3. Determine if all indoubt transaction records have been obtained. This can be done by comparing *oNumIndoubtsReturned* to *oNumIndoubtTotal*. If *oNumIndoubtsTotal* is greater than *oNumIndoubtsReturned*, the application can repeat the above steps.

Siehe auch

"sqlxhfrg - Forget Transaction Status", "sqlxphcm - Commit an Indoubt Transaction", and "sqlxphrl - Roll Back an Indoubt Transaction" in the *Administrative API Reference*.

db2GetSnapshot – Get Snapshot

The syntax for the db2GetSnapshot API should be as follows:

```
int db2GetSnapshot( unsigned char version;
db2GetSnapshotData *data,
struct sqlca *sqlca);
```

The parameters described in data are:

```
typedef struct db2GetSnapshotData{
    sqlma *piSqlmaData;
    sqlm_collected *poCollectedData
    void *poBuffer;
    db2uint32 iVersion;
    db2int32 iBufferSize;
    db2uint8 iStoreResult;
    db2uint16 iNodeNumber;
    db2uint32 *poOutputFormat;
}db2GetSnapshotData;
```

Forget Log Record

The following information will be added to Appendix F following the MPP Subordinator Prepare section.

This log record is written after a rollback of indoubt transactions or after a commit of two-phase commit. The log record is written to mark the end of the transaction and releases any log resources held. In order for the transaction to be forgotten, it must be in a heuristically completed state.

Tabelle 21. Forget Log Record Structure

Description	Type	Offset (Bytes)
Log header	LogManagerLogRecordHeader	0(20)
time	sqluint64	20(8)
<i>Total Length: 28 bytes</i>		

sqlaintp – Get Error Message

The following usage note is to be added to the description of this API:

In a multi-threaded application, sqlaintp must be attached to a valid context; otherwise, the message text for SQLCODE -1445 cannot be obtained.

sqlbctcq – Close Tablespace Container Query

Load is not a valid Authorization level for this API.

sqlubkp – Backup Database

For the **BackupType** parameter the SQLUB_FULL value will be replaced by the SQLUB_DB. A backup of all tablespaces in the database will be taken.

To support the new incremental backup functionality the SQLUB_INCREMENTAL and SQLUB_DELTA parameters will also be added. An incremental backup image is a copy of all database data which has changed since the most recent successful, full backup. A delta backup image is a copy of all database data that has changed since the most recent successful backup of any type

sqlureot – Reorganize Table

The following sentence will be added to the Usage Notes:

REORGANIZE TABLE cannot use an index that is based on an index extension.

sqlurestore – Restore Database

For the **RestoreType** parameter the SQLUD_FULL value will be replaced by the SQLUD_DB. A restore of all table spaces in the database will be taken. This will be run offline.

To support the new incremental restore functionality the SQLUD_INCREMENTAL parameter will also be added.

An incremental backup image is a copy of all database data which has changed since the most recent successful full backup.

Documentation Error Regarding AIX Extended Shared Memory Support (EXTSHM)

In "Appendix E. Threaded Applications with Concurrent Access", Note 2 should now read:

2. By default, AIX does not permit 32-bit applications to attach to more than 11 shared memory segments per process, of which a maximum of 10 can be used for local DB2 connections.

To use EXTSHM with DB2, do the following:

In client sessions:

```
export EXTSHM=ON
```

When starting DB2:

```
export EXTSHM=ON
db2set DB2ENVLIST=EXTSHM
db2start
```

On EEE, also add the following lines to sqllib/db2profile:

```
EXTSHM=ON
export EXTSHM
```

SQLFUPD

locklist

The name of the token has changed from SQLF_DBTN_LOCKLIST to SQLF_DBTN_LOCK_LIST. The locklist parameter has been changed from a SMALLINT to a 64-bit unsigned INTEGER. The following addition should be made to the table of Updatable Database Configuration Parameters.

Parameter Name	Token	Token Value	Data Type
locklist	SQLF_DBTN_LOCK_LIST	704	UInt64

The new maximum for this parameter is 524 288.

SQLEDBDESC

Two values will be added to the list of valid values for SQLDBCSS (defined in sqlenv). They are:

SQL_CS_SYSTEM_NLSCHAR

Collating sequence from system using the NLS version of compare routines for character types.

SQL_CS_USER_NLSCHAR

Collating sequence from user using the NLS version of compare routines for character types.

SQLFUPD Documentation Error

In „Chapter 3. Data Structures“, Table 53. Updatable Database Configuration Parameters incorrectly lists the token value for dbheap as 701. The correct value is 58.

Application Building Guide

Chapter 1. Introduction

Supported Software

Anmerkung: **PHP.** PHP can now be used as a method to access DB2 from web-based applications. PHP is a server-side, HTML-embedded, cross-platform scripting language. It supports DB2 access using the Unified-ODBC access method, in which the user-level PHP communicates to DB2 using ODBC calls. Unlike standard ODBC, with the Unified-ODBC method, communication is directly to the DB2 CLI layer, not through the ODBC layer. For more information about using PHP with DB2, search the DB2 support site at www.ibm.com/software/data/db2/udb/winos2unix/support.

AIX

The listed versions for C and C++ compilers should be the following:

IBM C and C++ Compilers for AIX Version 3.6.6 (Version 3.6.6.3 for 64-bit)
IBM C for AIX 4.4
IBM VisualAge C++ Version 4.0

Anmerkung: Please download the latest available FixPaks for these compiler versions from

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

The listed versions for the Micro Focus COBOL compiler should be the following:

AIX 4.2.1

Micro Focus COBOL Version 4.0.20 (PRN 12.03 or later)
Micro Focus COBOL Version 4.1.10 (PRN 13.04 or later)

AIX 4.3

Micro Focus COBOL Server Express Version 1.0

Anmerkung: For information on DB2 support for Micro Focus COBOL stored procedures and UDFs on AIX 4.3, see the DB2 Application Development Web page:

<http://www.ibm.com/software/data/db2/udb/ad>

To build 64-bit applications with the IBM XL Fortran for AIX Version 5.1.0 compiler, use the "-q64" option in the compile and link steps. Note that 64-bit applications are not supported on earlier versions of this compiler.

HP-UX

The listed version for the C++ compiler should be the following:

HP aC++, Version A.03.25

Anmerkung: HP does not support binary compatibility among objects compiled with old and new compilers, so this will force recompiles of any C++

application built to access DB2 on HP-UX. C++ applications must also be built to handle exceptions with this new compiler.

This is the URL for the aCC transition guide:
<http://www.hp.com/esy/lang/cpp/tguide>. The C++ incompatibility portion is here:

<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.1.2>
<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.3.3>

The C vs C++ portion is here:

<http://www.hp.com/esy/lang/cpp/tguide/transcontent.html#RN.CVT.3.3.1>

Even though C and aCC are compatible, when using the two different object types, the object containing "main" must be compiled with aCC, and the final executable must be linked with aCC.

Linux

DB2 for Linux supports the following REXX version:

Object REXX Interpreter for Linux Version 2.1

Linux/390

DB2 for Linux/390 supports only Java, C and C++.

OS/2

The listed versions for C/C++ compiler should be the following:

IBM VisualAge C++ for OS/2 Version 3.6.5 and Version 4.0

Anmerkung: Please download the latest available FixPaks for these compiler versions from

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

For limitations on future service support for these VisualAge C++ compilers, please see the news section at:

<http://www-4.ibm.com/software/ad/vacpp/>

Solaris

The listed version for the Micro Focus COBOL compiler should be:

Micro Focus COBOL Server Express Version 1.0

Windows 32-bit Operating Systems

The listed versions for the IBM VisualAge C++ compiler should be the following:

IBM VisualAge C++ for Windows Versions 3.6.5 and 4.0

Anmerkung: Please download the latest available FixPaks for these compiler versions from

<http://www.ibm.com/software/ad/vacpp/service/csd.html>

For limitations on future service support for these VisualAge C++ compilers, please see the news section at:

<http://www-4.ibm.com/software/ad/vacpp/>

The listed versions for the Micro Focus COBOL compiler should be the following:

Micro Focus COBOL Version 4.0.20
Micro Focus COBOL Net Express Version 3.0

Sample Programs

The following should be added to the "Object Linking and Embedding Samples" section:

salaryctvc

A Visual C++ DB2 CLI sample that calls the Visual Basic stored procedure, salarysrv.

SALSVADO

A sample OLE automation stored procedure (SALSVADO) and a SALCLADO client (SALCLADO), implemented in 32-bit Visual Basic and ADO, that calculates the median salary in table staff2.

The following should be added to the "Log Management User Exit Samples" section:

Applications on AIX using the ADSM API Client at level 3.1.6 and higher must be built with the xlc_r or xlc_r compiler invocations, not with xlc or xlc, even if the applications are single-threaded. This ensures that the libraries are thread-safe. This applies to the Log Management User Exit Sample, db2uext2.cadsm.

If you have an application that is compiled with a non thread-safe library, you can apply fixtest IC21925E or contact your application provider. The fixtest is available on the index.storsys.ibm.com anonymous ftp server. This will regress the ADSM API level to 3.1.3.

Chapter 3. General Information for Building DB2 Applications

Build Files, Makefiles, and Error-checking Utilities

The entry for blddevm in table 16 should read:

blddevm

The event monitor sample program, evm (only available on AIX, OS/2, and Windows 32-bit operating systems).

Table 17 should include the entries:

bldmevm

The event monitor sample program, evm, with the Microsoft Visual C++ compiler.

bldvevm

The event monitor sample program, evm, with the VisualAge C++ compiler.

Chapter 4. Building Java Applets and Applications

Setting the Environment

If you are using IBM JDK 1.1.8 on supported platforms to build SQLJ programs, a JDK build date of November 24, 1999 (or later) is required. Otherwise you may get JNI panic errors during compilation.

If you are using IBM JDK 1.2.2 on supported platforms to build SQLJ programs, a JDK build date of April 17, 2000 (or later) is required. Otherwise, you may get Invalid Java type errors during compilation.

For sub-sections **AIX**, **HP-UX**, **Linux**, and **Solaris**, replace the information on JDBC 2.0 with the following:

Using the JDBC 2.0 Driver with Java Applications

The JDBC 1.22 driver is still the default driver on all operating systems. To take advantage of the new features of JDBC 2.0, you must install JDK 1.2 support. Before executing an application that takes advantage of the new features of JDBC 2.0, you must set your environment by issuing the `usejdbc2` command from the `sqllib/java12` directory. If you want your applications to always use the JDBC 2.0 driver, consider adding the following line to your login profile, such as `.profile`, or your shell initialization script, such as `.bashrc`, `.cshrc`, or `.kshrc`:

```
. sqllib/java12/usejdbc2
```

Ensure that this command is placed after the command to run `db2profile`, as `usejdbc2` should be run after `db2profile`.

To switch back to the JDBC 1.22 driver, execute the following command from the `sqllib/java12` directory:

```
. usejdbc1
```

Using the JDBC 2.0 Driver with Java Stored Procedures and UDFs

To use the JDBC 2.0 driver with Java stored procedures and UDFs, you must set the environment for the fenced user ID for your instance. The default fenced user ID is `db2fenc1`. To set the environment for the fenced user ID, perform the following steps:

1. Add the following line to the fenced user ID profile, such as `.profile`, or the fenced user ID shell initialization script, such as `.bashrc`, `.cshrc`, or `.kshrc`:

```
. sqllib/java12/usejdbc2
```

2. Issue the following command from the CLP:

```
db2set DB2_USE_JDK12=1
```

To switch back to the JDBC 1.22 driver support for Java UDFs and stored procedures, perform the following steps:

1. Remove the following line from the fenced user ID profile, such as `.profile`, or the fenced user ID shell initialization script, such as `.bashrc`, `.cshrc`, or `.kshrc`:

```
. sqllib/java12/usejdbc2
```

2. Issue the following command from the CLP:

```
db2set DB2_USE_JDK12=
```

If you want your applications to always use the JDBC 2.0 driver, you can add the following line to your login profile, such as `.profile`, or your shell initialization script, such as `.bashrc`, `.cshrc`, or `.kshrc`:

```
. sqllib/java12/usejdbc2
```

Ensure that this command is placed after the command to run `db2profile`, as `usejdbc2` should be run after `db2profile`.

HP-UX

Java stored procedures and user-defined functions are not supported on DB2 for HP-UX with JDK 1.1.

Silicon Graphics IRIX

When building SQLJ applications with the `-o32` object type, using the Java JIT compiler with JDK 1.2.2, if the SQLJ translator fails with a segmentation fault, try turning off the JIT compiler with this command:

```
export JAVA_COMPILER=NONE
```

JDK 1.2.2 is required for building Java SQLJ programs on Silicon Graphics IRIX.

Windows 32-bit Operating Systems

Using the JDBC 2.0 Driver with Java Stored Procedures and UDFs

To use the JDBC 2.0 driver with Java stored procedures and UDFs, you must set the environment by performing the following steps:

1. Issue the following command in the `sqllib\java12` directory:

```
usejdbc2
```

2. Issue the following command from the CLP:

```
db2set DB2_USE_JDK12=1
```

To switch back to the JDBC 1.22 driver support for Java UDFs and stored procedures, perform the following steps:

1. Issue the following command in the `sqllib\java12` directory:

```
usejdbc2
```

2. Issue the following command from the CLP:

```
db2set DB2_USE_JDK12=
```

JDK Level on OS/2

Some messages will not display on OS/2 running versions of JDK 1.1.8 released prior to 09/99. Ensure that you have the latest JDK Version 1.1.8.

Java2 on HP-UX

To run Java2 stored procedures, the shared library path has to be changed to be similar to the following:

```
export SHLIB_PATH=$JAVADIR/jre/lib/PA_RISC:$JAVADIR/  
jre/lib/PA_RISC/classic:$HOME/sqllib/lib:/usr/lib:$SHLIB_PATH
```

`$JAVADIR` is the location of the Java2 SDK.

Chapter 5. Building SQL Procedures

Setting the SQL Procedures Environment

These instructions are in addition to the instructions for setting up the DB2 environment in "Setup".

For SQL procedures support, you have to install the Application Development Client on the server. For information about installing the Application Development Client, refer to the *Einstieg* book for your platform. For the C and C++ compilers supported by DB2 on your platform, see "Supported Software by Platform".

Anmerkung: On an OS/2 FAT file system, you are limited to a schema name for SQL Procedures of eight characters or less. You have to use the HPFS file system for schema names longer than eight characters.

The compiler configuration consists of two parts: setting the environment variables for the compiler, and defining the compilation command. The environment variables provide the paths to the compiler's binaries, libraries and include files. The compilation command is the full command that DB2 will use to compile the C files generated for SQL procedures.

Setting the Compiler Environment Variables

There are different rules for configuring the environment on OS/2, Windows, and UNIX based operating systems, as explained below. In some cases, no configuration is needed; in other cases, the DB2_SQLROUTINE_COMPILER_PATH DB2 registry variable must be set to point to an executable script that sets the environment variables appropriately.

Anmerkung: You can either use the **db2set** command or use the SQL Stored Procedures Build Options dialog from the Stored Procedure Builder to set the value of this DB2 registry variable. Using the SQL Stored Procedures Build Options dialog eliminates the need to physically access the database server or for the database server to be restarted in order for the changes to take effect.

On OS/2:

for IBM VisualAge C++ for OS/2 Version 3.6:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcxxo\bin\setenv.cmd"
```

for IBM VisualAge C++ for OS/2 Version 4:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcpp40\bin\setenv.cmd"
```

Anmerkung: For these commands, it is assumed that the C++ compiler is installed on the c: drive. Change the drive or the path, if necessary, to reflect the location of the C++ compiler on your system.

On Windows 32-bit operating systems, if the environment variables for your compiler are set as SYSTEM variables, no configuration is needed. Otherwise, set the DB2_SQLROUTINE_COMPILER_PATH DB2 registry variable as follows:

for Microsoft Visual C++ Versions 5.0:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\devstudio\vc\bin\vcvars32.bat"
```

for Microsoft Visual C++ Versions 6.0:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\Micros~1\vc98\bin\vcvars32.bat"
```

for IBM VisualAge C++ for Windows Version 3.6:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcxxw\bin\setenv.bat"
```

for IBM VisualAge C++ for Windows Version 4:

```
db2set DB2_SQLROUTINE_COMPILER_PATH="c:\ibmcppw40\bin\setenv.bat"
```

Anmerkung: For these commands, it is assumed that the C++ compiler is installed on the c: drive. Change the drive or the path, if necessary, to reflect the location of the C++ compiler on your system.

On UNIX based operating systems, DB2 will generate the executable script file \$HOME/sqllib/function/routine/sr_cpath (which contains the default values for the compiler environment variables) the first time you compile a stored procedure. You can edit this file if the default values are not appropriate for your compiler. Alternatively, you can set the DB2_SQLROUTINE_COMPILER_PATH DB2 registry variable to contain the full path name of another executable script that specifies the desired settings (see examples above).

Customizing the Compilation Command

The installation of the Application Development Client provides a default compilation command that works for at least one of the compilers supported on each platform:

```
AIX: IBM C Set++ for AIX Version 3.6.6
Solaris: SPARCCompiler C++ Versions 4.2 and 5.0
HP-UX: HP-UX C++ Version A.12.00
Linux: GNU/Linux g++ Version egcs-2.90.27 980315 (egcs-1.0.2 release)
PTX: ptx/C++ Version 5.2
OS/2: IBM VisualAge C++ for OS/2 Version 3
Windows NT and Windows 2000: Microsoft Visual C++ Versions 5.0 and 6.0
```

To use other compilers, or to customize the default command, you must set the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable with a command like:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=compilation_command
```

where *compilation_command* is the C or C++ compilation command, including the options and parameters required to create stored procedures.

In the compilation command, use the keyword SQLROUTINE_FILENAME to replace the filename for the generated SQC, C, PDB, DEF, EXP, messages log and shared library files. For AIX only, use the keyword SQLROUTINE_ENTRY to replace the entry point name.

Anmerkung: You can either use the **db2set** command or use the SQL Stored Procedures Build Options dialog from the Stored Procedure Builder to set the value of this DB2 registry variable. Using the SQL Stored Procedures Build Options dialog eliminates the need to physically access the database server or for the database server to be restarted in order for the changes to take effect.

The following are the default values for the DB2_SQLROUTINE_COMPILE_COMMAND for C or C++ compilers on supported server platforms.

AIX

To use IBM C for AIX Version 3.6.6:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=xlc -H512 -T512 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \  
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -lc -ldb2
```

To use IBM C Set++ for AIX Version 3.6.6:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=x1C -H512 -T512 \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c -bE:SQLROUTINE_FILENAME.exp \  
-e SQLROUTINE_ENTRY -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -lc -ldb2
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

Anmerkung: To compile 64-bit SQL procedures on AIX, add the -q64 option to the above commands.

To use IBM VisualAge C++ for AIX Version 4:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

If you do not specify the configuration file after vacbld command, DB2 will create the following default configuration file at the first attempt of creating any SQL procedure:

```
$HOME/sql1lib/function/routine/sqlproc.icc
```

If you want to use your own configuration file, you can specify your own configuration file when setting the DB2 registry value for DB2_SQLROUTINE_COMPILE_COMMAND:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld  
%DB2PATH%/function/sqlproc.icc"
```

HP-UX

To use HP C Compiler Version A.11.00.03:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc +DAportable +ul -Aa +z \  
-I$HOME/sql1lib/include -c SQLROUTINE_FILENAME.c; \  
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \  
-L$HOME/sql1lib/lib -ldb2
```

To use HP-UX C++ Version A.12.00:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC +DAportable +a1 +z -ext \  
-I$HOME/sql1lib/include -c SQLROUTINE_FILENAME.c; \  
ld -b -o SQLROUTINE_FILENAME SQLROUTINE_FILENAME.o \  
-L$HOME/sql1lib/lib -ldb2
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

Linux

To use GNU/Linux gcc Version 2.7.2.3:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

To use GNU/Linux g++ Version egcs-2.90.27 980315 (egcs-1.0.2 release):

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=g++ \
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \
-shared -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib -ldb2
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

PTX

To use ptx/C Version 4.5:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -KPIC \
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \
-G -o SQLROUTINE_FILENAME.so -L$HOME/sql1lib/lib -ldb2 ; \
cp SQLROUTINE_FILENAME.so SQLROUTINE_FILENAME
```

To use ptx/C++ Version 5.2:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=c++ -KPIC \
-D_RWSTD_COMPILE_INSTANTIATE=0 \
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \
-G -o SQLROUTINE_FILENAME.so -L$HOME/sql1lib/lib -ldb2 ; \
cp SQLROUTINE_FILENAME.so SQLROUTINE_FILENAME
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

OS/2

To use IBM VisualAge C++ for OS/2 Version 3:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="icc -Ge- -Gm+ -W2 \
-I%DB2PATH%\include SQLROUTINE_FILENAME.c \
/B\"/NOFREE /NOI /ST:64000\" SQLROUTINE_FILENAME.def \
%DB2PATH%\lib\db2api.lib"
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

To use IBM VisualAge C++ for OS/2 Version 4:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

If you do not specify the configuration file after vacbld command, DB2 will create the following default configuration file at the first attempt of creating any SQL procedure:

```
%DB2PATH%\function\routine\sqlproc.icc
```

If you want to use your own configuration file, you can specify your own configuration file when setting the DB2 registry value for DB2_SQLROUTINE_COMPILE_COMMAND:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld \
%DB2PATH%\function\sqlproc.icc"
```

Solaris

To use SPARCompiler C Versions 4.2 and 5.0:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=cc -xarch=v8plusa -Kpic \
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \
-R$HOME/sql1lib/lib -ldb2
```

To use SPARCompiler C++ Versions 4.2 and 5.0:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=CC -xarch=v8plusa -Kpic \  
-I$HOME/sql1lib/include SQLROUTINE_FILENAME.c \  
-G -o SQLROUTINE_FILENAME -L$HOME/sql1lib/lib \  
-R$HOME/sql1lib/lib -ldb2
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

Anmerkungen:

1. The compiler option `-xarch=v8plusa` has been added to the default compiler command. For details on why this option has been added, see „Chapter 12. Building Solaris Applications“ auf Seite 354.
2. To compile 64-bit SQL procedures on Solaris, take out the `-xarch=v8plusa` option and add the `-xarch=v9` option to the above commands.

Windows NT and Windows 2000

Anmerkung: SQL procedures are not supported on Windows 98 or Windows 95.

To use Microsoft Visual C++ Versions 5.0 and 6.0:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=c1 -Od -W2 /TC -D_X86_=1  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c /link -dll  
-def:SQLROUTINE_FILENAME.def /out:SQLROUTINE_FILENAME.dll  
%DB2PATH%\lib\db2api.lib
```

This is the default compile command if the DB2_SQLROUTINE_COMPILE_COMMAND DB2 registry variable is not set.

To use IBM VisualAge C++ for Windows Version 3.6:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="i1lib /GI  
SQLROUTINE_FILENAME.def & icc -Ti -Ge- -Gm+ -W2  
-I%DB2PATH%\include SQLROUTINE_FILENAME.c  
/B\"/ST:64000 /PM:VIO /DLL\" SQLROUTINE_FILENAME.exp  
%DB2PATH%\lib\db2api.lib"
```

To use IBM VisualAge C++ for Windows Version 4:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld"
```

If you do not specify the configuration file after `vacbld` command, DB2 will create the following default configuration file at the first attempt of creating any SQL procedure:

```
%DB2PATH%\function\routine\sqlproc.icc
```

If you want to use your own configuration file, you can specify your own configuration file when setting the DB2 registry value for DB2_SQLROUTINE_COMPILE_COMMAND:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="vacbld  
%DB2PATH%\function\sqlproc.icc"
```

To return to the default compiler options, set the DB2 registry value for DB2_SQLROUTINE_COMPILE_COMMAND to null with the following command:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND=
```

Retaining Intermediate Files

You have to manually delete intermediate files that may be left when an SQL procedure is not created successfully. These files are in the following directories:

UNIX \$DB2PATH/function/routine/sqlproc/\$DATABASE/\$SCHEMA/tmp

where \$DB2PATH represents the directory in which the instance was created, \$DATABASE represents the database name, and \$SCHEMA represents the schema name with which the SQL procedures were created.

OS/2 and Windows

%DB2PATH%\function\routine\sqlproc\%DATABASE%\%SCHEMA%\tmp

where %DB2PATH% represents the directory in which the instance was created, %DATABASE% represents the database name, and %SCHEMA% represents the schema name with which the SQL procedures were created.

Backup and Restore

When an SQL procedure is created, the generated shared library/DLL is also kept in the catalog table if the generated shared library/DLL is smaller than 2 MB. When the database is backed up and restored, any SQL procedure with a generated shared library/DLL less than 2 MB will be backed up and restored with the version kept in the catalog table. If you have SQL procedures with a generated shared library/DLL larger than 2 MB, ensure that you also do the filesystem backup and restore with the database backup and restore. If not, you will have to recreate the shared library/DLL of the SQL procedure manually by using the source in the syscat.procedures catalog table.

Anmerkung: At database recovery time, all the SQL procedure executables on the filesystem belonging to the database being recovered will be removed. If the index creation configuration parameter (indexrec) is set to RESTART, all SQL procedure executables will be extracted from the catalog table and put back on the filesystem at next connect time. Otherwise, the SQL executables will be extracted on first execution of the SQL procedures.

The executables will be put back in the following directory:

UNIX \$DB2PATH/function/routine/sqlproc/\$DATABASE

where \$DB2PATH represents the directory in which the instance was created and \$DATABASE represents the database name with which the SQL procedures were created.

OS/2 and Windows

%DB2PATH%\function\routine\sqlproc\%DATABASE%

where %DB2PATH% represents the directory in which the instance was created and %DATABASE% represents the database name with which the SQL procedures were created.

Creating SQL Procedures

Set the database manager configuration parameter KEEP DARI to 'NO' for developing SQL procedures. If an SQL procedure is kept loaded once it is executed, you may have problems dropping and recreating the stored procedure with the same name, as the library cannot be refreshed and the executables cannot be dropped from the filesystem. You will also have problems when you try to rollback the changes or drop the database because the executables cannot be deleted.

See 'Updating the Database Manager Configuration File' in "Chapter 2. Setup" of the 'Application Building Guide' for more information on setting the KEEPDAI parameter.

Anmerkung: SQL procedures do not support the following data types for parameters:

- LONG VARGRAPHIC
- Binary Large Object (BLOB)
- Character Large Object (CLOB)
- Double-byte Character Large Object (DBCLOB)

Calling Stored Procedures

The first paragraph in 'Using the CALL Command' should read:

To use the call command, you must enter the stored procedure name plus any IN or INOUT parameters, as well as '?' as a place-holder for each OUT parameter. For details on the syntax of the CALL command, see „CALL“ auf Seite 153.

Distributing Compiled SQL Procedures

Anmerkung: To distribute compiled SQL procedures between DB2 servers, you must perform the following steps for every DB2 server that serves as the source of, or the destination for, a compiled SQL procedure:

Schritt 1. Install FixPak 3

Schritt 2. Issue the **db2updv7** command to enable DB2 to extract and install compiled SQL procedures:

```
db2updv7 -d database_name
```

When you define an SQL procedure, it is converted to a C program, precompiled, bound against the target database, compiled and linked to create a shared library. The compile and link steps require a C or C++ compiler to be available on the database server machine. However, once you define an SQL procedure, you can distribute it in compiled form to DB2 databases that run on the same platform but do not necessarily have access to a C or C++ compiler. DB2 allows the user to extract SQL procedures in compiled form from one database and install SQL procedures in compiled form into another database.

DB2 provides both a command line interface and a programming interface to the extraction and installation operations. The command line interface consists of two CLP commands: GET ROUTINE and PUT ROUTINE. The programmatic interface consists of two built-in stored procedures: GET_ROUTINE_SAR and PUT_ROUTINE_SAR. For more information on the command line interface, refer to the Command Reference. For more information on the the programming interface, refer to the SQL Reference.

To distribute a compiled SQL procedure from one database server to another database server, perform the following steps:

Schritt 1. Develop the application, including defining the SQL procedures that are part of the application.

Schritt 2. After testing the procedures, extract the compiled version of each procedure into a different file. For more information, refer to the GET ROUTINE command in the Command Reference or the GET_ROUTINE_SAR stored procedure in the SQL Reference.

Schritt 3. Install the compiled version of each procedure on each server, either by issuing the PUT ROUTINE command, or by invoking the PUT_ROUTINE_SAR stored procedure, using the files created by 2 auf Seite 352. Each database server must have the same operating system and DB2 level.

Chapter 7. Building HP-UX Applications.

HP-UX C

In "Multi-threaded Applications", the bldmt script file has been revised with different compile options. The new version is in the sqllib/samples/c directory.

HP-UX C++

In the build scripts, the C++ compiler variable CC has been replaced by aCC, for the HP aC++ compiler. The revised build scripts are in the sqllib/samples/cpp directory.

The "+u1" compile option should be used to build stored procedures and UDFs with the aCC compiler. This option allows unaligned data access. The sample build scripts shipped with DB2 for HP-UX, bldsrv and bldudf, and the sample makefile, have not been updated with this option. They should be revised to add this option before use. Here is the new compile step for the bldsrv and bldudf scripts:

```
aCC +DAportable +u1 -Aa +z -ext -I$DB2PATH/include -c $1.C
```

In "Multi-threaded Applications", the bldmt script file has been revised with different compile options. The new version is in the sqllib/samples/cpp directory.

Chapter 9. Building OS/2 Applications

VisualAge C++ for OS/2 Version 4.0

For OS/2 and Windows, use the set command instead of the export command documented in this section. For example, set CLI=tbinfo.

In 'DB2 CLI Applications', sub-section 'Building and Running Embedded SQL Applications', for OS/2 and Windows the cliapi.icc file must be used instead of the cli.icc file, as embedded SQL applications need the db2api.lib library linked in by cliapi.icc.

Chapter 10. Building PTX Applications

ptx/C++

Libraries need to be linked with the -shared option to build stored procedures and user-defined functions. In the sqllib/samples directory, the makefile, the build scripts bldsrv, and bldudf have been updated to include this option, as in the following link step from bldsrv:

```
c++ -shared -G -o $1 $1.o -L$DB2PATH/lib -ldb2
```

Chapter 12. Building Solaris Applications

SPARCompiler C++

Problems with executing C/C++ Applications and running SQL Procedures on Solaris

When using the Sun WorkShop Compiler C/C++, if you experience problems with your executable where you receive errors like the following:

1. syntax error at line 1: '(' unexpected
2. ksh: <application name>: cannot execute (where application name is the name of the compiled executable)

you may be experiencing a problem that the compiler does not produce valid executables when linking with libdb2.so. One suggestion to fix this is to add the following compiler option to your compile and link commands:

```
-xarch=v8plusa
```

for example, when compiling the sample application, dynamic.c.sqc:

```
embprep dynamic sample
embprep utilemb sample
cc -c utilemb.c -xarch=v8plusa -I/export/home/db2inst1/sqllib/include
cc -o dynamic dynamic.c utilemb.o -xarch=v8plusa -I/export/home/db2inst1/
    sqllib/include \
    -L/export/home/db2inst1/sqllib/lib -R/export/home/db2inst1/sqllib/lib -l db2
```

Anmerkungen:

1. If you are using SQL Procedures on Solaris, and you are using your own compile string via the DB2_SQLROUTINE_COMPILE_COMMAND profile variable, please ensure that you include the compiler option given above. The default compiler command includes this option:

```
db2set DB2_SQLROUTINE_COMPILE_COMMAND="cc -# -Kpic -xarch=v8plusa
-I$HOME/sqllib/include \
SQLROUTINE_FILENAME.c -G -o SQLROUTINE_FILENAME -L$HOME/sqllib/lib
-R$HOME/sqllib/lib -ldb2
```

2. To compile 64-bit SQL procedures on Solaris, take out the -xarch=v8plusa option and add the -xarch=v9 option to the above commands.

Chapter 13. Building Applications for Windows 32-bit Operating Systems

VisualAge C++ Version 4.0

For OS/2 and Windows, use the set command instead of the export command documented in this section. For example, set CLI=tbinfo.

In 'DB2 CLI Applications', sub-section 'Building and Running Embedded SQL Applications', for OS/2 and Windows the cliapi.icc file must be used instead of the cli.icc file, as embedded SQL applications need the db2api.lib library linked in by cliapi.icc.

Application Development Guide

Chapter 2. Coding a DB2 Application

Activating the IBM DB2 Universal Database Project and Tool Add-ins for Microsoft Visual C++

Before running the `db2vccmd` command (step 1), please ensure that you have started and stopped Visual C++ at least once with your current login ID. The first time you run Visual C++, a profile is created for your user ID, and that is what gets updated by the `db2vccmd` command. If you have not started it once, and you try to run `db2vccmd`, you may see errors like the following:

```
"Registering DB2 Project add-in ...Failed! (rc = 2)"
```

Chapter 6. Common DB2 Application Techniques

Generating Sequential Values

Generating sequential values is a common database application development problem. The best solution to that problem is to use sequence objects and sequence expressions in SQL. Each *sequence object* is a uniquely named database object that can be accessed only by sequence expressions. There are two *sequence expressions*: the PREVVAL expression and the NEXTVAL expression. The PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement. The NEXTVAL sequence expression increments the value of the sequence object and returns the new value of the sequence object.

To create a sequence object, issue the CREATE SEQUENCE statement. For example, to create a sequence object called `id_values` using the default attributes, issue the following statement:

```
CREATE SEQUENCE id_values
```

To display the current value of the sequence object, issue a VALUES statement using the PREVVAL expression:

```
VALUES PREVVAL FOR id_values
```

```
1
-----
1
          1 record(s) selected.
```

You can repeatedly retrieve the current value of the sequence object, and the value that the sequence object returns does not change until you issue a NEXTVAL expression. In the following example, the PREVVAL expression returns a value of 1, until the NEXTVAL expression increments the value of the sequence object:

```
VALUES PREVVAL FOR id_values
```

```
1
-----
1
          1 record(s) selected.
```

```
VALUES PREVVAL FOR id_values
```

```

1
-----
1
          1 record(s) selected.

```

```
VALUES NEXTVAL FOR id_values
```

```

1
-----
          2
          1 record(s) selected.

```

```
VALUES PREVVAL FOR id_values
```

```

1
-----
          2
          1 record(s) selected.

```

To update the value of a column with the next value of the sequence object, include the NEXTVAL expression in the UPDATE statement, as follows:

```

UPDATE staff
   SET id = NEXTVAL FOR id_values
  WHERE id = 350

```

To insert a new row into a table using the next value of the sequence object, include the NEXTVAL expression in the INSERT statement, as follows:

```

INSERT INTO staff (id, name, dept, job)
   VALUES (NEXTVAL FOR id_values, 'Kandil', 51, 'Mgr')

```

For more information on the PREVVAL and NEXTVAL expressions, refer to the *SQL Reference*.

Controlling Sequence Behavior

You can tailor the behavior of sequence objects to meet the needs of your application. You change the attributes of a sequence object when you issue the CREATE SEQUENCE statement to create a new sequence object, and when you issue the ALTER SEQUENCE statement for an existing sequence object. Following are some of the attributes of a sequence object that you can specify:

Data type

The AS clause of the CREATE SEQUENCE statement specifies the numeric data type of the sequence object. The data type, as specified in the “SQL Limits” appendix of the *SQL Reference*, determines the possible minimum and maximum values of the sequence object. You cannot change the data type of a sequence object; instead, you must drop the sequence object by issuing the DROP SEQUENCE statement and issuing a CREATE SEQUENCE statement with the new data type.

Start value

The START WITH clause of the CREATE SEQUENCE statement sets the initial value of the sequence object. The RESTART WITH clause of the ALTER SEQUENCE statement resets the value of the sequence object to a specified value.

Minimum value

The MINVALUE clause sets the minimum value of the sequence object.

Maximum value

The MAXVALUE clause sets the maximum value of the sequence object.

Increment value

The INCREMENT BY clause sets the value that each NEXTVAL expression adds to the sequence object. To decrement the value of the sequence object, specify a negative value.

Sequence cycling

The CYCLE clause causes the value of a sequence object that reaches its maximum or minimum value to return to its start value on the following NEXTVAL expression.

For example, to create a sequence object called `id_values` that starts with a value of 0, has a maximum value of 1000, increments by 2 with each NEXTVAL expression, and returns to its start value when the maximum value is reached, issue the following statement:

```
CREATE SEQUENCE id_values
  START WITH 0
  INCREMENT BY 2
  MAXVALUE 1000
  CYCLE
```

For more information on the CREATE SEQUENCE and ALTER SEQUENCE statements, refer to the *SQL Reference*.

Improving Performance with Sequence Objects

Like identity columns, using sequence objects to generate values generally improves the performance of your applications in comparison to alternative approaches. The alternative to sequence objects is to create a single-column table that stores the current value and incrementing that value with either a trigger or under the control of the application. In a distributed environment where applications concurrently access the single-column table, the locking required to force serialized access to the table can seriously affect performance.

Sequence objects avoid the locking issues that are associated with the single-column table approach and can cache sequence values in memory to improve DB2 response time. To maximize the performance of applications that use sequence objects, ensure that your sequence object caches an appropriate amount of sequence values. The CACHE clause of the CREATE SEQUENCE and ALTER SEQUENCE statements specifies the maximum number of sequence values that DB2 generates and stores in memory.

If your sequence object must generate values in order, without introducing gaps in that order due to a system failure or database deactivation, use the ORDER and NO CACHE clauses in the CREATE SEQUENCE statement. The NO CACHE clause guarantees that no gaps appear in the generated values at the cost of some of your application's performance because it forces your sequence object to write to the database log every time it generates a new value.

Comparing Sequence Objects and Identity Columns

Although sequence objects and identity columns appear to serve similar purposes for DB2 applications, there are a number of important differences:

- An identity column automatically generates values for a column in a single table. A sequence object generates sequential values that can be used in any SQL statement.
- An identity column generates values that are guaranteed to be unique. Including the CYCLE clause in a CREATE SEQUENCE or ALTER SEQUENCE statement enables that sequence object to generate duplicate values.

Chapter 7. Stored Procedures

DECIMAL Type Fails in Linux Java Routines

This problem occurs because the IBM Developer Kit for Java does not create links for its libraries in the `/usr/lib` directory. The security model for DB2 routines does not allow them to access libraries outside of the standard system libraries. To enable DECIMAL support in Java routines on Linux, perform the following steps:

1. Create symbolic links from the IBM Developer Kit for Java libraries to `/usr/lib/` by issuing the following command with root authority:

For IBM Developer Kit for Java 1.1.8:

```
ln -sf /usr/jdk118/lib/linux/native_threads/* /usr/lib/
```

For IBM Developer Kit for Java 1.3:

```
ln -sf /opt/IBMJava2-13/jre/bin/*.so /usr/lib/
```

2. Issue the `ldconfig` command to update the list of system-wide libraries.

Using Cursors in Recursive Stored Procedures

To avoid errors when using SQL Procedures or stored procedures written in embedded SQL, close all open cursors before issuing a recursive CALL statement.

For example, assume the stored procedure MYPROC contains the following code fragment:

```
OPEN c1;
CALL MYPROC();
CLOSE c1;
```

DB2 returns an error when MYPROC is called because cursor `c1` is still open when MYPROC issues a recursive CALL statement. The specific error returned by DB2 depends on the actions MYPROC performs on the cursor.

To successfully call MYPROC, rewrite MYPROC to close any open cursors *before* the nested CALL statement as shown in the following example:

```
OPEN c1;
CLOSE c1;
CALL MYPROC();
```

Close all open cursors before issuing the nested CALL statement to avoid an error.

Writing OLE Automation Stored Procedures

The last sentence in the following paragraph is missing from the second paragraph under section "Writing OLE automation Stored Procedures":

```
After you code an OLE automation object, you must register the methods of the object as stored procedures using the CREATE PROCEDURE statement. To register an OLE automation stored procedure, issue a CREATE PROCEDURE statement with the LANGUAGE OLE clause. The external name consists of the OLE progID identifying the OLE automation object and the method name separated by ! (exclamation mark). The OLE automation object needs to be implemented as an in-process server (.DLL).
```

Chapter 12. Working with Complex Objects: User-Defined Structured Types

Inserting Structured Type Attributes Into Columns

The following rule applies to embedded static SQL statements: To insert an attribute of a user-defined structured type into a column that is of the same type as the attribute, enclose the host variable that represents the instance of the type in parentheses, and append the double-dot operator and attribute name to the closing parenthesis. For example, consider the following situation:

- PERSON_T is a structured type that includes the attribute NAME of type VARCHAR(30).
- T1 is a table that includes a column C1 of type VARCHAR(30).
- personhv is the host variable declared for type PERSON_T in the programming language.

The proper syntax for inserting the NAME attribute into column C1 is:

```
EXEC SQL INSERT INTO T1 (C1) VALUES (:personhv)..NAME
```

Chapter 13. Using Large Objects (LOBs)

Large object (LOBs) support in federated database systems

DB2 supports three types of large objects (LOBs): character large objects (CLOBs), double-byte character large objects (DBCLOBs) and binary large objects (BLOBs). For general information about DB2 LOB support, see the following DB2 books:

- *DB2 Application Development Guide*
- *DB2 SQL Reference*
- *DB2 Administration Guide: Planning*

In a federated database system, you can access and manipulate LOBs at remote data sources. Because LOBs can be very large, transferring LOBs from a remote data source can be time consuming. The DB2 federated database attempts to minimize transferring LOB data from the data sources, and also attempts to deliver requested LOB data directly from the data source to the requesting application without materializing the LOB at DB2.

This section discusses:

- How DB2 retrieves LOBs
- How applications can use LOB locators
- Restrictions on LOBs
- Mappings between LOB and non-LOB data types
- Tuning the system

How DB2 retrieves LOBs

DB2 federated systems use two mechanisms to retrieve LOBs: LOB streaming and LOB materialization.

LOB streaming: In LOB streaming, LOB data is retrieved in stages. DB2 uses LOB streaming for data in result sets of queries that are completely pushed down. For example, consider the following query:

```
SELECT empname, picture FROM orc_emp_table WHERE empno = '01192345'
```

where *picture* represents a LOB column and *orc_emp_table* represents a nickname referencing an Oracle table containing employee data. The DB2 query processor

marks the *picture* column for streaming if it decides to run the entire query at the Oracle data source. At execution time, if DB2 notes that a LOB is marked for streaming, it retrieves the LOB in stages from the data source. DB2 then transfers the data to the application memory space.

LOB materialization: In LOB materialization, the remote LOB data is retrieved by DB2 and stored locally at the federated server. DB2 uses LOB materialization when:

- The LOB column cannot be deferred or streamed.
- A function must be applied to a LOB column locally, before the data is transferred. This happens when DB2 compensates for functions not available at a remote data source. For example, Microsoft SQL Server does not provide a SUBSTR function for LOB columns. To compensate, DB2 materializes the LOB column locally and applies the DB2 SUBSTR function to the retrieved LOB.

How applications can use LOB locators

Applications can request LOB locators for LOBs stored in remote data sources. A LOB locator is a 4-byte value stored in a host variable that a program can use to refer to a LOB value (or LOB expression) held in the database system. Using a LOB locator, a program can manipulate the LOB value as if the LOB value was stored in a regular host variable. The difference in using the LOB locator is that there is no need to transport the LOB value from the server to the application (and possibly back again). See the *DB2 Application Development Guide* for additional information about LOB locators.

DB2 can retrieve LOBs from remote data sources, store them at DB2, and then issue a LOB locator against the stored LOB. LOB locators are released when:

- Applications issue "FREE LOCATOR" SQL statements.
- Applications issue COMMIT statements.
- DB2 is restarted.

Restrictions on LOBs

When using and retrieving LOBs, consider that:

- DB2 is unable to bind remote LOBs to a file reference variable.
- LOBs are not supported in pass-through mode.

Mappings between LOB and non-LOB data types

There are a few cases in which you can map a DB2 LOB data type to a non-LOB data type at a data source. When you need to create a mapping between a column with a DB2 LOB type and its counterpart column at a data source, it is recommended that you use a LOB data type as a counterpart if at all possible.

To create a mapping, use the create type mapping DDL statement. For example:
`CREATE TYPE MAPPING my_oracle_lob FROM sysibm.clob TO SERVER TYPE oracle TYPE long`

where:

my_oracle_lob

Is the name of the type mapping.

sysibm.clob

Is the DB2 CLOB data type.

oracle

Is the type of server you are connecting to.

long

Is the Oracle data type counterpart.

Tuning the system

If an application that retrieves remote LOBs returns an error message indicating there is not enough system resources to process the statement, increase the value of the application heap size parameter, `APPLHEAPSZ`, in the database configuration file. For example:

```
DB2 UPDATE DB CFG FOR EMPLOYEE USING APPLHEAPSZ 512
```

where `EMPLOYEE` is the name of the database you are tuning and `512` is the value of the application heap size parameter.

Part 5. DB2 Programming Considerations

IBM DB2 OLE DB Provider

Installing IBM DB2 Version 7.1 FixPak 1 or later corrects the condition that caused DB2 to issue the following error:

```
Test connection failed because of an error in initializing provider.  
The IBM OLE DB Provider is not available at this time. Please refer to  
the readme file for more information.
```

For more information on using the IBM OLE DB Provider for DB2, please refer to <http://www.ibm.com/software/data/db2/udb/ad/v71/oledb.html>.

Chapter 20. Programming in C and C++

The following table supplements the information included in chapter 7, "Stored Procedures", chapter 15, "Writing User-Defined Functions and Methods", and chapter 20, "Programming in C and C++". The table lists the supported mappings between SQL data types and C data types for stored procedures, UDFs, and methods.

C/C++ Types for Stored Procedures, Functions, and Methods

Tabelle 22. SQL Data Types Mapped to C/C++ Declarations

SQL Column Type	C/C++ Data Type	SQL Column Type Description
SMALLINT (500 or 501)	sqlint16	16-bit signed integer
INTEGER(496 or 497)	sqlint32	32-bit signed integer
BIGINT (492 or 493)	sqlint64	64-bit signed integer
REAL (480 or 481)	float	Single-precision floating point
DOUBLE (480 or 481)	double	Double-precision floating point
DECIMAL(<i>p,s</i>) (484 or 485)	Not supported.	To pass a decimal value, define the parameter to be of a data type castable from DECIMAL (for example CHAR or DOUBLE) and explicitly cast the argument to this type.
CHAR(<i>n</i>) (452 or 453)	char[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=254	Fixed-length, null-terminated character string
CHAR(<i>n</i>) FOR BIT DATA (452 or 453)	char[<i>n+1</i>] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=254	Fixed-length character string

Tabelle 22. SQL Data Types Mapped to C/C++ Declarations (Forts.)

SQL Column Type	C/C++ Data Type	SQL Column Type Description
VARCHAR(<i>n</i>) (448 or 449) (460 or 461)	char[<i>n</i> +1] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=32 672	Null-terminated varying length string
VARCHAR(<i>n</i>) FOR BIT DATA (448 or 449)	struct { sqluint16 length; char[<i>n</i>] } 1<= <i>n</i> <=32 672	Not null-terminated varying length character string
LONG VARCHAR (456 or 457)	struct { sqluint16 length; char[<i>n</i>] } 32 673<= <i>n</i> <=32 700	Not null-terminated varying length character string
CLOB(<i>n</i>) (408 or 409)	struct { sqluint32 length; char data[<i>n</i>]; } 1<= <i>n</i> <=2 147 483 647	Non null-terminated varying length character string with 4-byte string length indicator
BLOB(<i>n</i>) (404 or 405)	struct { sqluint32 length; char data[<i>n</i>]; } 1<= <i>n</i> <=2 147 483 647	Non null-terminated varying binary string with 4-byte string length indicator
DATE (384 or 385)	char[11]	null-terminated character form
TIME (388 or 389)	char[9]	null-terminated character form
TIMESTAMP (392 or 393)	char[27]	null-terminated character form
Anmerkung: The following data types are only available in the DBCS or EUC environment when precompiled with the WCHARTYPE NOCONVERT option.		
GRAPHIC(<i>n</i>) (468 or 469)	sqldbchar[<i>n</i> +1] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=127	Fixed-length, null-terminated double-byte character string
VARGRAPHIC(<i>n</i>) (400 or 401)	sqldbchar[<i>n</i> +1] where <i>n</i> is large enough to hold the data 1<= <i>n</i> <=16 336	Not null-terminated, variable-length double-byte character string
LONG VARGRAPHIC (472 or 473)	struct { sqluint16 length; sqldbchar[<i>n</i>] } 16 337<= <i>n</i> <=16 350	Not null-terminated, variable-length double-byte character string
DBCLOB(<i>n</i>) (412 or 413)	struct { sqluint32 length; sqldbchar data[<i>n</i>]; } 1<= <i>n</i> <=1 073 741 823	Non null-terminated varying length character string with 4-byte string length indicator

Chapter 21. Programming in Java

Java Method Signature in PARAMETER STYLE JAVA Procedures and Functions

If specified after the Java method name in the EXTERNAL NAME clause of the CREATE PROCEDURE or CREATE FUNCTION statement, the Java method signature must correspond to the default Java type mapping for the signature specified after the procedure or function name. For example, the default Java mapping of the SQL type INTEGER is "int", not "java.lang.Integer".

Connecting to the JDBC Applet Server

It is essential that the db2java.zip file used by the Java applet be at the same FixPak level as the JDBC applet server. Under normal circumstances, db2java.zip is loaded from the Web Server where the JDBC applet server is running, as shown in Figure 22 of the book. This ensures a match. If, however, your configuration has the Java applet loading db2java.zip from a different location, a mismatch can occur. Prior to FixPak 2, this could lead to unexpected failures. As of FixPak 2, matching FixPak levels between the two files is strictly enforced at connection time. If a mismatch is detected, the connection is rejected, and the client receives one of the following exceptions:

- If db2java.zip is at FixPak 2 or later:
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0621E Unsupported JDBC server configuration.
- If db2java.zip is prior to FixPak 2:
COM.ibm.db2.jdbc.DB2Exception: [IBM][JDBC Driver]
CLI0601E Invalid statement handle or statement is closed.
SQLSTATE=S1000

If a mismatch occurs, the JDBC applet server logs one of the following messages in the jdbcerr.log file:

- If the JDBC applet server is at FixPak 2 or later:
jdbcFSQLConnect: JDBC Applet Server and client (db2java.zip)
versions do not match. Unable to proceed with connection., einfo= -111
- If the JDBC applet server is prior to FixPak 2:
jdbcServiceConnection(): Invalid Request Received., einfo= 0

Appendix B. Sample Programs

The following should be added to the "Object Linking and Embedding Samples" section:

salarycltvc A Visual C++ DB2 CLI sample that calls the Visual Basic stored procedure, salarysrv.

SALSVADO A sample OLE automation stored procedure (SALSVADO) and a SALCLADO client (SALCLADO), implemented in 32-bit Visual Basic and ADO, that calculates the median salary in table staff2.

CLI Guide and Reference

Binding Database Utilities Using the Run-Time Client

The Run-Time Client cannot be used to bind the database utilities (import, export, reorg, the command line processor) and DB2 CLI bind files to each database before they can be used with that database. You must use the DB2 Administration Client or the DB2 Application Development Client instead.

You must bind these database utilities and DB2 CLI bind files to each database before they can be used with that database. In a network environment, if you are using multiple clients that run on different operating systems, or are at different versions or service levels of DB2, you must bind the utilities once for each operating system and DB2-version combination.

Using Static SQL in CLI Applications

For more information on using static SQL in CLI applications, see the Web page at: <http://www.ibm.com/software/data/db2/udb/staticcli/>

Limitations of JDBC/ODBC/CLI Static Profiling

JDBC/ODBC/CLI static profiling currently targets straightforward applications. It is not meant for complex applications with many functional components and complex program logic during execution.

An SQL statement must have successfully executed for it to be captured in a profiling session. In a statement matching session, unmatched dynamic statements will continue to execute as dynamic JDBC/ODBC/CLI calls.

An SQL statement must be identical character-by-character to the one that was captured and bound to be a valid candidate for statement matching. Spaces are significant: for example, "COL = 1" is considered different than "COL=1". Use parameter markers in place of literals to improve match hits.

When executing an application with pre-bound static SQL statements, dynamic registers that control the dynamic statement behavior will have no effect on the statements that are converted to static.

If an application issues DDL statements for objects that are referenced in subsequent DML statements, you will find all of these statements in the capture file. The JDBC/ODBC/CLI Static Profiling Bind Tool will attempt to bind them. The bind attempt will be successful with DBMSs that support the VALIDATE(RUN) bind option, but it fail with ones that do not. In this case, the application should not use Static Profiling.

The Database Administrator may edit the capture file to add, change, or remove SQL statements, based on application-specific requirements.

ADT Transforms

The following supercedes existing information in the book.

- There is a new descriptor type (smallint) `SQL_DESC_USER_DEFINED_TYPE_CODE`, with values:

```
SQL_TYPE_BASE 0      (this is not a USER_DEFINED_TYPE)
SQL_TYPE_DISTINCT 1
SQL_TYPE_STRUCTURED 2
This value can be queried with either SQLColAttribute
or SQLGetDescField (IRD only).
```

The following attributes are added to obtain the actual type names:

```
SQL_DESC_REFERENCE_TYPE
SQL_DESC_STRUCTURED_TYPE
SQL_DESC_USER_TYPE
```

The above values can be queried using `SQLColAttribute` or `SQLGetDescField` (IRD only).

- Add `SQL_DESC_BASE_TYPE` in case the application needs it. For example, the application may not recognize the structured type, but intends to fetch or insert it, and let other code deal with the details.
- Add a new connection attribute called `SQL_ATTR_TRANSFORM_GROUP` to allow an application to set the transform group (rather than use the SQL "SET CURRENT DEFAULT TRANSFORM GROUP" statement).
- Add a new statement/connection attribute called `SQL_ATTR_RETURN_USER_DEFINED_TYPES` that can be set or queried using `SQLSetConnectAttr`, which causes CLI to return the value `SQL_DESC_USER_DEFINED_TYPE_CODE` as a valid SQL type. This attribute is required before using any of the transforms.
 - By default, the attribute is off, and causes the base type information to be returned as the SQL type.
 - When enabled, `SQL_DESC_USER_DEFINED_TYPE_CODE` will be returned as the `SQL_TYPE`. The application is expected to check for `SQL_DESC_USER_DEFINED_TYPE_CODE`, and then to retrieve the appropriate type name. This will be available to `SQLColAttribute`, `SQLDescribeCol`, and `SQLGetDescField`.
- The `SQLBindParameter` does not give an error when you bind `SQL_C_DEFAULT`, because there is no code to allow `SQLBindParameter` to specify the type `SQL_USER_DEFINED_TYPE`. The standard default C types will be used, based on the base SQL type flowed to the server. Beispiel:

```
sqlrc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR, SQL_VARCHAR, 30,
                          0, &c2, 30, NULL);
```

Chapter 3. Using Advanced Features

Writing Multi-Threaded Applications

The following should be added to the end of the "Multi-Threaded Mixed Applications" section:

Anmerkung: It is recommended that you do not use the default stack size, but instead increase the stack size to at least 256 000. DB2 requires a minimum stack size of 256 000 when calling a DB2 function. You must ensure therefore, that you allocate a total stack size that is large enough for both your application and the minimum requirements for a DB2 function call.

Scrollable Cursors

The following information should be added to the "Scrollable Cursors" section:

Server-side Scrollable Cursor Support for OS/390

The UDB client for the Unix, Windows, and OS/2 platforms supports updatable server-side scrollable cursors when run against OS/390 Version 7 databases. To access an OS/390 scrollable cursor on a three-tier environment, the client and the gateway must be running DB2 UDB Version 7.1, FixPak 3 or later.

There are two application enablement interfaces that can access scrollable cursors: ODBC and JDBC. The JDBC interface can only access static scrollable cursors, while the ODBC interface can access static and keyset-driven server-side scrollable cursors.

Cursor Attributes: The table below lists the default attributes for OS/390 Version 7 cursors in ODBC.

Table 23. Default attributes for OS/390 cursors in ODBC

Cursor Type	Cursor Sensitivity	Cursor Updatable	Cursor Concurrency	Cursor Scrollable
forward-only ^a	unspecified	non-updatable	read-only concurrency	non-scrollable
static	insensitive	non-updatable	read-only concurrency	scrollable
keyset-driven	sensitive	updatable	values concurrency	scrollable

^a Forward-only is the default behavior for a scrollable cursor without the FOR UPDATE clause. Specifying FOR UPDATE on a forward-only cursor creates an updatable, lock concurrency, non-scrollable cursor.

Supported Fetch Orientations: All ODBC fetch orientations are supported via the SQLFetchScroll or SQLExtendedFetch interfaces.

Updating the Keyset-Driven Cursor: A keyset-driven cursor is an updatable cursor. The CLI driver appends the FOR UPDATE clause to the query, except when the query is issued as a SELECT ... FOR READ ONLY query, or if the FOR UPDATE clause already exists. The keyset-driven cursor implemented in DB2 for OS/390 is a values concurrency cursor. A values concurrency cursor results in optimistic locking, where locks are not held until an update or delete is attempted. When an update or delete is attempted, the database server compares the previous values the application retrieved to the current values in the underlying table. If the values match, then the update or delete succeeds. If the values do not match, then the operation fails. If failure occurs, the application should query the values again and re-issue the update or delete if it is still applicable.

An application can update a keyset-driven cursor in two ways:

- Issue an UPDATE WHERE CURRENT OF "<cursor name>" or DELETE WHERE CURRENT OF "<cursor name>" using SQLPrepare() with SQLExecute() or SQLExecDirect().
- Use SQLSetPos() or SQLBulkOperations() to update, delete, or add a row to the result set.

Anmerkung: Rows added to a result set via `SQLSetPos()` or `SQLBulkOperations()` are inserted into the table on the server, but are not added to the server's result set. Therefore, these rows are not updatable nor are they sensitive to changes made by other transactions. The inserted rows will appear, however, to be part of the result set, since they are cached on the client. Any triggers that apply to the inserted rows will appear to the application as if they have not been applied. To make the inserted rows updatable, sensitive, and to see the result of applicable triggers, the application must issue the query again to regenerate the result set.

Troubleshooting for Applications Created Before Scrollable Cursor Support: Since scrollable cursor support is new, some ODBC applications that were working with previous releases of UDB for OS/390 or UDB for Unix, Windows, and OS/2 may encounter behavioral or performance changes. This occurs because before scrollable cursors were supported, applications that requested a scrollable cursor would receive a forward-only cursor. To restore an application's previous behavior before scrollable cursor support, set the following configuration keywords in the `db2cli.ini` file:

Tabelle 24. Configuration keyword values restoring application behavior before scrollable cursor support

Configuration Keyword Setting	Description
<code>PATCH2=6</code>	Returns a message that scrollable cursors (both keyset-driven and static) are not supported. CLI automatically downgrades any request for a scrollable cursor to a forward-only cursor.
<code>DisableKeysetCursor=1</code>	Disables both the server-side and client-side keyset-driven scrollable cursors. This can be used to force the CLI driver to give the application a static cursor when a keyset-driven cursor is requested.
<code>UseServerKeysetCursor=0</code>	Disables the server-side keyset-driven cursor for applications that are using the client-side keyset-driven cursor library to simulate a keyset-driven cursor. Only use this option when problems are encountered with the server-side keyset-driven cursor, since the client-side cursor incurs a large amount of overhead and will generally have poorer performance than a server-side cursor.

Using compound SQL

The following note is missing from the book:

Any SQL statement that can be prepared dynamically, other than a query, can be executed as a statement inside a compound statement.

Note: Inside Atomic Compound SQL, `savepoint`, `release savepoint`, and `rollback to savepoint` SQL statements are also disallowed. Conversely, Atomic Compound SQL is disallowed in `savepoint`.

Using Stored Procedures

Writing a Stored Procedure in CLI

Following is an undocumented limitation on CLI stored procedures:

If you are making calls to multiple CLI stored procedures, the application must close the open cursors from one stored procedure before calling the next stored procedure. More specifically, the first set of open cursors must be closed before the next stored procedure tries to open a cursor.

CLI Stored Procedures and Autobinding

The following supplements information in the book:

The CLI/ODBC driver will normally autobind the CLI packages the first time a CLI/ODBC application executes SQL against the database, provided the user has the appropriate privilege or authorization. Autobinding of the CLI packages cannot be performed from within a stored procedure, and therefore will not take place if the very first thing an application does is call a CLI stored procedure. Before running a CLI application that calls a CLI stored procedure against a new DB2 database, you must bind the CLI packages once with this command:

UNIX

```
db2 bind <BNDPATH>/@db2cli.lst blocking all
```

Windows and OS/2

```
db2bind "%DB2PATH%\bnd\@db2cli.lst" blocking
```

The recommended approach is to always bind these packages at the time the database is created to avoid autobind at runtime. Autobind can fail if the user does not have privilege, or if another application tries to autobind at the same time.

Chapter 4. Configuring CLI/ODBC and Running Sample Applications

Configuration Keywords

Disregard the last paragraph in the CURRENTFUNCTIONPATH keyword. The correct information is as follows:

This keyword is used as part of the process for resolving unqualified function and stored procedure references that may have been defined in a schema name other than the current user's schema. The order of the schema names determines the order in which the function and procedure names will be resolved. For more information on function and procedure resolution, refer to the SQL Reference.

Chapter 5. DB2 CLI Functions

SQLBindFileToParam - Bind LOB File Reference to LOB Parameter

The last parameter - IndicatorValue - in the SQLBindFileToParam() CLI function is currently documented as "output (deferred)". It should be "input (deferred)".

SQLNextResult - Associate Next Result Set with Another Statement Handle

The following text should be added to Chapter 5, „DB2 CLI Functions“:

Purpose

Specification: DB2 CLI 7.x

Syntax

```
SQLRETURN SQLNextResult (SQLHSTMT StatementHandle1  
SQLHSTMT StatementHandle2);
```

Function Arguments

Tabelle 25. SQLNextResult Arguments

Data Type	Argument	Use	Description
SQLHSTMT	<i>StatementHandle</i>	input	Statement handle.
SQLHSTMT	<i>StatementHandle</i>	input	Statement handle.

Usage

A stored procedure returns multiple result sets by leaving one or more cursors open after exiting. The first result set is always accessed by using the statement handle that called the stored procedure. If multiple result sets are returned, either `SQLMoreResults()` or `SQLNextResult()` can be used to describe and fetch the result set.

`SQLMoreResults()` is used to close the cursor for the first result set and allow the next result set to be processed, whereas `SQLNextResult()` moves the next result set to *StatementHandle2*, without closing the cursor on *StatementHandle1*. Both functions return `SQL_NO_DATA_FOUND` if there are no result sets to be fetched.

Using `SQLNextResult()` allows result sets to be processed in any order once they have been transferred to other statement handles. Mixed calls to `SQLMoreResults()` and `SQLNextResult()` are allowed until there are no more cursors (open result sets) on *StatementHandle1*.

When `SQLNextResult()` returns `SQL_SUCCESS`, the next result set is no longer associated with *StatementHandle1*. Instead, the next result set is associated with *StatementHandle2*, as if a call to `SQLExecDirect()` had just successfully executed a query on *StatementHandle2*. The cursor, therefore, can be described using `SQLNumResultSets()`, `SQLDescribeCol()`, or `SQLColAttribute()`.

After `SQLNextResult()` has been called, the result set now associated with *StatementHandle2* is removed from the chain of remaining result sets and cannot be used again in either `SQLNextResult()` or `SQLMoreResults()`. This means that for 'n' result sets, `SQLNextResult()` can be called successfully at most 'n-1' times.

If `SQLFreeStmt()` is called with the `SQL_CLOSE` option, or `SQLFreeHandle()` is called with *HandleType* set to `SQL_HANDLE_STMT`, all pending result sets on this statement handle are discarded.

`SQLNextResult()` returns `SQL_ERROR` if *StatementHandle2* has an open cursor or *StatementHandle1* and *StatementHandle2* are not on the same connection. If any errors or warnings are returned, `SQLError()` must always be called on *StatementHandle1*.

Anmerkung: `SQLMoreResults()` also works with a parameterized query with an array of input parameter values specified with `SQLParamOptions()` and `SQLBindParameter()`. `SQLNextResult()`, however, does not support this.

Return Codes

- `SQL_SUCCESS`
- `SQL_SUCCESS_WITH_INFO`
- `SQL_STILL_EXECUTING`
- `SQL_ERROR`
- `SQL_INVALID_HANDLE`
- `SQL_NO_DATA_FOUND`

Diagnostics

Tabelle 26. SQLNextResult SQLSTATES

SQLSTATE	Description	Explanation
40003 08S01	Communication Link failure.	The communication link between the application and data source failed before the function completed.
58004	Unexpected system failure.	Unrecoverable system error.
HY001	Memory allocation failure.	DB2 CLI is unable to allocate the memory required to support execution or completion of the function.
HY010	Function sequence error.	The function was called while in a data-at-execute (<code>SQLParamData()</code> , <code>SQLPutData()</code>) operation. <i>StatementHandle2</i> has an open cursor associated with it. The function was called while within a BEGIN COMPOUND and END COMPOUND SQL operation.
HY013	Unexpected memory handling error.	DB2 CLI was unable to access the memory required to support execution or completion of the function.
HYT00	Time-out expired.	The time-out period expired before the data source returned the result set. Time-outs are only supported on non-multitasking systems such as Windows 3.1 and Macintosh System 7. The time-out period can be set using the <code>SQL_ATTR_QUERY_TIMEOUT</code> attribute for <code>SQLSetConnectAttr()</code> .

Restrictions

Only `SQLMoreResults()` can be used for parameterized queries.

References

- "SQLMoreResults - Determine If There Are More Result Sets" on page 535
- "Returning Result Sets from Stored Procedures" on page 120

Appendix D. Extended Scalar Functions

Date and Time Functions

The following functions are missing from the Date and Time Functions section of Appendix D "Extended Scalar Functions":

DAYOFWEEK_ISO(*date_exp*)

Returns the day of the week in *date_exp* as an integer value in the range 1-7, where 1 represents Monday. Note the difference between this function and the DAYOFWEEK() function, where 1 represents Sunday.

WEEK_ISO(*date_exp*)

Returns the week of the year in *date_exp* as an integer value in the range of 1-53. Week 1 is defined as the first week of the year to contain a Thursday. Therefore, Week1 is equivalent to the first week that contains Jan 4, since Monday is considered to be the first day of the week.

Note that WEEK_ISO() differs from the current definition of WEEK(), which returns a value up to 54. For the WEEK() function, Week 1 is the week containing the first Saturday. This is equivalent to the week containing Jan. 1, even if the week contains only one day.

DAYOFWEEK_ISO() and WEEK_ISO() are automatically available in a database created in Version 7. If a database was created prior to Version 7, these functions may not be available. To make DAYOFWEEK_ISO() and WEEK_ISO() functions available in such a database, use the **db2updb** system command. For more information about **db2updb**, see the "Command Reference" section in these Release Notes.

Appendix K. Using the DB2 CLI/ODBC/JDBC Trace Facility

The sections within this appendix have been updated. See the „Traces“ chapter in the *Troubleshooting Guide* for the most up-to-date information on this trace facility.

Fehlernachrichten

Abrufen der Hilfe zu Nachrichten und SQLSTATE-Werten

Die Hilfe, die im Befehlszeilenprozessor verfügbar ist, enthält neue und aktualisierte Hilfe zu Nachrichten und SQLSTATE-Werten, die im Handbuch *Fehlernachrichten* nicht verfügbar ist. Geben Sie den folgenden Befehl an der Eingabeaufforderung des Betriebssystems ein, um die Nachrichtenhilfe des Befehlszeilenprozessors anzuzeigen:

```
db2 "? XXXnnnn"
```

Dabei gilt Folgendes: XXX steht für das Nachrichtenpräfix und nnnnn steht für die Nachrichtennummer.

Beispiel: db2 "? SQL30081" zeigt Hilfe zu der Nachricht SQL30081 an.

Geben Sie den folgenden Befehl an der Eingabeaufforderung des Betriebssystems ein, um SQLSTATE-Text des Befehlszeilenprozessors anzuzeigen:

```
db2 "? XXXXX"
```

Dabei gilt Folgendes: XXXXX steht für den SQLSTATE-Wert.

Beispiel: db2 "? 428F1" zeigt den Text für SQLSTATE 428F1 an.

Änderung der Zuordnung von SQLCODE-Werten in DB2 Connect

Der Standardwert für die Zuordnung von SQLCODE-Werten für DB2 Connect wurde in Version 7.2 geändert. Wenn eine Host-Datenbank den SQLCODE-Wert -567 zurückgibt, ordnet DB2 Connect dem SQLCODE-Wert jetzt den Wert -551 zu, bevor der SQLCODE-Wert an den DB2-Client zurückgegeben wird.

Neue und geänderte Nachrichten

Die folgende Liste enthält die Nachrichtennummern von Nachrichten, die geändert worden sind, seit das Handbuch *Fehlernachrichten* für DB2 Version 7.1 veröffentlicht wurde. Wenn Sie eine dieser Nachrichten erhalten, während Sie DB2 verwenden, empfangen Sie die korrekte aktualisierte Nachricht. Die Nachricht stimmt allerdings nicht mit den Informationen in dem Handbuch *Fehlernachrichten* überein.

CLI-Nachrichten (Call Level Interface)

CLI0645E

CLI0646E

CLI0647E

DB2-Nachrichten

DB21086I

DB210060E

DB210061E

DB210062E

DB210113E
DB210114E
DB210115E
DB210116E
DB210117E
DB210118E
DB210120E
DB210121E
DB210200I
DB210201I

DBI-Nachrichten

DBI1172E
DBI1793W
DBI1794E
DBI1795E
DBI1796W
DBI1797I

Nachrichten der Data Warehouse-Zentrale

DWC0000I
DWC03504E
DWC08900E
DWC08901E
DWC08902E
DWC08903E
DWC08904E
DWC08907C
DWC08908C
DWC08909C
DWC08910E
DWC08911E
DWC08912E
DWC08913E
DWC08914E
DWC08915E
DWC08917E
DWC08919I
DWC08930E
DWC08931E
DWC08932E
DWC08933E
DWC08934E
DWC08935E
DWC08936W

DWC08937I
DWC08938I
DWC08939I
DWC08940I
DWC08941I
DWC08960I
DWC08961I
DWC08962I
DWC08963I
DWC08964I
DWC08965I
DWC08966E
DWC08967E
DWC08968E
DWC13239E
DWC13300E
DWC13301E
DWC13302E
DWC13304E
DWC13603E
DWC13700E
DWC13701E
DWC13702E
DWC13703E
DWC13705E
DWC13706E
DWC13707E

SQL-Nachrichten

SQL0017N
SQL0056N
SQL0057N
SQL0058N
SQL0097N
SQL0224N
SQL0225N
SQL0227N
SQL0228N
SQL0231W
SQL0243N
SQL0244N
SQL0270N
SQL0301N
SQL0303N
SQL0336N

SQL0348N
SQL0349N
SQL0357N
SQL0358N
SQL0368N
SQL0408N
SQL0423N
SQL0590N
SQL0670N
SQL0845N
SQL0846N
SQL1179W
SQL1186N
SQL1550N
SQL1551N
SQL1552N
SQL1553N
SQL1704N
SQL2077W
SQL2078N
SQL2417N
SQL2426N
SQL2571N
SQL2572N
SQL2573N
SQL2574N
SQL2575N
SQL2576N
SQL4942N
SQL5012N
SQL6583N
SQL20005N
SQL20117N
SQL20121N
SQL20133N
SQL20134N
SQL20135N
SQL20143N
SQL20144N
SQL20145N
SQL20146N
SQL20147N
SQL20148N
SQL20153N
SQL21000N

Korrigierte SQLSTATE-Werte

Tabelle 27.

42630	Eine SQLSTATE- oder SQLCODE-Variable ist in diesem Kontext nicht gültig.
42631	In einer RETURN-Anweisung in einer SQL-Funktion muss ein Ausdruck angegeben werden.
42632	In einer SQL-Funktion oder -Methode muss eine RETURN-Anweisung vorhanden sein.
428F2	In einer RETURN-Anweisung in einer SQL-Funktion muss ein ganzzahliger Ausdruck angegeben werden.
560B7	Beim Einfügen (INSERT) mehrerer Zeilen muss die Syntax eines NEXTVAL-Sequenzausdrucks in jeder Zeile dieselbe sein.

SQL Reference

SQL-Referenz in einer PDF-Datei verfügbar

Im Anhang "Verwenden der DB2-Bibliothek" (Using the DB2 Library) in jedem Handbuch wird angegeben, dass die SQL-Referenz im PDF-Format in zwei Bänden verfügbar ist. Dies ist nicht korrekt.

Obwohl das gedruckte Buch aus zwei Bänden besteht und die beiden Formnummern korrekt sind, gibt es nur eine PDF-Datei, die beide Bände enthält. Der Name dieser PDF-Datei lautet: db2s0x70.

Chapter 3. Language Elements

Naming conventions and Implicit Object Name Qualifications

Add the following note to this section in Chapter 3:

The following names, when used in the context of SQL Procedures, are restricted to the characters allowed in an ordinary identifier, even if the names are delimited:

- condition-name
- label
- parameter-name
- procedure-name
- SQL-variable-name
- statement-name

DATALINK Assignments

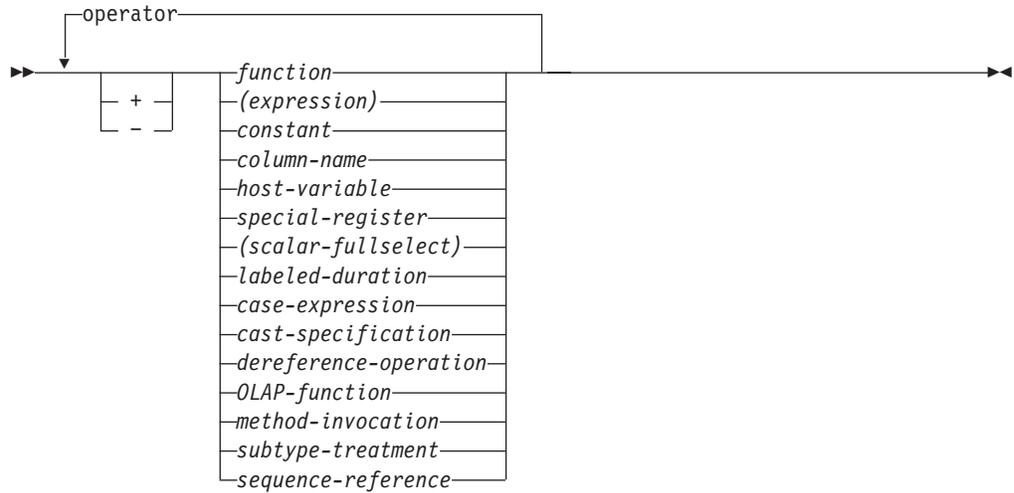
A paragraph in this section has been changed to the following:

Note that the size of a URL parameter or function result is the same on both input or output and is bound by the length of the DATALINK column. However, in some cases the URL value returned has an access token attached. In situations where this is possible, the output location must have sufficient storage space for the access token and the length of the DATALINK column. Hence, the actual length of the comment and URL in its fully expanded form provided on input should be restricted to accommodate the output storage space. If the restricted length is exceeded, this error is raised.

Expressions

Syntax Diagram

The syntax diagram has changed:



operator:



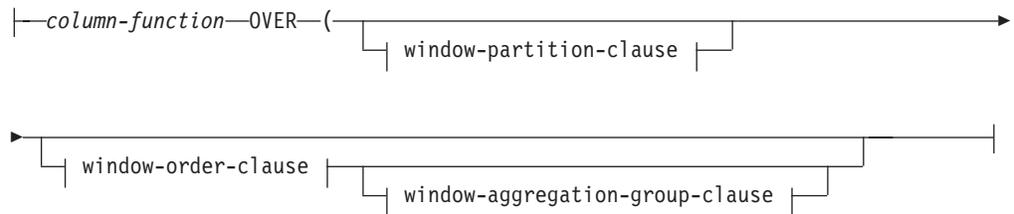
Anmerkungen:

1 || may be used as a synonym for CONCAT.

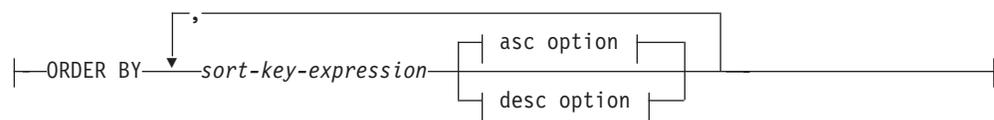
OLAP Functions

The following represents a correction to the "OLAP Functions" section under "Expressions" in Chapter 3.

aggregation-function:



window-order-clause:



asc option:



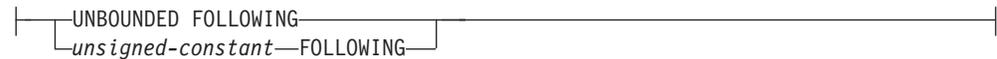
desc option:



window-aggregation-group-clause:



group-end:



In the window-order-clause description:

NULLS FIRST

The window ordering considers null values before all non-null values in the sort order.

NULLS LAST

The window ordering considers null values after all non-null values in the sort order.

In the window-aggregation-group-clause description:

window-aggregation-group-clause

The aggregation group of a row R is a set of rows, defined relative to R in the ordering of the rows of R's partition. This clause specifies the aggregation group. If this clause is not specified, the default is the same as RANGE BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW, providing a cumulative aggregation result.

ROWS

Indicates the aggregation group is defined by counting rows.

RANGE

Indicates the aggregation group is defined by an offset from a sort key.

group-start

Specifies the starting point for the aggregation group. The aggregation

group end is the current row. Specification of the group-start clause is equivalent to a group-between clause of the form "BETWEEN group-start AND CURRENT ROW".

group-between

Specifies the aggregation group start and end based on either ROWS or RANGE.

group-end

Specifies the ending point for the aggregation group. The aggregation group start is the current row. Specification of the *group-end* clause is equivalent to a *group-between* clause of the form "BETWEEN CURRENT ROW AND *group-end*".

UNBOUNDED PRECEDING

Includes the entire partition preceding the current row. This can be specified with either ROWS or RANGE. Also, this can be specified with multiple sort-key-expressions in the window-order-clause.

UNBOUNDED FOLLOWING

Includes the entire partition following the current row. This can be specified with either ROWS or RANGE. Also, this can be specified with multiple sort-key-expressions in the window-order-clause.

CURRENT ROW

Specifies the start or end of the aggregation group based on the current row. If ROWS is specified, the current row is the aggregation group boundary. If RANGE is specified, the aggregation group boundary includes the set of rows with the same values for the *sort-key-expressions* as the current row. This clause cannot be specified in *group-bound2* if *group-bound1* specifies *value* FOLLOWING.

value **PRECEDING**

Specifies either the range or number of rows preceding the current row. If ROWS is specified, then *value* is a positive integer indicating a number of rows. If RANGE is specified, then the data type of *value* must be comparable to the type of the sort-key-expression of the window-order-clause. There can only be one sort-key-expression, and the data type of the sort-key-expression must allow subtraction. This clause cannot be specified in *group-bound2* if *group-bound1* is CURRENT ROW or *value* FOLLOWING.

value **FOLLOWING**

Specifies either the range or number of rows following the current row. If ROWS is specified, then *value* is a positive integer indicating a number of rows. If RANGE is specified, then the data type of *value* must be comparable to the type of the sort-key-expression of the window-order-clause. There can only be one sort-key-expression, and the data type of the sort-key-expression must allow addition.

Sequence Reference

The following information should be added to the end of the Expressions section (after "Subtype Treatment").

sequence-reference:



nextval-expression:

```
|—NEXTVAL FOR—sequence-name—|
```

prevval-expression:

```
|—PREVVAL FOR—sequence-name—|
```

NEXTVAL FOR *sequence-name*

A NEXTVAL expression returns the next value for the sequence specified by *sequence-name*.

PREVVAL FOR *sequence-name*

A PREVVAL expression returns the most recently generated value for the specified sequence for a previous statement within the current session. This value can be repeatedly referenced using PREVVAL expressions specifying the name of the sequence. There may be multiple instances of PREVVAL expressions specifying the same sequence name within a single statement and they all return the same value.

A PREVVAL expression can only be used if a NEXTVAL expression specifying the same sequence name has already been referenced in the current user session (in the current or a previous transaction) (SQLSTATE 51035).

Anmerkung:

- A new sequence number is generated when a NEXTVAL expression specifies the name of the sequence. However, if there are multiple instances of a NEXTVAL expression specifying the same sequence name within a query, the counter for the sequence is incremented only once for each row of the result.
- The most recently generated value for a sequence can be repeatedly referenced using a PREVVAL expression specifying the name of the sequence. There may be multiple instances of a PREVVAL expression specifying the same sequence name within a single statement.
- The same sequence number can be used as a unique key value in two separate tables by referencing the sequence number with a NEXTVAL expression for the first row (this generates the sequence value), and a PREVVAL expression for the other rows (this instance of PREVVAL refers to the sequence value generated by the NEXTVAL expression in the previous statement), as shown below:

```
INSERT INTO order(orderno, custno)
VALUES (NEXTVAL FOR order_seq, 123456);
```

```
INSERT INTO line_item (orderno, partno, quantity)
VALUES (PREVVAL FOR order_seq, 987654, 1);
```

- Examples of where NEXTVAL and PREVVAL expressions can be specified are:
 - select-statement or SELECT INTO statement: within the select-clause as long as the statement does not contain a DISTINCT

- keyword, a GROUP BY clause, an ORDER BY clause, a UNION keyword, an INTERSECT keyword, or EXCEPT keyword
- INSERT statement: within a VALUES clause
- INSERT statement: within the select-clause of the fullselect
- UPDATE statement: within the select-clause of the fullselect of an expression in the SET clause (either searched or positioned UPDATE statement)
- VALUES INTO statement: within the select-clause of the fullselect of an expression
- Examples of where NEXTVAL and PREVVVAL expressions cannot be specified (SQLSTATE 428F9) are:
 - join condition of a full outer join
 - DEFAULT value for a column in a CREATE TABLE or ALTER TABLE statement
 - generated column definition in a CREATE TABLE or ALTER TABLE statement
 - condition of a CHECK constraint
 - CREATE TRIGGER statement
 - CREATE VIEW statement
 - CREATE METHOD statement
 - CREATE FUNCTION statement.
- In addition, a NEXTVAL expression cannot be specified (SQLSTATE 428F9) in :
 - CASE expression
 - parameter list of an aggregate function
 - subquery
 - SELECT statement that contains a DISTINCT operator
 - join condition of join
 - GROUP BY clause of a SELECT statement
 - SELECT statement that is combined with another SELECT statement using the UNION, INTERSECT, or EXCEPT set operator
 - nested table expression
 - parameter list of a table function
 - WHERE clause of a SELECT, DELETE, or UPDATE statement
 - ORDER BY clause
 - parameter list of a CALL statement.
- When a value is generated for a sequence, that value is consumed, and the next time that a value is needed, a new value will be generated. This is true even when the statement containing the NEXTVAL expression fails.
- If an INSERT statement includes a NEXTVAL expression in the VALUES list for the column, and if some error occurs at some point during the execution of the INSERT (it could be a problem in generating the next sequence value, or a problem with the value for another column), then an insertion failure occurs, and the value generated for the sequence is considered to be consumed. In some cases, reissuing the same INSERT statement might lead to success. For example, consider an error that is the result of the existence of a unique index for the column for which NEXTVAL was used and

the sequence value generated already exists in the index. It is possible that the next value generated for the sequence is a value that does not exist in the index and so the subsequent INSERT would succeed.

- If in generating a value for a sequence, the maximum value for the sequence is exceeded (or the minimum value for a descending sequence) and cycles are not permitted, then an error occurs (SQLSTATE 23522). In this case, the user could ALTER the sequence to extend the range of acceptable values, or enable cycles for the sequence, or DROP and CREATE a new sequence with a different data type that has a larger range of values.

For example, a sequence may have been defined with a data type of SMALLINT, and eventually the sequence runs out of assignable values. To redefine the sequence as INTEGER, you would need to drop and recreate the sequence with the new definition.

- A reference to PREVVAL in a select statement of a cursor refers to a value that was generated for the specified sequence prior to the opening of the cursor. However, closing the cursor can affect the values returned by PREVVAL for the specified sequence in subsequent statements, or even for the same statement in the event that the cursor is reopened. This would be the case when the select statement of the cursor included a reference to NEXTVAL for the same sequence name.

Examples:: These examples assume that there is a table called "order" and that a sequence called "order_seq" is created as follows:

```
CREATE SEQUENCE order_seq
  START WITH 1
  INCREMENT BY 1
  NOMAXVALUE
  NOCYCLE
  CACHE 24
```

- Some examples of how to generate an "order_seq" sequence number with a NEXTVAL expression for the sequence created above:

```
INSERT INTO order(orderno, custno)
  VALUES (NEXTVAL FOR order_seq, 123456);
```

or,

```
UPDATE order
  SET orderno = NEXTVAL FOR order_seq
  WHERE custno = 123456;
```

or,

```
VALUES NEXTVAL FOR order_seq INTO :hv_seq;
```

Chapter 4. Functions

Enabling the New Functions and Procedures

Version 7 FixPaks deliver new SQL built-in scalar functions. Refer to the *SQL Reference* updates for a description of these new functions. The new functions are not automatically enabled on each database when the database server code is upgraded to the new service level. To enable these new functions, the system administrator must issue the command **db2updv7**, specifying each database at the server.

This command makes an entry in the database that ensures that database objects created prior to executing this command use existing function signatures that may match the new function signatures.

For information on enabling the MQSeries functions (those defined in the MQDB2 schema), see „MQSeries“ auf Seite 191.

Scalar Functions

ABS or ABSVAL

►► $\left[\begin{array}{l} \text{ABS} \\ \text{ABSVAL} \end{array} \right] (\textit{expression})$ —————►►

The schema is SYSIBM.

This function was first available in FixPak 2 of Version 7.1.

Anmerkung: The SYSFUN version of the ABS (or ABSVAL) function continues to be available.

Returns the absolute value of the argument.

The argument is an expression that returns a value of any built-in numeric data type.

The result of the function has the same data type and length attribute as the argument. If the argument can be null or the database is configured with DFT_SQLMA-THWARN set to yes, then the result can be null; if the argument is null, the result is the null value.

For example:

ABS(-51234)

returns an INTEGER with a value of 51234.

DECRYPT_BIN and DECRYPT_CHAR

►► $\left[\begin{array}{l} \text{DECRYPT_BIN} \\ \text{DECRYPT_CHAR} \end{array} \right]$ —————►►

►► $(\textit{--encrypted-data} \left[\textit{, --password-string-expression} \right])$ —————►►

The schema is SYSIBM.

This function was first available in FixPak 3 of Version 7.1.

The DECRYPT_BIN and DECRYPT_CHAR functions return a value that is the result of decrypting *encrypted-data*. The password used for decryption is either the *password-string-expression* value or the ENCRYPTION PASSWORD value (as assign-

ned using the SET ENCRYPTION PASSWORD statement). The DECRYPT_BIN and DECRYPT_CHAR functions can only decrypt values that were encrypted using the ENCRYPT function (SQLSTATE 428FE).

encrypted-data

An expression that returns a CHAR FOR BIT DATA or VARCHAR FOR BIT DATA value that is a complete, encrypted data string that was encrypted using the ENCRYPT function.

password-string-expression

An expression that returns a CHAR or VARCHAR value with at least 6 bytes and no more than 127 bytes (SQLSTATE 428FC). This should be the same password used to encrypt the data or decryption will result in an error (SQLSTATE 428FD). If the value of the password argument is null or not provided, the data will be decrypted using the ENCRYPTION PASSWORD value, which must have been set for the session (SQLSTATE 51039).

The result of the DECRYPT_BIN function is VARCHAR FOR BIT DATA. The result of the DECRYPT_CHAR function is VARCHAR. If the *encrypted-data* included a hint, the hint is not returned by the function. The length attribute of the result is the length attribute of the data type of *encrypted-data* minus 8 bytes. The actual length of the value returned by the function will match the length of the original string that was encrypted. If the *encrypted-data* includes bytes beyond the encrypted string, these bytes are not returned by the function. If the first argument can be null, the result can be null; if the first argument is null, the result is the null value.

If the data is decrypted on a different system using a code page other than the code page in which the encryption took place, it is possible that expansion may occur when converting the decrypted value to the database code page. In such situations, the *encrypted-data* value should be cast to a VARCHAR string with a larger number of bytes.

Also see „ENCRYPT“ and „GETHINT“ auf Seite 389 for additional information on using this function.

Examples:

Example 1: This example uses the ENCRYPTION PASSWORD value to hold the encryption password.

```
SET ENCRYPTION PASSWORD = 'Ben123';
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832');
SELECT DECRYPT_CHAR(SSN)
FROM SSN;
```

The value returned is '289-46-8832'.

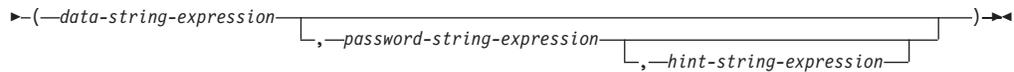
Example 2: This example explicitly passes the encryption password.

```
SELECT DECRYPT_CHAR(SSN, 'Ben123')
FROM SSN;
```

The value returned is '289-46-8832'.

ENCRYPT

►—ENCRYPT—►



The schema is SYSIBM.

This function was first available in FixPak 3 of Version 7.1.

The ENCRYPT function returns a value that is the result of encrypting *data-string-expression*. The password used for encryption is either the *password-string-expression* value or the ENCRYPTION PASSWORD value (as assigned using the SET ENCRYPTION PASSWORD statement).

data-string-expression

An expression that returns a CHAR or VARCHAR value to be encrypted. The length attribute for the data type of *data-string-expression* is limited to 32663 without a *hint-string-expression* argument and 32631 when the *hint-string-expression* argument is specified (SQLSTATE 42815).

password-string-expression

An expression that returns a CHAR or VARCHAR value with at least 6 bytes and no more than 127 bytes (SQLSTATE 428FC). The value represents the password used to encrypt the *data-string-expression*. If the value of the password argument is null or not provided, the data will be encrypted using the ENCRYPTION PASSWORD value, which must have been set for the session (SQLSTATE 51039).

hint-string-expression

An expression that returns a CHAR or VARCHAR value up to 32 bytes that will help data owners remember passwords (for example, 'Ocean' as a hint to remember 'Pacific'). If a hint value is given, the hint is embedded into the result and can be retrieved using the GETHINT function. If this argument is null or not provided, no hint will be embedded in the result.

The result data type of the function is VARCHAR FOR BIT DATA.

The length attribute of the result is:

- When the optional hint parameter is specified, the length attribute of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary + 32 bytes for the hint length.
- With no hint parameter, the length attribute of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary.

If the first argument can be null, the result can be null; if the first argument is null, the result is the null value.

Notice that the encrypted result is longer than the *data-string-expression* value. Therefore, when assigning encrypted values, ensure that the target is declared with sufficient size to contain the entire encrypted value.

Notes:

- **Encryption Algorithm:** The internal encryption algorithm used is RC2 block cipher with padding, the 128-bit secret key is derived from the password using a MD2 message digest.
- **Encryption Passwords and Data:** It is the user's responsibility to perform password management. Once the data is encrypted only the password used to

encrypt it can be used to decrypt it (SQLSTATE 428FD). Be careful when using CHAR variables to set password values as they may be padded with blanks. The encrypted result may contain null terminator and other non-printable characters.

Table Column Definition: When defining columns and types to contain encrypted data always calculate the length attribute as follows. For encrypted data with no hint:

Maximum length of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary = encrypted data column length.

For encrypted data with embedded hint :

Maximum length of the non-encrypted data + 8 bytes + the number of bytes to the next 8 byte boundary + 32 bytes for the hint length = encrypted data column length.

Any assignment or cast to a length shorter than the suggested data length may result in failed decryption in the future and **lost data**. Blanks are valid encrypted data values that may be truncated when stored in a column that is too short.

Sample Column Length Calculations

Maximum length of non-encrypted data	6 bytes
8 bytes	8 bytes
Number of bytes to the next 8 byte boundary	2 bytes

Encrypted data column length	16 bytes
Maximum length of non-encrypted data	32 bytes
8 bytes	8 bytes
Number of bytes to the next 8 byte boundary	8 bytes

Encrypted data column length	48 bytes

- **Administration of encrypted data:** Encrypted data can only be decrypted on servers that support the decryption functions that correspond to the ENCRYPT function. Hence, replication of columns with encrypted data should only be done to servers that support the DECRYPT_BIN or DECRYPT_CHAR function.

Also see „DECRYPT_BIN and DECRYPT_CHAR“ auf Seite 386 and „GETHINT“ for additional information on using this function.

Examples:

Example 1: This example uses the ENCRYPTION PASSWORD value to hold the encryption password.

```
SET ENCRYPTION PASSWORD = 'Ben123';
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832');
```

Example 2: This example explicitly passes the encryption password.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832','Ben123',');
```

Example 3: The hint 'Ocean' is stored to help the user remember the encryption password of 'Pacific'.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832','Pacific','Ocean');
```

GETHINT

►►—GETHINT—(—encrypted-data—)—————►►

The schema is SYSIBM.

This function was first available in FixPak 3 of Version 7.1.

The GETHINT function will return the password hint if one is found in the *encrypted-data*. A password hint is a phrase that will help data owners remember passwords (For example, 'Ocean' as a hint to remember 'Pacific').

encrypted-data

An expression that returns a CHAR FOR BIT DATA or VARCHAR FOR BIT DATA value that is a complete, encrypted data string that was encrypted using the ENCRYPT function (SQLSTATE 428FE).

The result of the function is VARCHAR(32). The result can be null; if the hint parameter was not added to the *encrypted-data* by the ENCRYPT function or the first argument is null, the result is the null value.

Also see „DECRYPT_BIN and DECRYPT_CHAR“ auf Seite 386 and „ENCRYPT“ auf Seite 387 for additional information on using this function.

Example:

In this example the hint 'Ocean' is stored to help the user remember the encryption password 'Pacific'.

```
INSERT INTO EMP (SSN) VALUES ENCRYPT('289-46-8832', 'Pacific', 'Ocean');
SELECT GETHINT(SSN)
FROM EMP;
```

The value returned is 'Ocean'.

IDENTITY_VAL_LOCAL

►►—IDENTITY_VAL_LOCAL—(—)—————►►

The schema is SYSIBM.

This procedure was first available in FixPak 3 of Version 7.1.

The IDENTITY_VAL_LOCAL function is a non-deterministic function that returns the most recently assigned value for an identity column, where the assignment occurred as a result of a single row INSERT statement using a VALUES clause. The function has no input parameters.

The result is a DECIMAL(31,0), regardless of the actual data type of the corresponding identity column.

The value returned by the function is the value assigned to the identity column of the table identified in the most recent single row INSERT statement. The INSERT statement must be made using a VALUES clause on a table containing an identity column. Also, the INSERT statement must be issued at the same level¹ (that is, the value is available locally at the level it was assigned, until it is replaced by the next assigned value).

1. A new level is initiated each time a trigger, function, or stored procedure is invoked.

The assigned value is either a value supplied by the user (if the identity column is defined as GENERATED BY DEFAULT), or an identity value generated by DB2.

The function returns a null value in the following situations:

- When a single row INSERT statement with a VALUES clause has not been issued at the current processing level for a table containing an identity column.
- When a COMMIT or ROLLBACK of a unit of work has occurred since the most recent INSERT statement that assigned a value².

The result of the function is not affected by the following:

- A single row INSERT statement with a VALUES clause for a table without an identity column.
- A multiple row INSERT statement with a VALUES clause.
- An INSERT statement with a fullselect.
- A ROLLBACK TO SAVEPOINT statement.

Notes:

- Expressions in the VALUES clause of an INSERT statement are evaluated prior to the assignments for the target columns of the INSERT statement. Thus, an invocation of an IDENTITY_VAL_LOCAL function inside the VALUES clause of an INSERT statement will use the most recently assigned value for an identity column from a previous INSERT statement. The function returns the null value if no previous single row INSERT statement with a VALUES clause for a table containing an identity column has been executed within the same level as the IDENTITY_VAL_LOCAL function.
- The identity column value of the table for which the trigger is defined can be determined within a trigger, by referencing the trigger transition variable for the identity column.
- The result of invoking the IDENTITY_VAL_LOCAL function from within the trigger condition of an insert trigger is a null value.
- It is possible that multiple before or after insert triggers exist for a table. In this case each trigger is processed separately, and identity values assigned by one triggered action are not available to other triggered actions using the IDENTITY_VAL_LOCAL function. This is true even though the multiple triggered actions are conceptually defined at the same level.
- It is not generally recommended to use the IDENTITY_VAL_LOCAL function in the body of a before insert trigger. The result of invoking the IDENTITY_VAL_LOCAL function from within the triggered action of a before insert trigger is the null value. The value for the identity column of the table for which the trigger is defined cannot be obtained by invoking the IDENTITY_VAL_LOCAL function within the triggered action of a before insert trigger. However, the value for the identity column can be obtained in the triggered action, by referencing the trigger transition variable for the identity column.
- The result of invoking the IDENTITY_VAL_LOCAL function from within the triggered action of an after insert trigger³ is the value assigned to an identity column of the table identified in the most recent single row INSERT statement invoked in the same triggered action that had a VALUES clause for a table containing an identity column. If a single row INSERT statement with a VALUES clause for a table containing an identity column was not executed within the

2. Unless the automatic commit is turned off, interfaces that automatically commit after each statement will return a null value when the function is invoked in separate statements.

3. This applies to both FOR EACH ROW and FOR EACH STATEMENT after insert triggers.

same triggered action, prior to the invocation of the `IDENTITY_VAL_LOCAL` function, then the function returns a null value.

- Since the results of the `IDENTITY_VAL_LOCAL` function are not deterministic, the result of an invocation of the `IDENTITY_VAL_LOCAL` function within the `SELECT` statement of a cursor can vary for each `FETCH` statement.
- The assigned value is the value actually assigned to the identity column (that is, the value that would be returned on a subsequent `SELECT` statement). This value is not necessarily the value provided in the `VALUES` clause of the `INSERT` statement, or a value generated by DB2. The assigned value could be a value specified in a `SET` transition variable statement, within the body of a before insert trigger, for a trigger transition variable associated with the identity column.
- The value returned by the function is unpredictable following a failed single row `INSERT` with a `VALUES` clause into a table with an identity column. The value may be the value that would have been returned from the function had it been invoked prior to the failed `INSERT`, or it may be the value that would have been assigned had the `INSERT` succeeded. The actual value returned depends on the point of failure and is therefore unpredictable.

Examples:

Example 1: Set the variable `IVAR` to the value assigned to the identity column in the `EMPLOYEE` table. If this insert is the first into the `EMPLOYEE` table, then `IVAR` would have a value of 1.

```
CREATE TABLE EMPLOYEE
  (EMPNO  INTEGER GENERATED ALWAYS AS IDENTITY,
   NAME CHAR(30),
   SALARY DECIMAL(5,2),
   DEPTNO SMALLINT)
```

Example 2: An `IDENTITY_VAL_LOCAL` function invoked in an `INSERT` statement returns the value associated with the previous single row `INSERT` statement, with a `VALUES` clause for a table with an identity column. Assume for this example that there are two tables, `T1` and `T2`. Both `T1` and `T2` have an identity column named `C1`. DB2 generates values in sequence, starting with 1, for the `C1` column in table `T1`, and values in sequence, starting with 10, for the `C1` column in table `T2`.

```
CREATE TABLE T1
  (C1 INTEGER GENERATED ALWAYS AS IDENTITY,
   C2 INTEGER),
CREATE TABLE T2
  (C1 DECIMAL(15,0) GENERATED BY DEFAULT AS IDENTITY
   (START WITH 10),
   C2 INTEGER),
INSERT INTO T1 (C2) VALUES (5),
INSERT INTO T1 (C2) VALUES (6),
SELECT * FROM T1
```

which gives a result of:

C1	C2
1	5
2	6

and now, declaring the function for the variable `IVAR`:

```
VALUES IDENTITY_VAL_LOCAL() INTO :IVAR
```

At this point, the `IDENTITY_VAL_LOCAL` function would return a value of 2 in `IVAR`, because that was the value most recently assigned by `DB2`. The following `INSERT` statement inserts a single row into `T2`, where column `C2` gets a value of 2 from the `IDENTITY_VAL_LOCAL` function.

```
INSERT INTO T2 (C2) VALUES (IDENTITY_VAL_LOCAL());
SELECT * FROM T2
WHERE C1 = DECIMAL(IDENTITY_VAL_LOCAL(),15,0)
```

returning a result of:

C1	C2
----- 10.	----- 2

Invoking the `IDENTITY_VAL_LOCAL` function after this insert results in a value of 10, which is the value generated by `DB2` for column `C1` of `T2`.

In a nested environment involving a trigger, use the `IDENTITY_VAL_LOCAL` function to retrieve the identity value assigned at a particular level, even though there might have been identity values assigned at lower levels. Assume that there are three tables, `EMPLOYEE`, `EMP_ACT`, and `ACCT_LOG`. There is an after insert trigger defined on `EMPLOYEE` that results in additional inserts into the `EMP_ACT` and `ACCT_LOG` tables.

```
CREATE TABLE EMPLOYEE
  (EMPNO SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 1000),
  NAME CHAR(30),
  SALARY DECIMAL(5,2),
  DEPTNO SMALLINT);

CREATE TABLE EMP_ACT
  (ACNT_NUM SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 1),
  EMPNO SMALLINT);

CREATE TABLE ACCT_LOG
  (ID SMALLINT GENERATED ALWAYS AS IDENTITY (START WITH 100),
  ACNT_NUM SMALLINT,
  EMPNO SMALLINT);

CREATE TRIGGER NEW_HIRE
AFTER INSERT ON EMPLOYEE
REFERENCING NEW AS NEW_EMP
FOR EACH ROW MODE DB2SQL
BEGIN ATOMIC      INSERT INTO EMP_ACT (EMPNO)
VALUES (NEW_EMP.EMPNO);
INSERT INTO ACCT_LOG (ACNT_NUM EMPNO)
VALUES (IDENTITY_VAL_LOCAL(), NEW_EMP.EMPNO);
END
```

The first triggered `INSERT` statement inserts a row into the `EMP_ACT` table. This `INSERT` statement uses a trigger transition variable for the `EMPNO` column of the `EMPLOYEE` table, to indicate that the identity value for the `EMPNO` column of the `EMPLOYEE` table is to be copied to the `EMPNO` column of the `EMP_ACT` table. The `IDENTITY_VAL_LOCAL` function could not be used to obtain the value assigned to the `EMPNO` column of the `EMPLOYEE` table. This is because an `INSERT` statement has not been issued at this level of the nesting, and as such, if the `IDENTITY_VAL_LOCAL` function were invoked in the `VALUES` clause of the `INSERT` for `EMP_ACT`, then it would return a null value. This `INSERT` statement for the `EMP_ACT` table also results in the generation of a new identity column value for the `ACNT_NUM` column.

A second triggered INSERT statement inserts a row into the ACCT_LOG table. This statement invokes the IDENTITY_VAL_LOCAL function to indicate that the identity value assigned to the ACNT_NUM column of the EMP_ACT table in the previous INSERT statement in the triggered action is to be copied to the ACNT_NUM column of the ACCT_LOG table. The EMPNO column is assigned the same value as the EMPNO column of EMPLOYEE table.

From the invoking application (that is, the level at which the INSERT to EMPLOYEE is issued), set the variable IVAR to the value assigned to the EMPNO column of the EMPLOYEE table by the original INSERT statement.

```
INSERT INTO EMPLOYEE (NAME, SALARY, DEPTNO)
VALUES ('Rupert', 989.99, 50);
```

The contents of the three tables after processing the original INSERT statement and all of the triggered actions are:

```
SELECT EMPNO, SUBSTR(NAME,10) AS NAME, SALARY, DEPTNO
FROM EMPLOYEE;
```

EMPNO	NAME	SALARY	DEPTNO
1000	Rupert	989.99	50

```
SELECT ACNT_NUM, EMPNO
FROM EMP_ACT;
```

ACNT_NUM	EMPNO
1	1000

```
SELECT * FROM ACCT_LOG;
```

ID	ACNT_NUM	EMPNO
100	1	1000

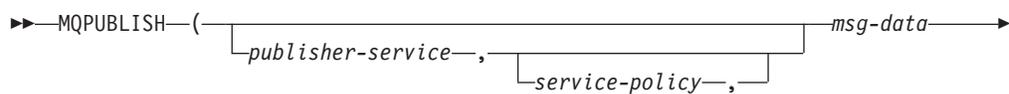
The result of the IDENTITY_VAL_LOCAL function is the most recently assigned value for an identity column at the same nesting level. After processing the original INSERT statement and all of the triggered actions, the IDENTITY_VAL_LOCAL function returns a value of 1000, because this is the value assigned to the EMPNO column of the EMPLOYEE table. The following VALUES statement results in setting IVAR to 1000. The insert into the EMP_ACT table (which occurred after the insert into the EMPLOYEE table and at a lower nesting level) has no effect on what is returned by this invocation of the IDENTITY_VAL_LOCAL function.

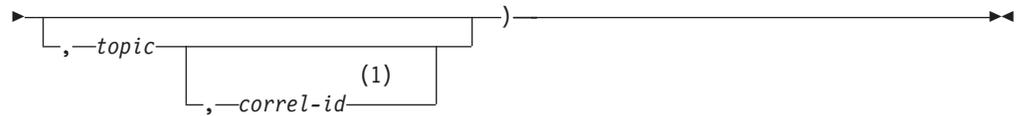
```
VALUES IDENTITY_VAL_LOCAL() INTO :IVAR;
```

LCASE and UCASE (Unicode)

In a Unicode database, the entire repertoire of Unicode characters is uppercase (or lowercase) based on the Unicode properties of these characters. Double-wide versions of ASCII characters, as well as Roman numerals, now convert to upper or lower case correctly.

MQPUBLISH





Anmerkungen:

- 1 The *correl-id* cannot be specified unless a *service* and a *policy* are previously defined.

The schema is MQDB2.

The MQPUBLISH function publishes data to MQSeries. This function requires the installation of either MQSeries Publish/Subscribe or MQSeries Integrator. Please consult www.ibm.com/software/MQSeries for further details.

The MQPUBLISH function publishes the data contained in *msg-data* to the MQSeries publisher specified in *publisher-service*, and using the quality of service policy defined by *service-policy*. An optional topic for the message can be specified, and an optional user-defined message correlation identifier may also be specified. The function returns a value of '1' if successful or a '0' if unsuccessful.

publisher-service

A string containing the logical MQSeries destination where the message is to be sent. If specified, the *publisher-service* must refer to a publisher Service Point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *publisher-service* is not specified, then the DB2.DEFAULT.PUBLISHER will be used. The maximum size of *publisher-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy to be used in handling of this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

msg-data

A string expression containing the data to be sent via MQSeries. The maximum size is 4000 characters.

topic

A string expression containing the topic for the message publication. If no topic is specified, none will be associated with the message. The maximum size of *topic* is 40 characters. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

correl-id

An optional string expression containing a correlation identifier to be associated with this message. The *correl-id* is often specified in request and reply sce-

narios to associate requests with replies. If not specified, no correlation id will be added to the message. The maximum size of *correl-id* is 24 characters.

Example 1: This example publishes the string "Testing 123" to the default publisher service (DB2.DEFAULT.PUBLISHER) using the default policy (DB2.DEFAULT.POLICY). No correlation identifier or topic is specified for the message.

```
VALUES MQPUBLISH('Testing 123')
```

Example 2: This example publishes the string "Testing 345" to the publisher service "MYPUBLISHER" under the topic "TESTS". The default policy is used and no correlation identifier is specified.

```
VALUES MQPUBLISH('MYPUBLISHER','Testing 345', 'TESTS')
```

Example 3: This example publishes the string "Testing 678" to the publisher service "MYPUBLISHER" using the policy "MYPOLICY" with a correlation identifier of "TEST1". The message is published with topic "TESTS".

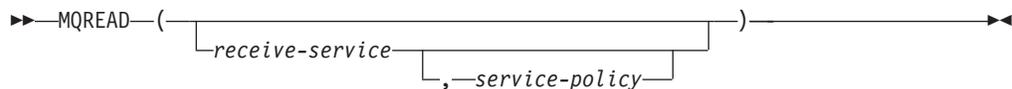
```
VALUES MQPUBLISH('MYPUBLISHER','MYPOLICY','Testing 678','TESTS','TEST1')
```

Example 4: This example publishes the string "Testing 901" to the publisher service "MYPUBLISHER" under the topic "TESTS" using the default policy (DB2.DEFAULT.POLICY) and no correlation identifier.

```
VALUES MQPUBLISH('Testing 901','TESTS')
```

All examples return the value '1' if successful.

MQREAD



The schema is MQDB2.

The MQREAD function returns a message from the MQSeries location specified by *receive-service*, using the quality of service policy defined in *service-policy*. Executing this operation does not remove the message from the queue associated with *receive-service*, but instead returns the message at the head of the queue. The return value is a VARCHAR(4000) containing the message. If no messages are available to be returned, a NULL is returned.

receive-service

A string containing the logical MQSeries destination from where the message is to be received. If specified, the *receive-service* must refer to a Service Point defined in the AMT.XML repository file. A service point is a logical end-point from where a message is sent or received. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in handling this message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service

options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

Examples:

Example 1: This example reads the message at the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY).

VALUES MQREAD()

Example 2: This example reads the message at the head of the queue specified by the service "MYSERVICE" using the default policy (DB2.DEFAULT.POLICY).

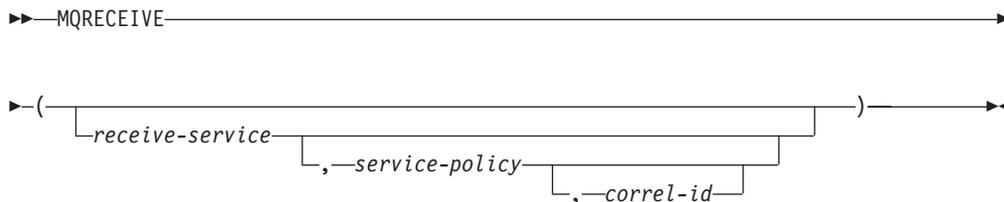
VALUES MQREAD('MYSERVICE')

Example 3: This example reads the message at the head of the queue specified by the service "MYSERVICE", and using the policy "MYPOLICY".

VALUES MQREAD('MYSERVICE', 'MYPOLICY')

All of these examples return the contents of the message as a VARCHAR(4000) if successful. If no messages are available, then a NULL is returned.

MQRECEIVE



The schema is MQDB2.

The MQRECEIVE function returns a message from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation removes the message from the queue associated with *receive-service*. If the *correl-id* is specified, then the first message with a matching correlation identifier will be returned. If *correl-id* is not specified, then the message at the head of the queue will be returned. The return value is a VARCHAR(4000) containing the message. If no messages are available to be returned, a NULL is returned.

receive-service

A string containing the logical MQSeries destination from which the message is received. If specified, the *receive-service* must refer to a Service Point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE is used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy to be used in the handling of this message. If specified, the *service-policy* must refer to a Policy defined in the AMT XML repository file⁴. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY is used. The maximum size of *service-policy* is 48 characters.

correl-id

A string containing an optional correlation identifier to be associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be specified. The maximum size of *correl-id* is 24 characters.

Examples:

Example 1: This example receives the message at the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY).

VALUES MQRECEIVE()

Example 2: This example receives the message at the head of the queue specified by the service "MYSERVICE" using the default policy (DB2.DEFAULT.POLICY).

VALUES MQRECEIVE('MYSERVICE')

Example 3: This example receives the message at the head of the queue specified by the service "MYSERVICE" using the policy "MYPOLICY".

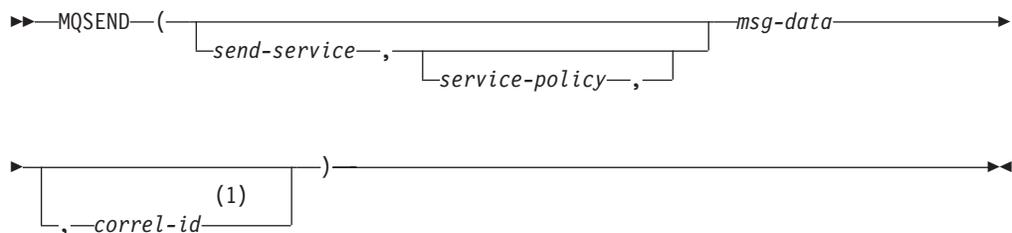
VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY')

Example 4: This example receives the first message with a correlation id that matches '1234' from the head of the queue specified by the service "MYSERVICE" using the policy "MYPOLICY".

VALUES MQRECEIVE('MYSERVICE', 'MYPOLICY', '1234')

All these examples return the contents of the message as a VARCHAR(4000) if successful. If no messages are available, a NULL will be returned.

MQSEND



Anmerkungen:

- 1 The *correl-id* cannot be specified unless a *service* and a *policy* are previously defined.

4. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details.

The schema is MQDB2.

The MQSEND function sends the data contained in *msg-data* to the MQSeries location specified by *send-service*, using the quality of service policy defined by *service-policy*. An optional user defined message correlation identifier may be specified by *correl-id*. The function returns a value of '1' if successful or a '0' if unsuccessful.

msg-data

A string expression containing the data to be sent via MQSeries. The maximum size is 4000 characters.

send-service

A string containing the logical MQSeries destination where the message is to be sent. If specified, the *send-service* refers to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message may be sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *send-service* is not specified, then the value of DB2.DEFAULT.SERVICE is used. The maximum size of *send-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in handling of this message. If specified, the *service-policy* must refer to a service policy defined in the AMT XML repository file. A Service Policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then a default value of DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

correl-id

An optional string containing a correlation identifier associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id will be sent. The maximum size of *correl-id* is 24 characters.

Examples:

Example 1: This example sends the string "Testing 123" to the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY), with no correlation identifier.

```
VALUES MQSEND('Testing 123')
```

Example 2: This example sends the string "Testing 345" to the service "MYSERVICE", using the policy "MYPOLICY", with no correlation identifier.

```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 345')
```

Example 3: This example sends the string "Testing 678" to the service "MYSERVICE", using the policy "MYPOLICY", with correlation identifier "TEST3".

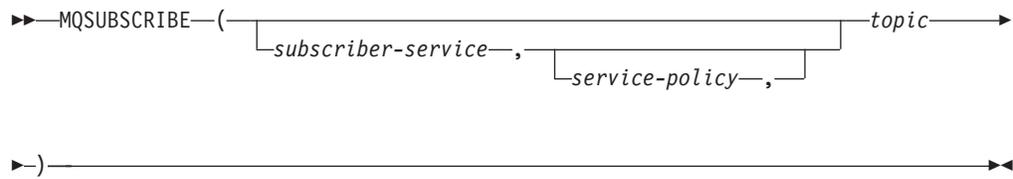
```
VALUES MQSEND('MYSERVICE','MYPOLICY','Testing 678','TEST3')
```

Example 4: This example sends the string "Testing 901" to the service "MYSERVICE", using the default policy (DB2.DEFAULT.POLICY), and no correlation identifier.

```
VALUES MQSEND('MYSERVICE','Testing 901')
```

All examples return a scalar value of '1' if successful.

MQSUBSCRIBE



The schema is MQDB2.

The MQSUBSCRIBE function is used to register interest in MQSeries messages published on a specified topic. The *subscriber-service* specifies a logical destination for messages that match the specified topic. Messages that match *topic* will be placed on the queue defined by *subscriber-service* and can be read or received through a subsequent call to MQREAD, MQRECEIVE, MQREADALL, or MQRECEIVEALL. This function requires the installation and configuration of an MQSeries based publish and subscribe system, such as MQSeries Integrator or MQSeries Publish/Subscribe. See www.ibm.com/software/MQSeries for further details.

The function returns a value of '1' if successful or a '0' if unsuccessful. Successfully executing this function will cause the publish and subscribe server to forward messages matching the topic to the service point defined by *subscriber-service*.

subscriber-service

A string containing the logical MQSeries subscription point to where messages matching *topic* will be sent. If specified, the *subscriber-service* must refer to a Subscribers Service Point defined in the AMT.XML repository file. Service points definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *subscriber-service* is not specified, then the DB2.DEFAULT.SUBSCRIBER will be used instead. The maximum size of *subscriber-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy to be used in handling the message. If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options to be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used instead. The maximum size of *service-policy* is 48 characters.

topic

A string defining the types of messages to receive. Only messages published with the specified topics will be received by this subscription. Multiple subscriptions may coexist. The maximum size of *topic* is 40 characters. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

Examples:

Example 1: This example registers an interest in messages containing the topic "Weather". The default subscriber-service (DB2.DEFAULT.SUBSCRIBER) is registered as the subscriber and the default service-policy (DB2.DEFAULT.POLICY) specifies the quality of service.

```
VALUES MQSUBSCRIBE('Weather')
```

Example 2: This example demonstrates a subscriber registering interest in messages containing "Stocks". The subscriber registers as "PORTFOLIO-UPDATES" with policy "BASIC-POLICY".

```
VALUES MQSUBSCRIBE('PORTFOLIO-UPDATES', 'BASIC-POLICY', 'Stocks')
```

All examples return a scalar value of '1' if successful.

MQUNSUBSCRIBE

```
▶—MQUNSUBSCRIBE—(—  
                    └──subscriber-service—, └──service-policy—, └──  
▶—topic—)——▶
```

The schema is MQDB2.

The MQUNSUBSCRIBE function is used to unregister an existing message subscription. The *subscriber-service*, *service-policy*, and *topic* are used to identify which subscription is cancelled. This function requires the installation and configuration of an MQSeries based publish and subscribe system, such as MQSeries Integrator or MQSeries Publish/Subscribe. See www.ibm.com/software/MQSeries for further details.

The function returns a value of '1' if successful or a '0' if unsuccessful. The result of successfully executing this function is that the publish and subscribe server will remove the subscription defined by the given parameters. Messages with the specified *topic* will no longer be sent to the logical destination defined by *subscriber-service*.

subscriber-service

If specified, the *subscriber-service* must refer to a Subscribers Service Point defined in the AMT.XML repository file. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *subscriber-service* is not specified, then the DB2.DEFAULT.SUBSCRIBER value is used. The maximum size of *subscriber-service* is 48 characters.

service-policy

If specified, the *service-policy* must refer to a Policy defined in the AMT.XML repository file. A Service Policy defines a set of quality of service options to be applied to this messaging operation. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

topic

A string specifying the subject of messages that are not to be received. The maximum size of *topic* is 40 characters. Multiple topics can be specified in one string (up to 40 characters long). Each topic must be separated by a colon. For example, "t1:t2:the third topic" indicates that the message is associated with all three topics: t1, t2, and "the third topic".

Examples:

Example 1: This example cancels an interest in messages containing the topic "Weather". The default subscriber-service (DB2.DEFAULT.SUBSCRIBER) is registered as the unsubscriber and the default service-policy (DB2.DEFAULT.POLICY) specifies the quality of service.

```
VALUES MQUNSUBSCRIBE('Weather')
```

Example 2: This example demonstrates a subscriber cancelling an interest in messages containing "Stocks". The subscriber is registered as "PORTFOLIO-UPDATES" with policy "BASIC-POLICY".

```
VALUES MQUNSUBSCRIBE('PORTFOLIO-UPDATES', 'BASIC-POLICY', 'Stocks')
```

These examples return a scalar value of '1' if successful and a scalar value of '0' if unsuccessful.

MULTIPLY_ALT

►—MULTIPLY_ALT—(*exact_numeric_expression*, *exact_numeric_expression*)—►

The schema is SYSIBM.

This function was first available in FixPak 2 of Version 7.1.

The MULTIPLY_ALT scalar function returns the product of the two arguments as a decimal value. It is provided as an alternative to the multiplication operator, especially when the sum of the precisions of the arguments exceeds 31.

The arguments can be any built-in exact numeric data type (DECIMAL, BIGINT, INTEGER, or SMALLINT).

The result of the function is a DECIMAL. The precision and scale of the result are determined as follows, using the symbols *p* and *s* to denote the precision and scale of the first argument, and the symbols *p'* and *s'* to denote the precision and scale of the second argument.

- The precision is $\text{MIN}(31, p + p')$
- The scale is:
 - 0 if the scale of both arguments is 0
 - $\text{MIN}(31, s+s')$ if $p+p'$ is less than or equal to 31
 - $\text{MAX}(\text{MIN}(3, s+s'), 31-(p-s+p'-s'))$ if $p+p'$ is greater than 31.

The result can be null if at least one argument can be null or the database is configured with DFT_SQLMATHWARN set to yes; the result is the null value if one of the arguments is null.

- character string (CHAR, VARCHAR)⁵
- datetime (DATE, TIME, TIMESTAMP)
- a user-defined type based on one of the above types

The same column name cannot be specified more than once (SQLSTATE 42734).

The result of the function is VARCHAR. The maximum length is 32672 bytes (SQLSTATE 54006).

Consider the following invocation:

REC2XML (dc, fs, rt, c₁, c₂, ..., c_n)

If the value of **fs** is "COLATTVAL" or "COLATTVAL_XML" the result is the same as the following expression:

```
'<' CONCAT rt CONCAT '>' CONCAT y1 CONCAT y2 CONCAT ... CONCAT yn
CONCAT '</' CONCAT rt CONCAT '>'
```

where y_n is equivalent to:

```
'<column name="' CONCAT xvcn CONCAT vn
```

and v_n is equivalent to:

```
'">' CONCAT rn CONCAT '</column>'
```

if the column is not null, and

```
'" null="true"/>'
```

if the column value is null.

xvc_n is equivalent to a string representation of the column name of c_n, where any characters appearing in Tabelle 29 auf Seite 406 are replaced with the corresponding representation. This ensures that the resulting string is a valid XML attribute or element value token.

r_n is equivalent to a string representation as indicated in Tabelle 28 auf Seite 406.

Result Column Values:

Based on the data type of the column and the actual *format-string* specified, the column values from the table may be transformed before being concatenated into the result string. The following table shows the transformations done on the column values.

5. A character string with a subtype of BIT DATA is not allowed.

Tabelle 28. Column Values String Result

Data type of c_n	r_n
CHAR, VARCHAR	The value is a string. If the <i>format-string</i> does not end in the characters "_XML", then each character in c_n is replaced with the corresponding replacement representation from Tabelle 29, as indicated. The length attribute is: $dc * \text{the length attribute of } c_n$.
SMALLINT, INTEGER, BIGINT, DECIMAL, NUMERIC, REAL, DOUBLE	The value is <code>LTRIM(RTRIM(CHAR(c_n)))</code> . The length attribute is the result length of <code>CHAR(c_n)</code> . The decimal character is always the period character.
DATE	The value is <code>CHAR(c_n, ISO)</code> . The length attribute is the result length of <code>CHAR(c_n, ISO)</code> .
TIME	The value is <code>CHAR(c_n, JIS)</code> . The length attribute is the result length of <code>CHAR(c_n, JIS)</code> .
TIMESTAMP	The value is <code>CHAR(c_n)</code> . The length attribute is the result length of <code>CHAR(c_n)</code> .

Character Replacement:

Depending on the value specified for the *format-string*, certain characters in column names and column values will be replaced to ensure that the column names form valid XML attribute values and the column values form valid XML element values.

Tabelle 29. Character Replacements for XML Attribute Values and Element Values

< is replaced by <
> is replaced by >
" is replaced by "
& is replaced by &
' is replaced by '

Examples:

- Using the DEPARTMENT table, format the department table row, except the DEPTNAME and LOCATION columns, for department 'D01' into a string of valid XML. Since the data does not contain any of the characters which require replacement, the expansion factor will be 1.0 (no expansion). Also note that the MGRNO value is null for this row.

```
SELECT REC2XML (1.0, 'COLATTVAL', '', DEPTNO, MGRNO, ADMRDEPT)
FROM DEPARTMENT
WHERE DEPTNO = 'D01'
```

This example returns the following VARCHAR(117) string:

```
<row>
  <column name="DEPTNO">D01</column>
  <column name="MGRNO" null="true"/>
  <column name="ADMRDEPT">A00</column>
</row>
```

Anmerkung: REC2XML does not insert new line characters in the output. The above example output is formatted for the sake of readability.

- A 5-day university schedule introduces a class with the name '&43<FIE' to a table called CL_SCHED, with a new format for the CLASS_CODE column. Using the REC2XML function, this example formats an XML string with this new class data, except for the class end time.

The length attribute for the REC2XML call (see below) with an expansion factor of 1.0 would be 128 (11 for the '<row>' and '</row>' overhead, 21 for the column names, 75 for the '<column name=', '>', '</column>' and double quotes, 7 for the CLASS_CODE data, 6 for the DAY data, and 8 for the STARTING data). Since the '&' and '<' characters will be replaced, an expansion factor of 1.0 will not be sufficient. The length attribute of the function will need to support an increase from 7 to 14 characters for the new format CLASS_CODE data.

However, since it is known that the DAY value will never be more than 1 digit long, an extra 5 is calculated into the length that will never be used. Therefore, the expansion only needs to handle an increase of 2. Since CLASS_CODE is the only character string column in the argument list, this is the only column value to which the expansion factor applies. To get an increase of 2 for the length, an expansion factor of 9/7 (approximately 1.2857) would be needed. An expansion factor of 1.3 will be used.

```
SELECT REC2XML (1.3, 'COLATTVAL', 'record', CLASS_CODE, DAY, STARTING)
FROM CL_SCHED
WHERE CLASS_CODE = '&43<FIE'
```

Returns the following string:

```
<record>
  <column name="CLASS_CODE">&amp;43&lt;FIE</column>
  <column name="DAY">5</column>
  <column name="STARTING">06:45:00</column>
</record>
```

Anmerkung: REC2XML does not insert new line characters in the output. The above example output is formatted for the sake of readability.

- This example shows characters replaced in a column name.

```
SELECT REC2XML (1.3, 'COLATTVAL', '', Class, "time<noon")
FROM (SELECT Class_code, Starting
FROM Cl_sched
WHERE Starting < '12:00:00')
AS Early (Class, "time<noon")
```

This query returns the following string:

```
<row>
  <column name="CLASS">&amp;43&lt;FIE</column>
  <column name="time&lt;noon">06:45:00</column>
</row>
```

Anmerkung: REC2XML does not insert new line characters in the output. The above example output is formatted for the sake of readability.

- Assume that new rows have been added to the EMP_RESUME table, storing the resumes as strings of valid XML. The COLATTVAL_XML *format-string* is used so character replacement will not be performed. None of the resumes are more than 3500 characters in length. The following query could be used to select the XML version of the resumes from the EMP_RESUME table and format it into an XML document fragment.

```
SELECT REC2XML (1.0,'COLATTVAL_XML', '', EMPNO, RESUME_XML)
FROM (SELECT EMPNO, CAST(RESUME as VARCHAR(3500)) as RESUME_XML
FROM EMP_RESUME
WHERE RESUME_FORMAT = 'XML')
AS EMP_RESUME_XML
```

ROUND

►►—ROUND—(*expression1*, *expression2*)——————►►

The schema is SYSIBM.

This function was first available in FixPak 2 of Version 7.1.

Anmerkung: The SYSFUN version of the ROUND function continues to be available.

The ROUND function returns *expression1* rounded to *expression2* places to the right of the decimal point if *expression2* is positive or to the left of the decimal point if *expression2* is zero or negative.

If *expression1* is positive, a value of 5 is rounded to the next higher positive number. For example, ROUND(3.5,0) = 4. If *expression1* is negative, a value of 5 is rounded to the next lower negative number. For example, ROUND(-3.5,0) = -4.

expression1

An expression that returns a value of any built-in numeric data type.

expression2

An expression that returns a small or large integer. When the value of *expression2* is not negative, it specifies rounding to that number of places to the right of the decimal separator. When the value of *expression2* is negative, it specifies rounding to the absolute value of *expression2* places to the left of the decimal separator.

If *expression2* is not negative, *expression1* is rounded to the absolute value of *expression2* number of places to the right of the decimal point. If the value of *expression2* is greater than the scale of *expression1* then the value is unchanged except that the result value has a precision that is larger by 1. For example, ROUND(748.58,5) = 748.58 where the precision is now 6 and the scale remains 2.

If *expression2* is negative, *expression1* is rounded to the absolute value of *expression2*+1 number of places to the left of the decimal point.

If the absolute value of a negative *expression2* is larger than the number of digits to the left of the decimal point, the result is 0. For example, ROUND(748.58,-4) = 0.

The data type and length attribute of the result are the same as the data type and length attribute of the first argument, except that the precision is increased by one if the *expression1* is DECIMAL or NUMERIC and the precision is less than 31. For example, an argument with a data type of DECIMAL(5,2) results in DECIMAL(6,2). An argument with a data type of DECIMAL(31,2) results in DECIMAL(31,2). The scale is the same as the scale of the first argument.

If either argument can be null or the database is configured with DFT_SQLMA-THWARN set to yes, the result can be null. If either argument is null, the result is the null value.

Examples:: Calculate the number 873.726 rounded to 2, 1, 0, -1, -2, -3, and -4 decimal places respectively.

```
VALUES (ROUND(873.726, 2),
        ROUND(873.726, 1),
        ROUND(873.726, 0),
        ROUND(873.726,-1),
        ROUND(873.726,-2),
        ROUND(873.726,-3),
        ROUND(873.726,-4) )
```

This example returns:

1	2	3	4	5	6	7
873.730	873.700	874.000	870.000	900.000	1000.000	0.000

Calculate a both positive and negative numbers.

```
VALUES (ROUND(3.5, 0),
        ROUND(3.1, 0),
        ROUND(-3.1, 0),
        ROUND(-3.5,0) )
```

This example returns:

1	2	3	4
4.0	3.0	-3.0	-4.0

WEEK_ISO

Change the description of this function to the following:

The schema is SYSFUN.

Returns the week of the year of the argument as an integer value in the range 1-53. The week starts with Monday and always includes 7 days. Week 1 is the first week of the year to contain a Thursday, which is equivalent to the first week containing January 4. It is therefore possible to have up to 3 days at the beginning of a year appear in the last week of the previous year. Conversely, up to 3 days at the end of a year may appear in the first week of the next year.

The argument must be a date, timestamp, or a valid character string representation of a date or timestamp that is neither a CLOB nor a LONG VARCHAR.

The result of the function is INTEGER. The result can be null; if the argument is null, the result is the null value.

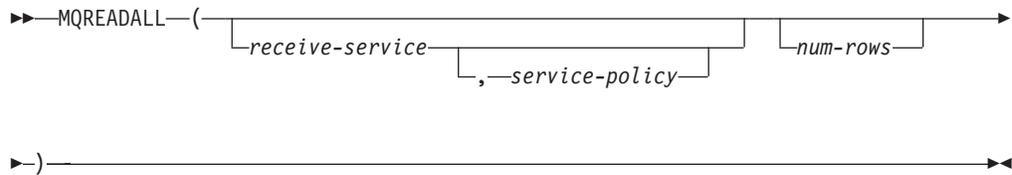
Example:

The following list shows examples of the result of WEEK_ISO and DAYOFWEEK_ISO.

DATE	WEEK_ISO	DAYOFWEEK_ISO
1997-12-28	52	7
1997-12-31	1	3
1998-01-01	1	4
1999-01-01	53	5
1999-01-04	1	1
1999-12-31	52	5
2000-01-01	52	6
2000-01-03	1	1

Table Functions

MQREADALL



The schema is MQDB2.

The MQREADALL function returns a table containing the messages and message metadata from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation does not remove the messages from the queue associated with *receive-service*.

If *num-rows* is specified, then a maximum of *num-rows* messages will be returned. If *num-rows* is not specified, then all available messages will be returned. The table returned contains the following columns:

- MSG - a VARCHAR(4000) column containing the contents of the MQSeries message.
- CORRELID - a VARCHAR(24) column holding a correlation ID used to relate messages.
- TOPIC - a VARCHAR(40) column holding the topic that the message was published with, if available.
- QNAME - a VARCHAR(48) column holding the queue name where the message was received.
- MSGID - a CHAR(24) column holding the assigned MQSeries unique identifier for this message.
- MSGFORMAT - a VARCHAR(8) column holding the format of the message, as defined by MQSeries. Typical strings have a MQSTR format.

receive-service

A string containing the logical MQSeries destination from which the message is read. If specified, the *receive-service* must refer to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in the handling of this message. If specified, the *service-policy* refers to a Policy defined in the AMT XML repository file. A service policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

num-rows

A positive integer containing the maximum number of messages to be returned by the function.

Examples:

Example 1: This example receives all the messages from the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). The messages and all the metadata are returned as a table.

```
SELECT *  
FROM table (MQREADALL()) T
```

Example 2: This example receives all the messages from the head of the queue specified by the service MYSERVICE, using the default policy (DB2.DEFAULT.POLICY). Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQREADALL('MYSERVICE')) T
```

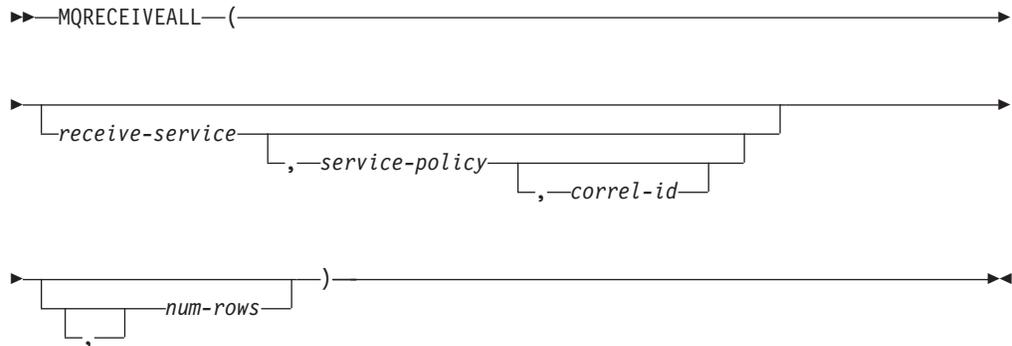
Example 3: This example reads the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). Only messages with a CORRELID of '1234' are returned. All columns are returned.

```
SELECT *  
FROM table (MQREADALL()) T  
WHERE T.CORRELID = '1234'
```

Example 4: This example receives the first 10 messages from the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). All columns are returned.

```
SELECT *  
FROM table (MQREADALL(10)) T
```

MQRECEIVEALL



The schema is MQDB2.

The MQRECEIVEALL function returns a table containing the messages and message metadata from the MQSeries location specified by *receive-service*, using the quality of service policy *service-policy*. Performing this operation removes the messages from the queue associated with *receive-service*.

If a *correl-id* is specified, then only those messages with a matching correlation identifier will be returned. If *correl-id* is not specified, then the message at the head of the queue will be returned.

If *num-rows* is specified, then a maximum of *num-rows* messages will be returned. If *num-rows* is not specified, then all available messages are returned. The table returned contains the following columns:

- MSG - a VARCHAR(4000) column containing the contents of the MQSeries message.
- CORRELID - a VARCHAR(24) column holding a correlation ID used to relate messages.
- TOPIC - a VARCHAR(40) column holding the topic that the message was published with, if available.
- QNAME - a VARCHAR(48) column holding the queue name where the message was received.
- MSGID - a CHAR(24) column holding the assigned MQSeries unique identifier for this message.
- MSGFORMAT - a VARCHAR(8) column holding the format of the message, as defined by MQSeries. Typical strings have a MQSTR format.

receive-service

A string containing the logical MQSeries destination from which the message is received. If specified, the *receive-service* must refer to a service point defined in the AMT.XML repository file. A service point is a logical end-point from which a message is sent or received. Service point definitions include the name of the MQSeries Queue Manager and Queue. See the MQSeries Application Messaging Interface manual for further details. If *receive-service* is not specified, then the DB2.DEFAULT.SERVICE will be used. The maximum size of *receive-service* is 48 characters.

service-policy

A string containing the MQSeries AMI Service Policy used in the handling of this message. If specified, the *service-policy* refers to a Policy defined in the AMT XML repository file. A service policy defines a set of quality of service options that should be applied to this messaging operation. These options include message priority and message persistence. See the MQSeries Application Messaging Interface manual for further details. If *service-policy* is not specified, then the default DB2.DEFAULT.POLICY will be used. The maximum size of *service-policy* is 48 characters.

correl-id

An optional string containing a correlation identifier associated with this message. The *correl-id* is often specified in request and reply scenarios to associate requests with replies. If not specified, no correlation id is specified. The maximum size of *correl-id* is 24 characters.

num-rows

A positive integer containing the maximum number of messages to be returned by the function.

Examples:

Example 1: This example receives all the messages from the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). The messages and all the metadata are returned as a table.

```
SELECT *  
FROM table (MQRECEIVEALL()) T
```

Example 2: This example receives all the messages from the head of the queue specified by the service MYSERVICE, using the default policy (DB2.DEFAULT.POLICY). Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALL('MYSERVICE')) T
```

Example 3: This example receives all of the message from the head of the queue specified by the service "MYSERVICE", using the policy "MYPOLICY". Only messages with a CORRELID of '1234' are returned. Only the MSG and CORRELID columns are returned.

```
SELECT T.MSG, T.CORRELID  
FROM table (MQRECEIVEALL('MYSERVICE', 'MYPOLICY', '1234')) T
```

Example 4: This example receives the first 10 messages from the head of the queue specified by the default service (DB2.DEFAULT.SERVICE), using the default policy (DB2.DEFAULT.POLICY). All columns are returned.

```
SELECT *  
FROM table (MQRECEIVEALL(10)) T
```

Procedures

GET_ROUTINE_SAR

►► GET_ROUTINE_SAR (—*sarblob*—, —*type*—, —*routine_name_string*—) ◀◀

The schema is SYSFUN.

This procedure was first available in FixPak 3 of Version 7.1.

The GET_ROUTINE_SAR procedure retrieves the necessary information to install the same routine in another database server running the same level on the same operating system. The information is retrieved into a single BLOB string representing an SQL archive file. The invoker of the GET_ROUTINE_SAR procedure must have DBADM authority.

sarblob

An output argument of type BLOB(3M) that contains the routine SAR file contents.

type

An input argument of type CHAR(2) that specifies whether the type of routine, using one of the following values:

- **P** for a procedure.
- **SP** for the specific name of a procedure.

routine_name_string

An input argument of type VARCHAR(257) that specifies a qualified name of the routine. If no schema name is specified, the default is the CURRENT SCHEMA when the routine is processed.

Anmerkung: The *routine_name_string* cannot include the double quote character (").

The qualified name of the routine is used to determine which routine to retrieve. The routine that is found must be an SQL routine or an error is raised (SQLSTATE 428F7). When not using a specific name, this may result in more than one routine and an error is raised (SQLSTATE 42725). If this occurs, the specific name of the routine must be used to get the routine.

The SAR file must include a bind file which may not be available at the server. If the bind file cannot be found and stored in the SAR file, an error is raised (SQLSTATE 55045).

PUT_ROUTINE_SAR

►► PUT_ROUTINE_SAR ((*sarblob* [, *new_owner* , *use_register_flag*])) ►►

The schema is SYSFUN.

This procedure was first available in FixPak 3 of Version 7.1.

The PUT_ROUTINE_SAR procedure passes the necessary file to create an SQL routine at the server and then defines the routine. The invoker of the PUT_ROUTINE_SAR procedure must have DBADM authority.

sarblob

An input argument of type BLOB(3M) that contains the routine SAR file contents.

new_owner

An input argument of type VARCHAR(128) that contains an authorization-name used for authorization checking of the routine. The *new-owner* must have the necessary privileges for the routine to be defined. If *new-owner* is not specified, the authorization-name of the original routine definer is used.

use_register_flag

An input argument of type INTEGER that indicates whether or not the CURRENT SCHEMA and CURRENT PATH special registers are used to define the routine. If the special registers are not used, the settings for the default schema and SQL path are the settings used when the routine was originally defined. Possible values for *use-register-flag*:

- 0 Do not use the special registers of the current environment
- 1 Use the CURRENT SCHEMA and CURRENT PATH special registers.

If the value is 1, CURRENT SCHEMA is used for unqualified object names in the routine definition (including the name of the routine) and CURRENT PATH is used to resolve unqualified routines and data types in the routine definition. If the *use-registers-flag* is not specified, the behavior is the same as if a value of 0 was specified.

The identification information contained in *sarblob* is checked to confirm that the inputs are appropriate for the environment, otherwise an error is raised (SQLSTATE 55046). The PUT_ROUTINE_SAR procedure then uses the contents of the *sarblob* to define the routine at the server.

The contents of the *sarblob* argument are extracted into the separate files that make up the SQL archive file. The shared library and bind files are written to files in a temporary directory. The environment is set so that the routine definition statement processing is aware that compiling and linking are not required, and that the location of the shared library and bind files is available. The contents of the DDL file are then used to dynamically execute the routine definition statement.

Anmerkung: No more than one procedure can be concurrently installed under a given schema.

Processing of this statement may result in the same errors as executing the routine definition statement using other interfaces. During routine definition processing,

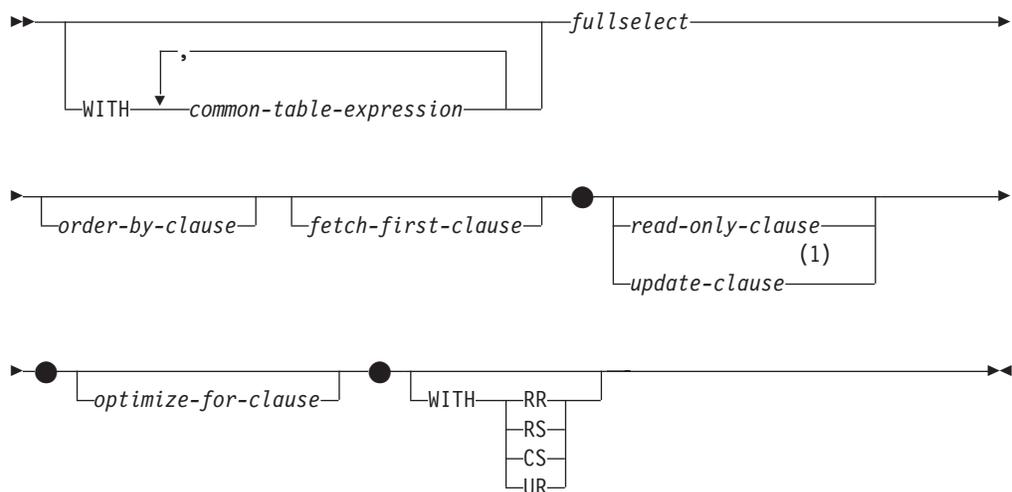
the presence of the shared library and bind files is noted and the precompile, compile and link steps are skipped. The bind file is used during bind processing and the contents of both files are copied to the usual directory for an SQL routine.

Anmerkung: If a GET ROUTINE or a PUT ROUTINE operation (or their corresponding procedure) fails to execute successfully, it will always return an error (SQLSTATE 38000), along with diagnostic text providing information about the cause of the failure. For example, if the procedure name provided to GET ROUTINE does not identify an SQL procedure, diagnostic "100, 02000" text will be returned, where "100" and "02000" are the SQLCODE and SQLSTATE, respectively, that identify the cause of the problem. The SQLCODE and SQLSTATE in this example indicate that the row specified for the given procedure name was not found in the catalog tables.

Chapter 5. Queries

select-statement/syntax diagram

The syntax diagram changes to:



Anmerkungen:

- 1 The update-clause and the order-by-clause cannot both be specified in the same select-statement.

Add the following paragraph to the description below the syntax diagram:

The optional WITH clause specifies the isolation level at which the select statement is executed.

- RR - Repeatable Read
- RS - Read Stability
- CS - Cursor Stability
- UR - Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

select-statement/fetch-first-clause

The last paragraph in the description of the fetch-first-clause:

Specification of the fetch-first-clause in a select-statement makes the cursor not deletable (read-only). This clause cannot be specified with the FOR UPDATE clause.

is incorrect and should be removed.

Chapter 6. SQL Statements

Update of the Partitioning Key Now Supported

Update the partitioning key is now supported. The following text from various statements in Chapter 6 should be **deleted** only if the DB2_UPDATE_PART_KEY=ON:

Anmerkung: If DB2_UPDATE_PART_KEY=OFF, then the restrictions still apply.

Statement: ALTER TABLE

Rules

- A partitioning key column of a table cannot be updated (SQLSTATE 42997).
- A nullable column of a partitioning key cannot be included as a foreign key column when the relationship is defined with ON DELETE SET NULL (SQLSTATE 42997).

Statement: CREATE TABLE

Rules

- A partitioning key column of a table cannot be updated (SQLSTATE 42997).
- A nullable column of a partitioning key cannot be included as a foreign key column when the relationship is defined with ON DELETE SET NULL (SQLSTATE 42997).

Statement: DECLARE GLOBAL TEMPORARY TABLE PARTITIONING KEY (column-name,...)

Anmerkung: The partitioning key columns cannot be updated (SQLSTATE 42997).

Statement: UPDATE

Footnotes

- 108 A column of a partitioning key is not updatable (SQLSTATE 42997). The row of data must be deleted and inserted to change columns in a partitioning key.

Larger Index Keys for Unicode Databases

ALTER TABLE

The length of variable length columns that are part of any index, including primary and unique keys, defined when the registry variable `DB2_INDEX_2BYTEVARLEN` was on, can be altered to a length greater than 255 bytes. The fact that a variable length column is involved in a foreign key will no longer prevent the length of that column from being altered to larger than 255 bytes, regardless of the registry variable setting. However, data with length greater than 255 cannot be inserted into the table unless the column in the corresponding primary key has length greater than 255 bytes, which is only possible if the primary key was created with the registry variable ON.

CREATE INDEX

Indexes can be defined on variable length columns whose length is greater than 255 bytes if the registry variable `DB2_INDEX_2BYTEVARLEN` is ON.

CREATE TABLE

Primary and unique keys with variable keyparts can have a size greater than 255 if the registry variable `DB2_INDEX_2BYTEVARLEN` is ON. Foreign keys can be defined on variable length columns whose length is greater than 255 bytes.

ALTER SEQUENCE

ALTER SEQUENCE

The `ALTER SEQUENCE` statement modifies the attributes of a sequence by:

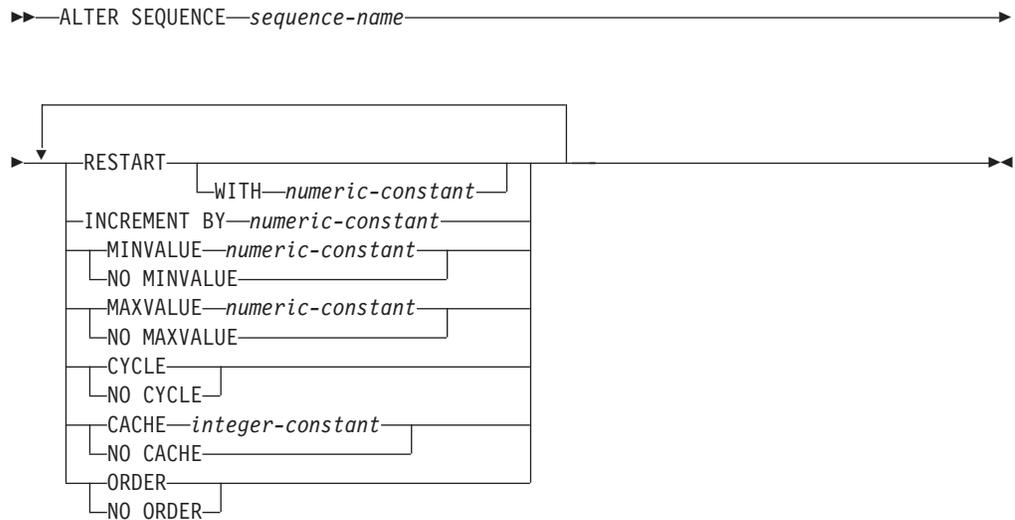
- Restarting the sequence
- Changing the increment between future sequence values
- Setting new minimum or maximum values
- Changing the number of cached sequence numbers
- Changing whether the sequence can cycle or not
- Changing whether sequence numbers must be generated in order of request

Aufruf: This statement can be embedded in an application program or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared. However, if the bind option `DYNAMICRULES BIND` applies, the statement cannot be dynamically prepared (SQLSTATE 42509).

Berechtigung: The privileges held by the authorization ID of the statement must include at least one of the following:

- Definer of the sequence
- The `ALTERIN` privilege for the schema implicitly or explicitly specified
- `SYSADM` or `DBADM` authority

Syntax:



Beschreibung:

sequence-name

Identifies the particular sequence. The combination of name, and the implicit or explicit schema name must identify an existing sequence at the current server. If no sequence by this name exists in the explicitly or implicitly specified schema, an error (SQLSTATE 42704) is issued.

RESTART

Restarts the sequence. If *numeric-constant* is not specified, the sequence is restarted at the value specified implicitly or explicitly as the starting value on the CREATE SEQUENCE statement that originally created the sequence.

WITH *numeric-constant*

Restarts the sequence with the specified value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820) as long as there are no non-zero digits to the right of the decimal point (SQLSTATE 42894).

INCREMENT BY

Specifies the interval between consecutive values of the sequence. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), and does not exceed the value of a large integer constant (SQLSTATE 42815), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA).

If this value is negative, then the sequence of values descends. If this value is positive, then the sequence of values ascends. If this value is 0 or greater than the range defined by MINVALUE and MAXVALUE, only one value will be generated, but the sequence is treated as an ascending sequence otherwise.

MINVALUE or NO MINVALUE

Specifies the minimum value at which a descending sequence either cycles or stops generating values, or an ascending sequence cycles to after reaching the maximum value.

MINVALUE *numeric-constant*

Specifies the numeric constant that is the minimum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be less than or equal to the maximum value (SQLSTATE 42815).

NO MINVALUE

For an ascending sequence, the value is the START WITH value, or 1 if START WITH is not specified. For a descending sequence, the value is the minimum value of the data type associated with the sequence. This is the default.

MAXVALUE or NO MAXVALUE

Specifies the maximum value at which an ascending sequence either cycles or stops generating values, or a descending sequence cycles to after reaching the minimum value.

MAXVALUE *numeric-constant*

Specifies the numeric constant that is the maximum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 428FA), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be greater than or equal to the minimum value (SQLSTATE 42815).

NO MAXVALUE

For an ascending sequence, the value is the maximum value of the data type associated with the sequence. For a descending sequence, the value is the START WITH value, or -1 if START WITH is not specified. This is the default.

CYCLE or NOCYCLE

Specifies whether the sequence should continue to generate values after reaching either its maximum or minimum value. The boundary of the sequence can be reached either with the next value landing exactly on the boundary condition, or by overshooting it in which case the next value would be determined from wrapping around to the START WITH value if cycles were permitted.

CYCLE

Specifies that values continue to be generated for this sequence after the maximum or minimum value has been reached. If this option is used, after an ascending sequence reaches its maximum value, it generates its minimum value; or after a descending sequence reaches its minimum value, it generates its maximum value. The maximum and minimum values for the sequence determine the range that is used for cycling.

When CYCLE is in effect, then duplicate values can be generated for the sequence.

NO CYCLE

Specifies that values will not be generated for the sequence once the maximum or minimum value for the sequence has been reached. This is the default.

CACHE or NO CACHE

Specifies whether to keep some preallocated values in memory for faster access. This is a performance and tuning option.

CACHE *integer-constant*

Specifies the maximum number of sequence values that are preallocated and kept in memory. Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence.

In the event of a system failure, all cached sequence values that have not been used in committed statements are lost (that is, they will never be used). The value specified for the CACHE option is the maximum number of sequence values that could be lost in case of system failure.

The minimum value is 2 (SQLSTATE 42815). The default value is CACHE 20.

NO CACHE

Specifies that values of the sequence are not to be preallocated. It ensures that there is not a loss of values in the case of a system failure, shutdown or database deactivation. When this option is specified, the values of the sequence are not stored in the cache. In this case, every request for a new value for the sequence results in synchronous I/O to the log.

NO ORDER or ORDER

Specifies whether the sequence numbers must be generated in order of request.

ORDER

Specifies that the sequence numbers are generated in order of request.

NO ORDER

Specifies that the sequence numbers do not need to be generated in order of request. This is the default.

After restarting a sequence or changing to CYCLE, it is possible for sequence numbers to be duplicate values of ones generated by the sequence previously.

Anmerkungen:

- Only future sequence numbers are affected by the ALTER SEQUENCE statement.
- The data type of a sequence cannot be changed. Instead, drop and recreate the sequence specifying the desired data type for the new sequence.
- All cached values are lost when a sequence is altered.

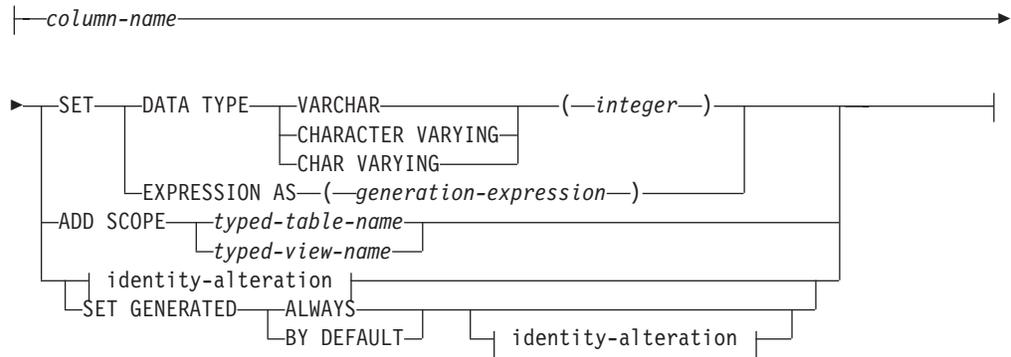
Beispiele: *Example 1:* A possible reason for specifying RESTART without a numeric value would be to reset the sequence to the START WITH value. In this example, the goal is to generate the numbers from 1 up to the number of rows in the table and then inserting the numbers into a column added to the table using temporary tables. Another use would be to get results back where all the resulting rows are numbered:

```
ALTER SEQUENCE org_seq
RESTART
SELECT NEXTVAL for org_seq, org.*
FROM org
```

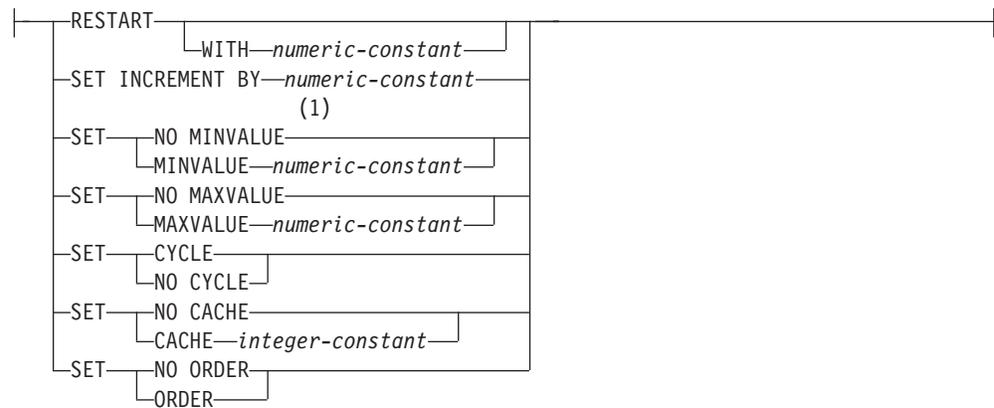
ALTER TABLE

Changes to syntax fragments:

column-alteration:



identity-alteration:



Anmerkungen:

- 1 These parameters can be specified without spaces: NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, and NOORDER. These single word versions are all acceptable alternatives to the two word versions.

Add the following parameters:

SET GENERATED

Specifies whether values are to be generated for the column always or only when a default value is needed.

ALWAYS

A value will always be generated for the column when a row is inserted or updated in the table. The column must already be defined as a generated column (SQLSTATE 42837).

BY DEFAULT

The value will be generated for the column when a row is inserted into the table, unless a value is specified. The column must already be defined as a generated column (SQLSTATE 42837).

RESTART or RESTART WITH *numeric-constant*

Resets the state of the sequence associated with the identity column. If **WITH *numeric-constant*** is not specified, then the sequence for the identity column is restarted at the value that was specified, either implicitly or explicitly, as the starting value when the identity column was originally created. The *numeric-constant* is an exact numeric constant that can be any positive or negative value that could be assigned to this column (SQLSTATE 42820) as long as there are no non-zero digits to the right of the decimal point (SQLSTATE 42894). The column must already be defined with the **IDENTITY** attribute (SQLSTATE 42837). The *numeric-constant* will be used as the next value for the column.

SET INCREMENT BY *numeric-constant*

Specifies the interval between consecutive values of the identity column. The column must already be defined with the **IDENTITY** attribute (SQLSTATE 42837). This value is any positive or negative value that could be assigned to this column (SQLSTATE 42820), and does not exceed the value of a large integer constant (SQLSTATE 42815), as long as there are no non-zero digits to the right of the decimal point (SQLSTATE 42894).

If this value is negative, then the sequence of values for this identity column descends. If this value is positive, then the sequence of values for this identity column ascends. If this value is 0, or is greater than the range defined by **MINVALUE** and **MAXVALUE**, then DB2 will only generate one value, but the sequence is treated as an ascending sequence otherwise.

SET MINVALUE *numeric-constant* or NO MINVALUE

Specifies the minimum value at which a descending identity column either cycles or stops generating values, or the value to which an ascending identity column cycles to after reaching the maximum value. The column must already be defined with the **IDENTITY** attribute (SQLSTATE 42837).

MINVALUE *numeric-constant*

Specifies the minimum numeric constant value. This value can be any positive or negative value that could be assigned to this column (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 42894), but the value must be less than the maximum value (SQLSTATE 42815).

NO MINVALUE

For an ascending sequence, the value is the **START WITH** value, or 1 if **START WITH** is not specified. For a descending sequence, the value is the minimum value of the data type of the column.

SET MAXVALUE *numeric-constant* or NO MAXVALUE

Specifies the maximum value at which an ascending identity column either cycles or stops generating values, or the value to which a descending identity column cycles to after reaching the minimum value. The column must already be defined with the **IDENTITY** attribute (SQLSTATE 42837).

MAXVALUE *numeric-constant*

Specifies the numeric constant that is the maximum value. This value can be any positive or negative value that could be assigned to this column (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 42894), but the value must be greater than the minimum value (SQLSTATE 42815).

NO MAXVALUE

For an ascending sequence, the value is the maximum value of the

data type of the column. For a descending sequence, the value is the START WITH value, or -1 if START WITH is not specified.

SET CYCLE or NO CYCLE

Specifies whether this identity column should continue to generate values after generating either the maximum or minimum value. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

CYCLE

Specifies that values continue to be generated for this column after the maximum or minimum value has been reached. If this option is used, then after an ascending identity column reaches the maximum value, it generates its minimum value; or after a descending sequence reaches the minimum value, it generates its maximum value. The maximum and minimum values for the identity column determine the range that is used for cycling.

When CYCLE is in effect, then duplicate values can be generated for an identity column. Although not required, if unique values are desired, a single-column unique index defined using the identity column will ensure uniqueness. If a unique index exists on such an identity column and a non-unique value is generated, then an error occurs (SQLSTATE 23505).

NO CYCLE

Specifies that values will not be generated for the identity column once the maximum or minimum value has been reached.

SET CACHE *integer-constant* or NO CACHE

Specifies whether to keep some preallocated values in memory for faster access. This is a performance and tuning option. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

CACHE *integer-constant*

Specifies how many values of the identity sequence are preallocated and kept in memory. When values are generated for the identity column, preallocating and storing values in the cache reduces synchronous I/O to the log.

If a new value is needed for the identity column and there are no unused values available in the cache, then the allocation of the value requires waiting for I/O to the log. However, when a new value is needed for the identity column and there is an unused value in the cache, the allocation of that identity value can happen more quickly by avoiding the I/O to the log.

When a database manager is stopped (database deactivation, system failure, or shutdown, for example), all cached sequence values that have not been used in committed statements are lost (that is, they will never be used). The value specified for the CACHE option is the maximum number of values for the identity column that could be lost in case of system failure.

The minimum value is 2 (SQLSTATE 42615).

NO CACHE

Specifies that values for the identity column are not to be preallocated.

When this option is specified, the values of the identity column are not stored in the cache. In this case, every request for a new identity value results in synchronous I/O to the log.

SET ORDER or NO ORDER

Specifies whether the identity column values must be generated in order of request. The column must already be defined with the IDENTITY attribute (SQLSTATE 42837).

ORDER

Specifies that the identity column values are generated in order of request.

NO ORDER

Specifies that the identity column values do not need to be generated in order of request.

Compound SQL (Embedded)

A prepared COMMIT statement is not allowed in an ATOMIC compound SQL statement.

Compound Statement (Dynamic)**Compound Statement (Dynamic)**

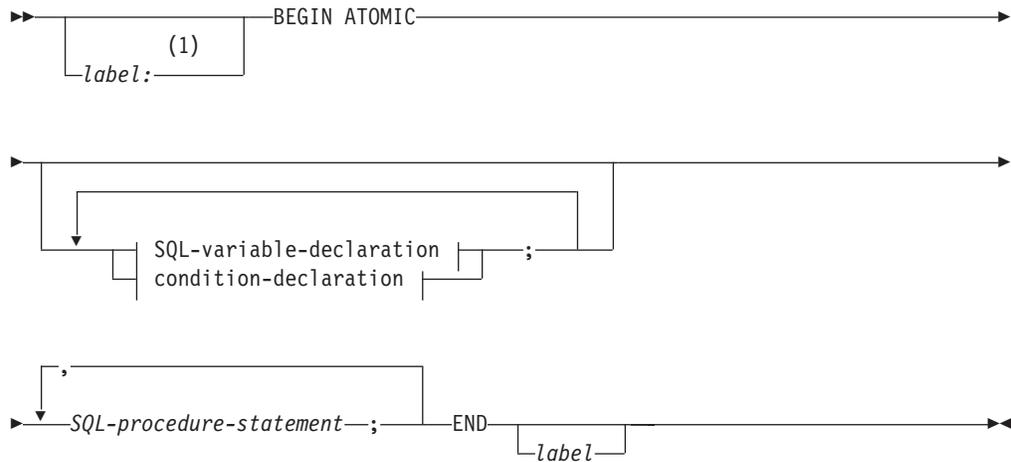
A compound statement groups other statements together into an executable block. You can declare SQL variables within a dynamically prepared atomic compound statement.

Aufruf: This statement can be embedded in a trigger, SQL Function, or SQL Method, or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared.

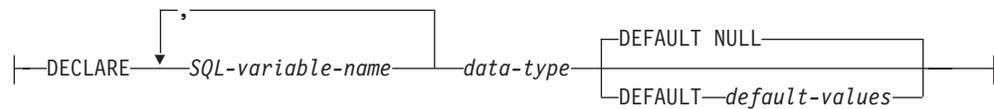
Berechtigung: No privileges are required to invoke a dynamic compound statement. However, the authorization ID of the compound statement must hold the necessary privileges to invoke the SQL statements embedded in the compound statement.

Syntax:

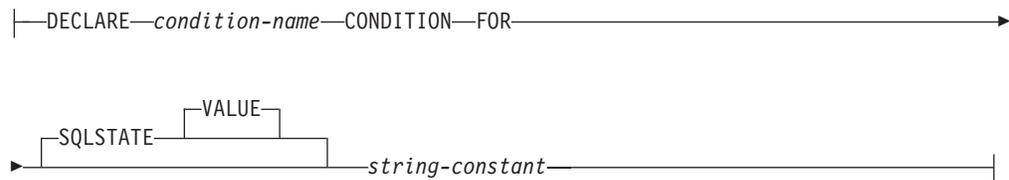
dynamic-compound-statement



SQL-variable-declaration:



condition-declaration:



Anmerkungen:

- 1 A label can only be specified when the statement is in a function, method, or trigger definition.

Beschreibung:

label

Defines the label for the code block. If the beginning label is specified, it can be used to qualify SQL variables declared in the dynamic compound statement and can also be specified on a LEAVE statement. If the ending label is specified, it must be the same as the beginning label.

ATOMIC

ATOMIC indicates that, if an error occurs in the compound statement, all SQL statements in the compound statement will be rolled back and any remaining SQL statements in the compound statement are not processed.

SQL-procedure-statement

The following list of *SQL-control-statements* can be used within the dynamic compound statement:

- FOR Statement
- GET DIAGNOSTICS Statement
- IF Statement
- ITERATE Statement
- LEAVE Statement
- SIGNAL Statement
- WHILE Statement

The SQL statements that can be issued are:

- fullselect⁶
- Searched UPDATE
- Searched DELETE
- INSERT
- SET variable statement

SQL-variable-declaration

Declares a variable that is local to the dynamic compound statement.

SQL-variable-name

Defines the name of a local variable. DB2 converts all SQL variable names to uppercase. The name cannot:

- Be the same as another SQL variable within the same compound statement.
- Be the same as a parameter name.
- Be the same as column names.

If an SQL statement contains an identifier with the same name as an SQL variable and a column reference, DB2 interprets the identifier as a column.

data-type

Specifies the data type of the variable.

DEFAULT *default-values* **or NULL**

Defines the default for the SQL variable. The variable is initialized when the dynamic compound statement is called. If a default value is not specified, the variable is initialized to NULL.

condition-declaration

Declares a condition name and corresponding SQLSTATE value.

condition-name

Specifies the name of the condition. The condition name must be unique within the procedure body and can be referenced only within the compound statement in which it is declared.

FOR SQLSTATE *string-constant*

Specifies the SQLSTATE associated with the condition. The *string-constant* must be specified as five characters enclosed in single quotes, and cannot be '00000'.

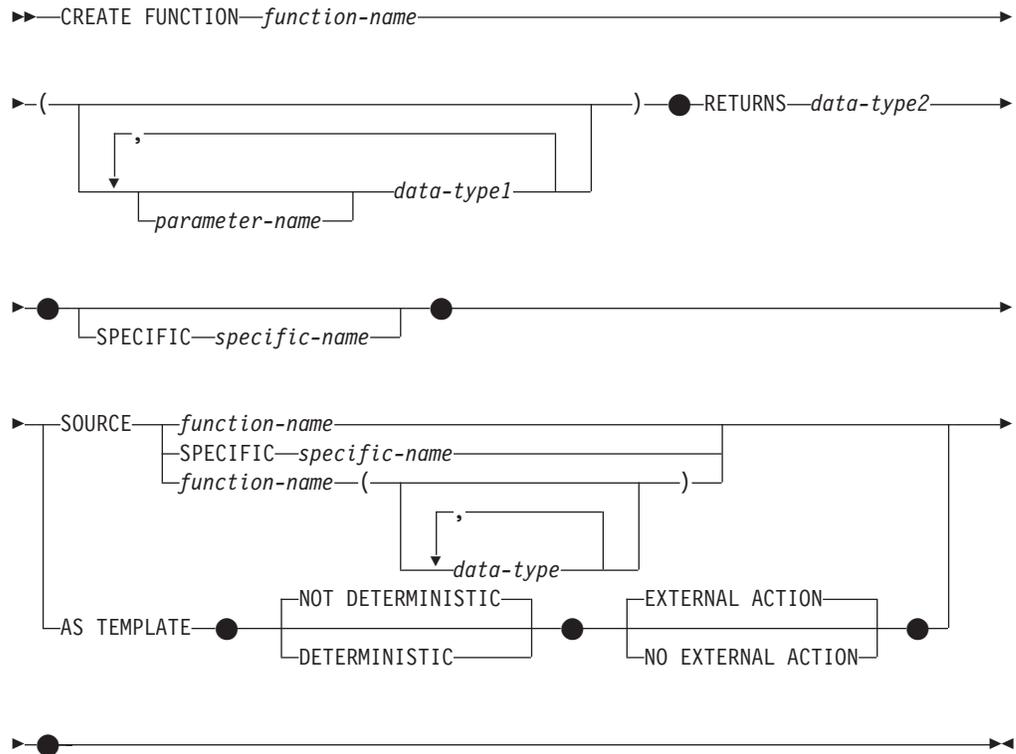
6. A common-table-expression may precede the fullselect

Anmerkungen:

- Dynamic compound statements are compiled by DB2 as one single statement. This statement is effective for short scripts involving little control flow logic but significant data flow. For larger constructs with nested complex control flow, a better choice is to use SQL procedures for details on using SQL procedures.

CREATE FUNCTION (Source or Template)

The syntax diagram changes to the following



Add the following to the "Description" section:

DETERMINISTIC or NOT DETERMINISTIC

This optional clause specifies whether the function always returns the same results for given argument values (DETERMINISTIC) or whether the function depends on some state values that affect the results (NOT DETERMINISTIC). That is, a DETERMINISTIC function must always return the same table from successive invocations with identical inputs. Optimizations taking advantage of the fact that identical inputs always produce the same results are prevented by specifying NOT DETERMINISTIC.

NOT DETERMINISTIC must be explicitly or implicitly specified if the body of the function accesses a special register or calls another non-deterministic function (SQLSTATE 428C2).

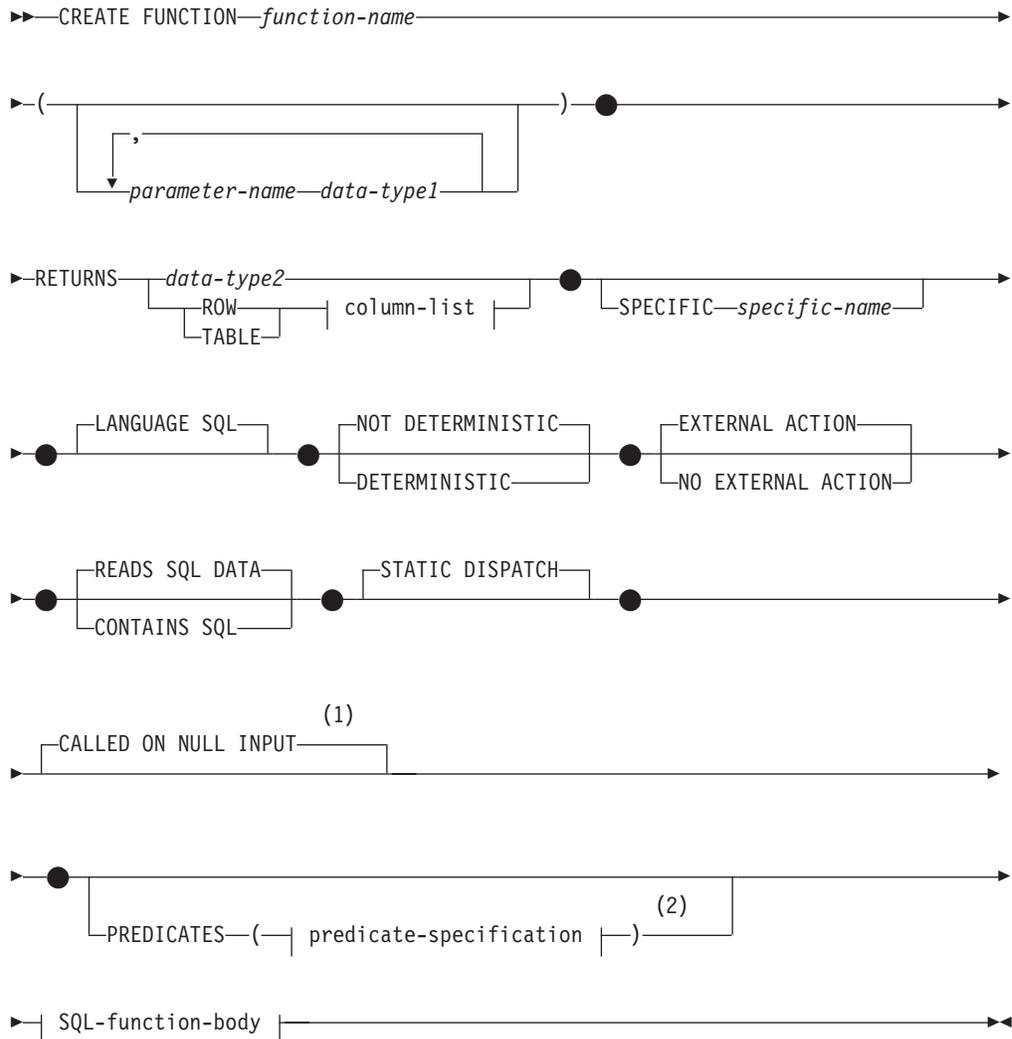
NO EXTERNAL ACTION or EXTERNAL ACTION

This optional clause specifies whether or not the function takes some action that changes the state of an object not managed by the database manager. By specifying NO EXTERNAL ACTION, the system can use certain optimizations that assume functions have no external impacts.

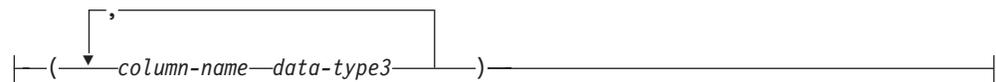
EXTERNAL ACTION must be explicitly or implicitly specified if the body of the function calls another function that has an external action (SQLSTATE 428C2).

CREATE FUNCTION (SQL Scalar, Table or Row)

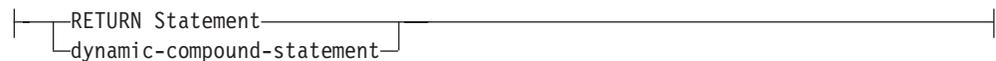
The syntax diagram changes to:



column-list:



SQL-function-body:



Anmerkungen:

- 1 NULL CALL may be specified in place of CALLED ON NULL INPUT

2 Valid only if RETURNS specifies a scalar result (data-type2)

Change the following parameters:

LANGUAGE SQL

Specifies that the function is written using SQL.

This parameter section replaces the "RETURN expression, NULL, WITH common-table-expression, fullselect" parameter section.

SQL-function-body

Specifies the body of the function. Parameter names can be referenced in the SQL-function-body. Parameter names may be qualified with the function name to avoid ambiguous references.

If the SQL-function-body is a dynamic compound statement, it must contain at least one RETURN statement and a RETURN statement must be executed when the function is called (SQLSTATE 42632). If the function is a table or row function, then it can contain only one RETURN statement which must be the last statement in the dynamic compound (SQLSTATE 429BD).

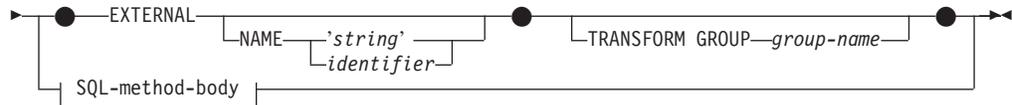
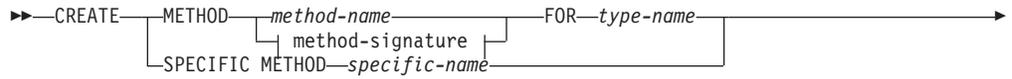
For additional details, see „Compound Statement (Dynamic)“ auf Seite 425 and „RETURN“ auf Seite 456.

CREATE METHOD

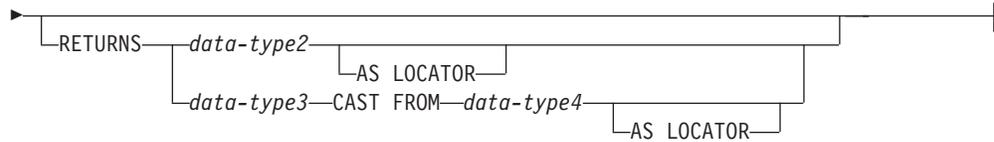
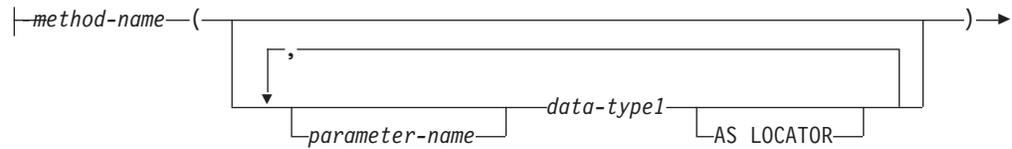
The syntax diagram changes to:

CREATE METHOD

Syntax:



method-signature:



SQL-method-body:



The following parameters replace the "RETURN scalar-expression or NULL" section:

SQL-method-body

The SQL-method-body defines the how the method is implemented if the method specification in CREATE TYPE is LANGUAGE SQL.

The SQL-method-body must comply with the following parts of the method specification:

- DETERMINISTIC or NOT DETERMINISTIC (SQLSTATE 428C2)
- EXTERNAL ACTION or NO EXTERNAL ACTION (SQLSTATE 428C2)
- CONTAINS SQL or READS SQL DATA (SQLSTATE 42985)

Parameter names can be referenced in the SQL-method-body. The subject of the method is passed to the method implementation as an implicit first parameter named SELF.

For additional details, see „Compound Statement (Dynamic)“ auf Seite 425 and „RETURN“ auf Seite 456.

CREATE SEQUENCE

CREATE SEQUENCE

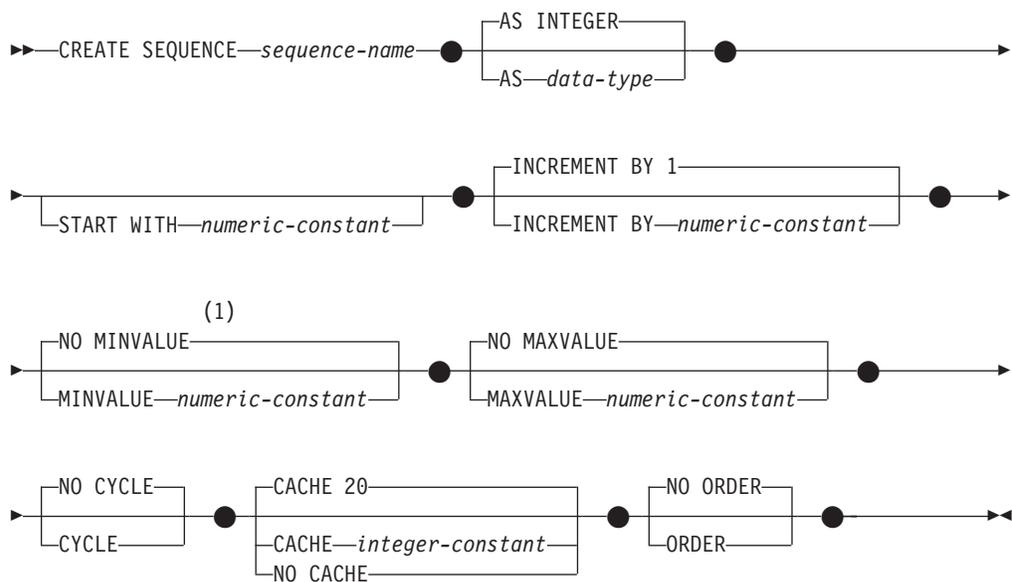
The CREATE SEQUENCE statement creates a sequence at the application server.

Aufruf: This statement can be embedded in an application program or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared. However, if the bind option DYNAMICRULES BIND applies, the statement cannot be dynamically prepared (SQLSTATE 42509).

Berechtigung: The privileges held by the authorization ID of the statement must include at least one of the following:

- CREATEIN privilege for the implicitly or explicitly specified schema
- SYSADM or DBADM authority

Syntax:



Anmerkungen:

- 1 These parameters can be specified without spaces: NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, and NOORDER. These single word versions are all acceptable alternatives to the two word versions.

Beschreibung:

sequence-name

Names the sequence. The combination of name, and the implicit or explicit schema name must not identify an existing sequence at the current server (SQLSTATE 42710).

The unqualified form of sequence-name is an SQL identifier. The qualified form is a qualifier followed by a period and an SQL identifier. The qualifier is a schema name.

If the sequence name is explicitly qualified with a schema name, the schema name cannot begin with 'SYS' or an error (SQLSTATE 42939) is raised.

AS *data-type*

Specifies the data type to be used for the sequence value. The data type can be any exact numeric type (SMALLINT, INTEGER, BIGINT or DECIMAL) with a scale of zero or a user-defined distinct type for which the source type is an exact numeric type with a scale of zero (SQLSTATE 42815). The default is INTEGER.

START WITH *numeric-constant*

Specifies the first value for the sequence. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA). The default is MINVALUE for ascending sequences and MAXVALUE for descending sequences.

This value is not necessarily the value that a sequence would cycle to after reaching the maximum or minimum value of the sequence. The START WITH clause can be used to start a sequence outside the range that is used for cycles. The range used for cycles is defined by MINVALUE and MAXVALUE.

INCREMENT BY *numeric-constant*

Specifies the interval between consecutive values of the sequence. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), and does not exceed the value of a large integer constant (SQLSTATE 42815), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA).

If this value is negative, then the sequence of values descends. If this value is positive, then the sequence of values ascends. If this value is 0 or greater than the range defined by MINVALUE and MAXVALUE, only one value will be generated, but the sequence is treated as an ascending sequence otherwise. The default is 1.

MINVALUE or NO MINVALUE

Specifies the minimum value at which a descending sequence either cycles or stops generating values, or an ascending sequence cycles to after reaching the maximum value.

MINVALUE *numeric-constant*

Specifies the numeric constant that is the minimum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 42820), without non-zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be less than or equal to the maximum value (SQLSTATE 42815).

NO MINVALUE

For an ascending sequence, the value is the START WITH value, or 1 if START WITH is not specified. For a descending sequence, the value is the minimum value of the data type associated with the sequence. This is the default.

MAXVALUE or NO MAXVALUE

Specifies the maximum value at which an ascending sequence either cycles or stops generating values, or a descending sequence cycles to after reaching the minimum value.

MAXVALUE *numeric-constant*

Specifies the numeric constant that is the maximum value. This value can be any positive or negative value that could be assigned to a column of the data type associated with the sequence (SQLSTATE 428FA), without non-

zero digits existing to the right of the decimal point (SQLSTATE 428FA), but the value must be greater than or equal to the minimum value (SQLSTATE 42815).

NO MAXVALUE

For an ascending sequence, the value is the maximum value of the data type associated with the sequence. For a descending sequence, the value is the START WITH value, or -1 if START WITH is not specified. This is the default.

CYCLE or NO CYCLE

Specifies whether the sequence should continue to generate values after reaching either its maximum or minimum value. The boundary of the sequence can be reached either with the next value landing exactly on the boundary condition, or by overshooting it.

CYCLE

Specifies that values continue to be generated for this sequence after the maximum or minimum value has been reached. If this option is used, after an ascending sequence reaches its maximum value it generates its minimum value; after a descending sequence reaches its minimum value it generates its maximum value. The maximum and minimum values for the sequence determine the range that is used for cycling.

When CYCLE is in effect, then duplicate values can be generated for the sequence.

NO CYCLE

Specifies that values will not be generated for the sequence once the maximum or minimum value for the sequence has been reached. This is the default.

CACHE or NO CACHE

Specifies whether to keep some preallocated values in memory for faster access. This is a performance and tuning option.

CACHE *integer-constant*

Specifies the maximum number of sequence values that are preallocated and kept in memory. Preallocating and storing values in the cache reduces synchronous I/O to the log when values are generated for the sequence.

In the event of a system failure, all cached sequence values that have not been used in committed statements are lost (that is, they will never be used). The value specified for the CACHE option is the maximum number of sequence values that could be lost in case of system failure.

The minimum value is 2 (SQLSTATE 42815). The default value is CACHE 20.

NO CACHE

Specifies that values of the sequence are not to be preallocated. It ensures that there is not a loss of values in the case of a system failure, shutdown or database deactivation. When this option is specified, the values of the sequence are not stored in the cache. In this case, every request for a new value for the sequence results in synchronous I/O to the log.

NO ORDER or ORDER

Specifies whether the sequence numbers must be generated in order of request.

ORDER

Specifies that the sequence numbers are generated in order of request.

NO ORDER

Specifies that the sequence numbers do not need to be generated in order of request. This is the default.

Anmerkungen:

- It is possible to define a constant sequence, that is, one that would always return a constant value. This could be done by specifying the same value for MINVALUE or MAXVALUE, or by specifying an INCREMENT value of zero. In either case, in order to allow for NEXTVAL to generate the same value more than once, CYCLE must be specified. A constant sequence can be used as a numeric global variable. ALTER SEQUENCE can be used to adjust the values that will be generated for a constant sequence.
- A sequence can be cycled manually, by using the ALTER SEQUENCE statement. If NO CYCLE is implicitly or explicitly specified, the sequence can be restarted or extended using the ALTER SEQUENCE statement to cause values to continue to be generated once the maximum or minimum value for the sequence has been reached.
- Caching sequence numbers implies that a range of sequence numbers can be kept in memory for fast access. When an application accesses a sequence that can allocate the next sequence number from the cache, the sequence number allocation can happen quickly. However, if an application accesses a sequence that cannot allocate the next sequence number from the cache, the sequence number allocation may require having to wait for I/O operations to persistent storage. The choice of the value for CACHE should be done keeping in mind the performance and application requirements tradeoffs.
- The owner has the ALTER and USAGE privileges on the new sequence. Only the USAGE privilege can be granted by the owner and only to PUBLIC.
- The following syntax is also supported: NOMINVALUE, NOMAXVALUE, NOCYCLE, NOCACHE, and NOORDER.

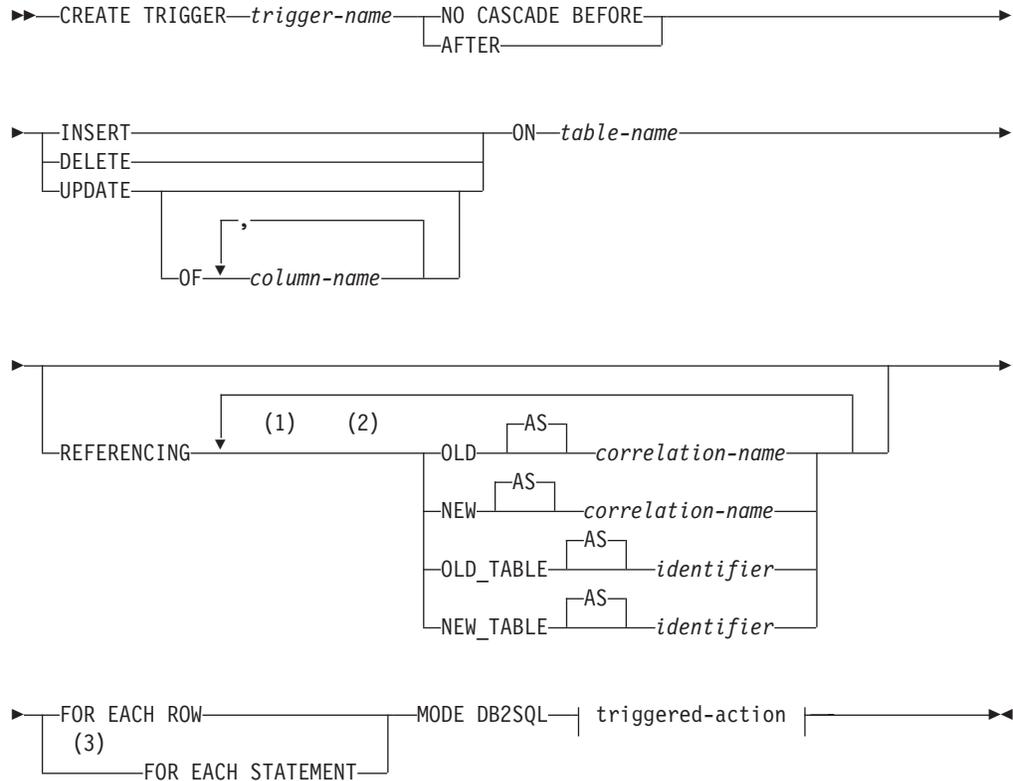
Beispiele: *Example 1:* Create a sequence called org_seq:

```
CREATE SEQUENCE org_seq
START WITH 1
INCREMENT BY 1
NO MAXVALUE
NO CYCLE
CACHE 24
```

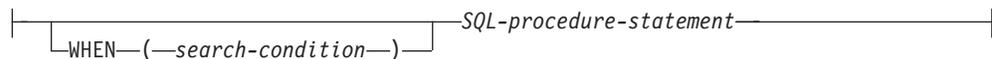
CREATE TRIGGER

CREATE TRIGGER

Syntax:



triggered-action:



Anmerkungen:

- 1 OLD and NEW may only be specified once each.
- 2 OLD_TABLE and NEW_TABLE may only be specified once each and only for AFTER triggers.
- 3 FOR EACH STATEMENT may not be specified for BEFORE triggers.

Replace the description of "triggered-action" with the following:

triggered-action

Specifies the action to be performed when a trigger is activated. A triggered-action is composed of an *SQL-procedure-statement* and an optional condition for the execution of the *SQL-procedure-statement*.

WHEN (*search-condition*)

Specifies a condition that is true, false, or unknown. The *search-condition* provides a capability to determine whether or not a certain triggered action should be executed.

The associated action is performed only if the specified search condition evaluates as true. If the WHEN clause is omitted, the associated *SQL-procedure-statement* is always performed.

SQL-procedure-statement

The *SQL-procedure-statement* can contain a dynamic compound statement or any of the SQL control statements listed in „Compound Statement (Dynamic)“ auf Seite 425.

If the trigger is a BEFORE trigger, then an *SQL-procedure-statement* can also include a fullselect or a SET variable statement (SQLSTATE 42987).

If the trigger is an AFTER trigger, then an *SQL-procedure-statement* can also include one of the following (SQLSTATE 42987):

- an INSERT SQL statement
- a searched UPDATE SQL statement
- a searched DELETE SQL statement
- a SET variable statement
- a fullselect ⁷

The *SQL-procedure-statement* cannot reference an undefined transition variable (SQLSTATE 42703) or a declared temporary table (SQLSTATE 42995).

The *SQL-procedure-statement* in a BEFORE trigger cannot reference a summary table defined with REFRESH IMMEDIATE (SQLSTATE 42997).

The *SQL-procedure-statement* in a BEFORE trigger cannot reference a generated column, other than the identity column, in the new transition variable (SQLSTATE 42989).

The Notes section changes to the following:

- The result of a fullselect specified in the *SQL-procedure-statement* is not available inside or outside of the trigger.
- **Inoperative triggers:** An *inoperative trigger* is a trigger that is no longer available and is therefore never activated. A trigger becomes inoperative if:
 - A privilege that the creator of the trigger is required to have for the trigger to execute is revoked.
 - An object such as a table, view or alias, upon which the triggered action is dependent, is dropped.
 - A view, upon which the triggered action is dependent, becomes inoperative.
 - An alias that is the subject table of the trigger is dropped.

In practical terms, an inoperative trigger is one in which a trigger definition has been dropped as a result of cascading rules for DROP or REVOKE statements. For example, when an view is dropped, any trigger with an *SQL-procedure-statement* defined using that view is made inoperative.

7. A common-table-expression may precede a fullselect.

When a trigger is made inoperative, all packages with statements performing operations that were activating the trigger will be marked invalid. When the package is rebound (explicitly or implicitly) the **inoperative trigger is completely ignored**. Similarly, applications with dynamic SQL statements performing operations that were activating the trigger will also completely ignore any inoperative triggers.

The trigger name can still be specified in the DROP TRIGGER and COMMENT ON TRIGGER statements.

An inoperative trigger may be recreated by issuing a CREATE TRIGGER statement using the definition text of the inoperative trigger. This trigger definition text is stored in the TEXT column of SYSCAT.TRIGGERS. Note that there is no need to explicitly drop the inoperative trigger in order to recreate it. Issuing a CREATE TRIGGER statement with the same *trigger-name* as an inoperative trigger will cause that inoperative trigger to be replaced with a warning (SQLSTATE 01595).

Inoperative triggers are indicated by an X in the VALID column of the SYSCAT.TRIGGERS catalog view.

- **Errors executing triggers:** Errors that occur during the execution of triggered SQL statements are returned using SQLSTATE 09000 unless the error is considered severe. If the error is severe, the severe error SQLSTATE is returned. The SQLERRMC field of the SQLCA for non-severe error will include the trigger name, SQLCODE, SQLSTATE and as many tokens as will fit from the tokens of the failure.

The *SQL-procedure-statement* could include a SIGNAL SQLSTATE statement or contain a RAISE_ERROR function. In both these cases, the SQLSTATE returned is the one specified in the SIGNAL SQLSTATE statement or the RAISE_ERROR condition.

CREATE WRAPPER

Linux uses libraries called LIBDRDA.S0 and LIBSQLNET.S0, not LIBDRDA.A and LIBSQLNET.A.

DECLARE CURSOR

Within the „DECLARE CURSOR“ statement, near the end of the Notes section the following sentence should be changed from:

An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL, otherwise it is considered deletable.

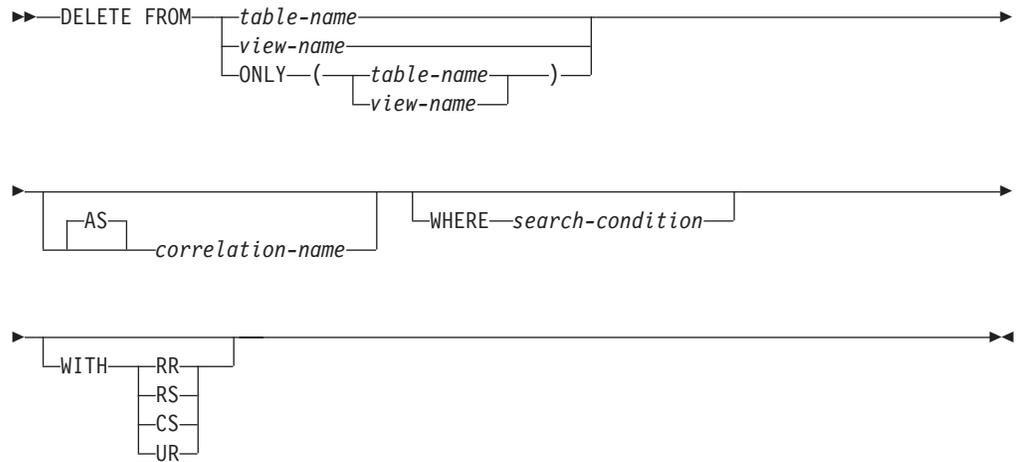
to:

An ambiguous cursor is considered read-only if the BLOCKING bind option is ALL; otherwise, it is considered updatable.

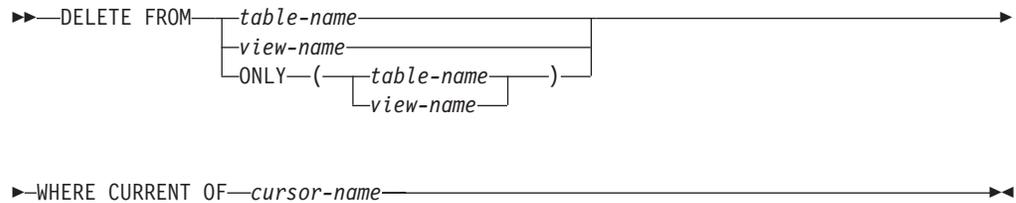
The change is from the word „deletable“ to the word „updatable“.

DELETE

The searched DELETE syntax diagram changes to the following:



Positioned DELETE:



Add the following to the description section:

WITH

Specifies the isolation level used when locating the rows to be deleted.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

DROP

Add the following option:

►►—SEQUENCE—*sequence-name*—RESTRICT—◄◄

Add the following parameters:

SEQUENCE *sequence-name* **RESTRICT**

Identifies the particular sequence that is to be dropped. The *sequence-name*, along with the implicit or explicit schema name, must identify an existing sequence at the current server. If no sequence by this name exists in the explicitly or implicitly specified schema, an error (SQLSTATE 42704) is raised.

The RESTRICT keyword enforces the rule that the sequence is not dropped if the definition of a table column refers to the sequence (through an IDENTITY column).

Anmerkung:

- System created sequences for IDENTITY columns cannot be dropped using the DROP sequence command.
- When a sequence is dropped, all privileges on the sequence are also dropped.

The table showing the dependencies that objects have on each other (Table 27) needs to be updated as follows:

New row: DROP SEQUENCE The entry at the intersection of the new row "DROP SEQUENCE" and the column "PACKAGE" will be an "A". The rest of the entries in this new row will be "-"

GRANT (Sequence Privileges)

GRANT (Sequence Privileges)

This form of the GRANT statement grants privileges on a user-defined sequence.

Aufruf: This statement can be embedded in an application program or issued through the use of dynamic SQL statements. It is an executable statement that can be dynamically prepared. However, if the bind option DYNAMICRULES BIND applies, the statement cannot be dynamically prepared (SQLSTATE 42509).

Berechtigung: The privileges held by the authorization ID of the statement must include at least one of the following:

- Owner of the sequence
- SYSADM or DBADM authority

Syntax:

```
►►—GRANT—USAGE—ON SEQUENCE—sequence-name—TO PUBLIC—————►►
```

Beschreibung:

USAGE

Grants the USAGE privilege for a sequence.

ON SEQUENCE *sequence-name*

Identifies the sequence on which the USAGE privilege is to be granted. The *sequence-name*, including the implicit or explicit schema qualifier, must uniquely identify an existing sequence at the current server. If no sequence by this name exists in the specified schema, an error (SQLSTATE 42704) is raised.

TO PUBLIC

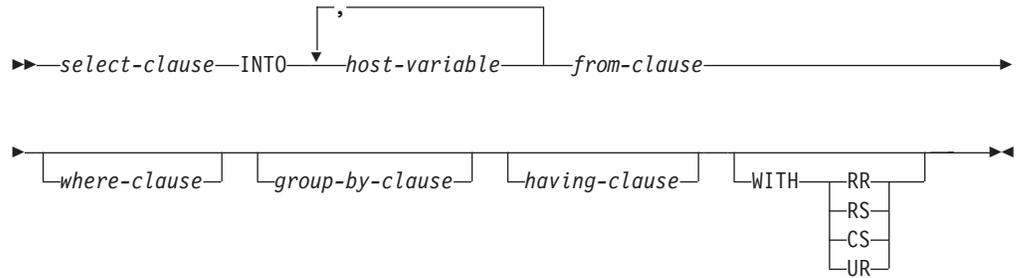
Grants the USAGE privilege to all users.

Beispiele: *Example 1:* Grant any user the privilege on a sequence called MYNUM

```
GRANT USAGE ON SEQUENCE MYNUM TO PUBLIC
```


SELECT INTO

The syntax diagram changes to:



Add the following to the description section:

WITH

Specifies the isolation level at which the `SELECT INTO` statement is executed.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

SET transition-variable

This section changes to the following:

SET Variable

The SET Variable statement assigns values to local variables or to new transition variables. It is under transaction control.

Aufruf: This statement can only be used as an SQL statement in either a dynamic compound statement, trigger, SQL function or SQL method.

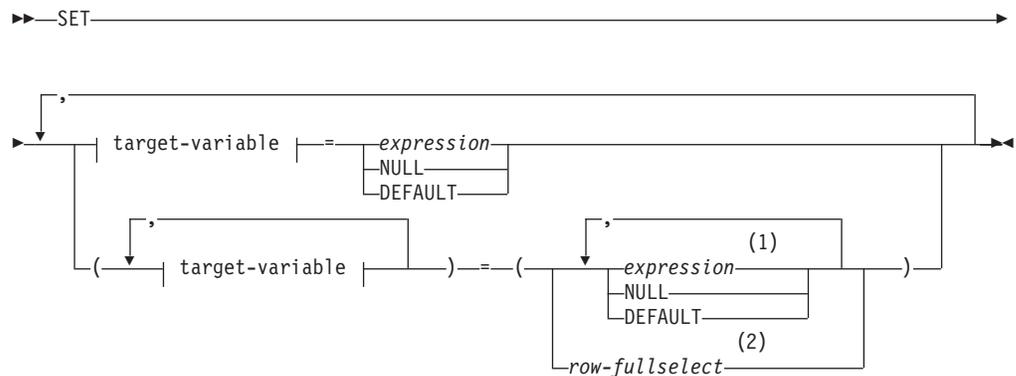
Berechtigung: To reference a transition variable, the privileges held by the authorization ID of the trigger creator must include at least one of the following:

- UPDATE of the columns referenced on the left hand side of the assignment and SELECT for any columns referenced on the right hand side.
- CONTROL privilege on the table (subject table of the trigger)
- SYSADM or DBADM authority.

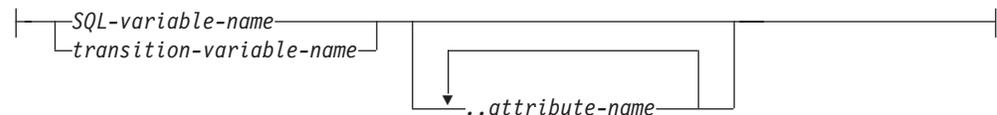
To execute this statement with a *row-fullselect* as the right hand side of the assignment, the privileges held by the authorization ID of either the trigger definer or the dynamic compound statement owner must also include at least one of the following, for each table or view referenced:

- SELECT privilege
- CONTROL privilege
- SYSADM or DBADM.

Syntax:



target-variable:



Anmerkungen:

- 1 The number of expressions, NULLs and DEFAULTs must match the number of *target-variables*.

- 2 The number of columns in the select list must match the number of *target-variables*.

Beschreibung:

target-variable

Identifies the target variable of the assignment. A *target-variable* representing the same variable must not be specified more than once (SQLSTATE 42701).

SQL-variable-name

Identifies the SQL variable that is the assignment target. SQL variables must be declared before they are used. SQL variables can be defined in a dynamic compound statement.

transition-variable-name

Identifies the column to be updated in the transition row. A *transition-variable-name* must identify a column in the subject table of a trigger, optionally qualified by a correlation name that identifies the new value (SQLSTATE 42703).

..attribute name

Specifies the attribute of a structured type that is set (referred to as an *attribute assignment*). The *SQL-variable-name* or *transition-variable-name* specified must be defined with a user-defined structured type (SQLSTATE 428DP). The *attribute-name* must be an attribute of the structured type (SQLSTATE 42703). An assignment that does not involve the *..attribute name* clause is referred to as a conventional assignment.

expression

Indicates the new value of the *target-variable*. The expression is any expression of the type described in Chapter 2 of the SQL Reference. The expression can not include a column function except when it occurs within a scalar fullselect (SQLSTATE 42903). In the context of a CREATE TRIGGER statement, an *expression* may contain references to OLD and NEW transition variables and must be qualified by the *correlation-name* to specify which transition variable (SQLSTATE 42702).

NULL

Specifies the null value and can only be specified for nullable columns (SQLSTATE 23502). NULL cannot be the value in an attribute assignment (SQLSTATE 429B9), unless it was specifically cast to the data type of the attribute.

DEFAULT

Specifies that the default value should be used.

If *target-variable* is a column, the value inserted depends on how the column was defined in the table.

- If the column was defined using the WITH DEFAULT clause, then the value is set to the default defined for the column.
- If the column was defined using the IDENTITY clause, the value is generated by the database manager.
- If the column was defined without specifying the WITH DEFAULT clause, the IDENTITY clause, or the NOT NULL clause, then the value is NULL.
- If the column was defined using the NOT NULL clause and the IDENTITY clause is not used, or the WITH DEFAULT clause was not used or DEFAULT NULL was used, the DEFAULT keyword cannot be specified for that column (SQLSTATE 23502).

If *target-variable* is an SQL variable, then the value inserted is the default as specified or implied in the variable declaration.

row-fullselect

A fullselect that returns a single row with the number of columns corresponding to the number of target-variables specified for assignment. The values are assigned to each corresponding target-variable. If the result of the row-fullselect is no rows, then null values are assigned. In the context of a CREATE TRIGGER statement, a *row-fullselect* may contain references to OLD and NEW transition variables which must be qualified by their *correlation-name* to specify which transition variable to use (SQLSTATE 42702). An error is returned if there is more than one row in the result (SQLSTATE 21000).

Regeln:

- The number of values to be assigned from expressions, NULLs and DEFAULTs or the *row-fullselect* must match the number of target-variables specified for assignment (SQLSTATE 42802).
- A SET Variable statement cannot assign an SQL variable and a transition variable in one statement (SQLSTATE 42997).
- Values are assigned to target-variables under the assignment rules described in Chapter 2 of the SQL Reference.

If the statement is used in a BEFORE UPDATE trigger, and the registry variable DB2_UPDATE_PART_KEY=OFF, then a *transition-variable* specified as *target-variable* cannot be a partitioning key column (SQLSTATE 42997).

Anmerkungen:

- If more than one assignment is included, all *expressions* and *row-fullselects* are evaluated before the assignments are performed. Thus references to target-variables in an expression or row fullselect are always the value of the target-variable prior to any assignment in the single SET statement.
- When an identity column defined as a distinct type is updated, the entire computation is done in the source type, and the result is cast to the distinct type before the value is actually assigned to the column.⁸
- To have DB2 generate a value on a SET statement for an identity column, use the DEFAULT keyword:

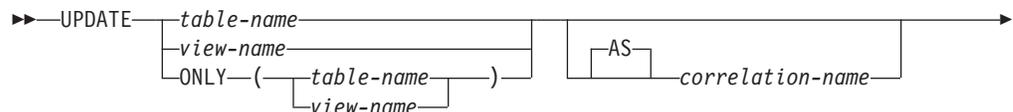
```
SET NEW.EMPNO = DEFAULT
```

In this example, NEW.EMPNO is defined as an identity column, and the value used to update this column is generated by DB2.

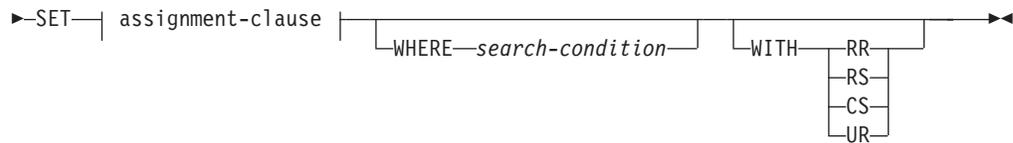
The examples for this statement stay the same.

UPDATE

The searched UPDATE syntax diagram is changed to:



8. There is no casting of the previous value to the source type prior to the computation.



Add the following to the description section:

WITH

Specifies the isolation level at which the UPDATE statement is executed.

RR

Repeatable Read

RS

Read Stability

CS

Cursor Stability

UR

Uncommitted Read

The default isolation level of the statement is the isolation level of the package in which the statement is bound.

Chapter 7. SQL Procedures now called Chapter 7. SQL Control Statements

Control statements are SQL statements that allow SQL to be used in a manner similar to writing a program in a structured programming language. SQL control statements can be used in the body of a routine, trigger or a dynamic compound statement.

This chapter contains the syntax and descriptions of the supported SQL control statements, along with the SQL-procedure-statement.

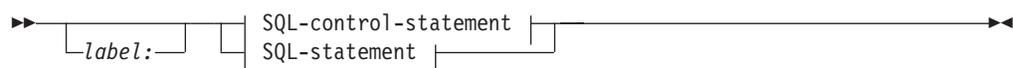
SQL Procedure Statement

The SQL Procedure Statement information changes to the following:

SQL Procedure Statement

This chapter contains syntax diagrams, semantic descriptions, rules, and examples of the use of the statements that constitute the procedure body of an SQL routine, trigger, or dynamic compound statement.

Syntax:



SQL-control-statement:

ALLOCATE CURSOR statement	(1)
assignment statement	(1)
ASSOCIATE LOCATORS statement	(1)
CASE statement	(1)
dynamic-compound statement	(2)
FOR statement	
GET DIAGNOSTICS statement	(1)
GOTO statement	
IF statement	
ITERATE statement	
LEAVE statement	(1)
LOOP statement	(1)
procedure-compound statement	(1)
REPEAT statement	(1)
RESIGNAL statement	
RETURN statement	
SIGNAL statement	
WHILE statement	

Anmerkungen:

- 1 This statement is only supported in the scope of an SQL Procedure.
- 2 This statement is only supported within a trigger, SQL function, or SQL method. It must be the outermost statement.

Beschreibung:

label:

Specifies the label for an SQL procedure statement. The label must be unique within a list of SQL procedure statements, including any compound statements nested within the list. Note that compound statements that are not nested may use the same label. A list of SQL procedure statements is possible in a number of SQL control statements.

In the context of a trigger, an SQL function or method, or a dynamic compound statement, only the dynamic compound statement, the FOR statement and the WHILE statement may be labeled.

SQL-statement

In the body of an SQL procedure, all executable SQL statements can be contained, with the exception of the following:

- CONNECT
- CREATE any object other than indexes, tables, or views
- DESCRIBE
- DISCONNECT
- DROP any object other than indexes, tables, or views
- FLUSH EVENT MONITOR

- REFRESH TABLE
- RELEASE (connection only)
- RENAME TABLE
- RENAME TABLESPACE
- REVOKE
- SET CONNECTION
- SET INTEGRITY

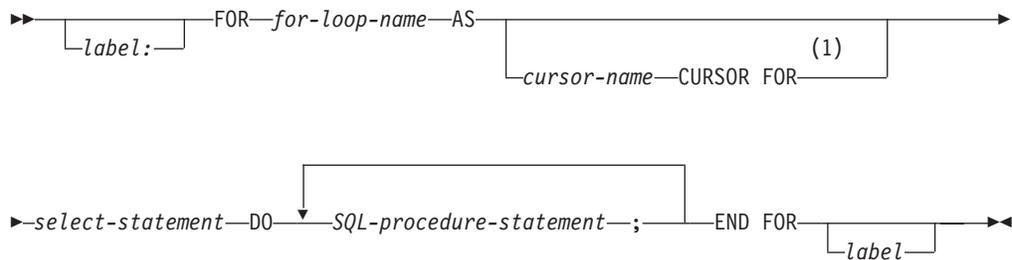
Anmerkung: You may include CALL statements within an SQL procedure body, but these CALL statements can only call another SQL procedure or a C procedure. CALL statements within an SQL procedure body cannot call other types of stored procedures.

FOR

FOR

The FOR statement executes a statement or group of statements for each row of a table.

Syntax:



Anmerkungen:

- 1 This option can only be used in the context of an SQL Procedure.

Beschreibung:*label*

Specifies the label for the FOR statement. If the beginning label is specified, that label can be used in LEAVE and ITERATE statements. If the ending label is specified, it must be the same as the beginning label.

for-loop-name

Specifies a label for the implicit compound statement generated to implement the FOR statement. It follows the rules for the label of a compound statement except that it cannot be used with an ITERATE or LEAVE statement within the FOR statement. The *for-loop-name* is used to qualify the column names returned by the specified *select-statement*.

cursor-name

Names the cursor that is used to select rows from the result table from the SELECT statement. If not specified, DB2 generates a unique cursor name.

select-statement

Specifies the SELECT statement of the cursor. All columns in the select list must have a name and there cannot be two columns with the same name.

In a trigger, function, method, or dynamic compound statement, the *select-statement* must consist of only a *fullselect* with optional common table expressions.

SQL-procedure-statement

Specifies a statement (or statements) to be invoked for each row of the table.

Regeln:

- The select list must consist of unique column names and the table specified in the select list must exist when the procedure is created, or it must be a table created in a previous SQL procedure statement.
- The cursor specified in a for-statement cannot be referenced outside the for-statement and cannot be specified in an OPEN, FETCH, or CLOSE statement.

Beispiele: In the following example, the for-statement is used to iterate over the entire employee table. For each row in the table, the SQL variable `fullname` is set to the last name of the employee, followed by a comma, the first name, a blank space, and the middle initial. Each value for `fullname` is inserted into table `tnames`.

```

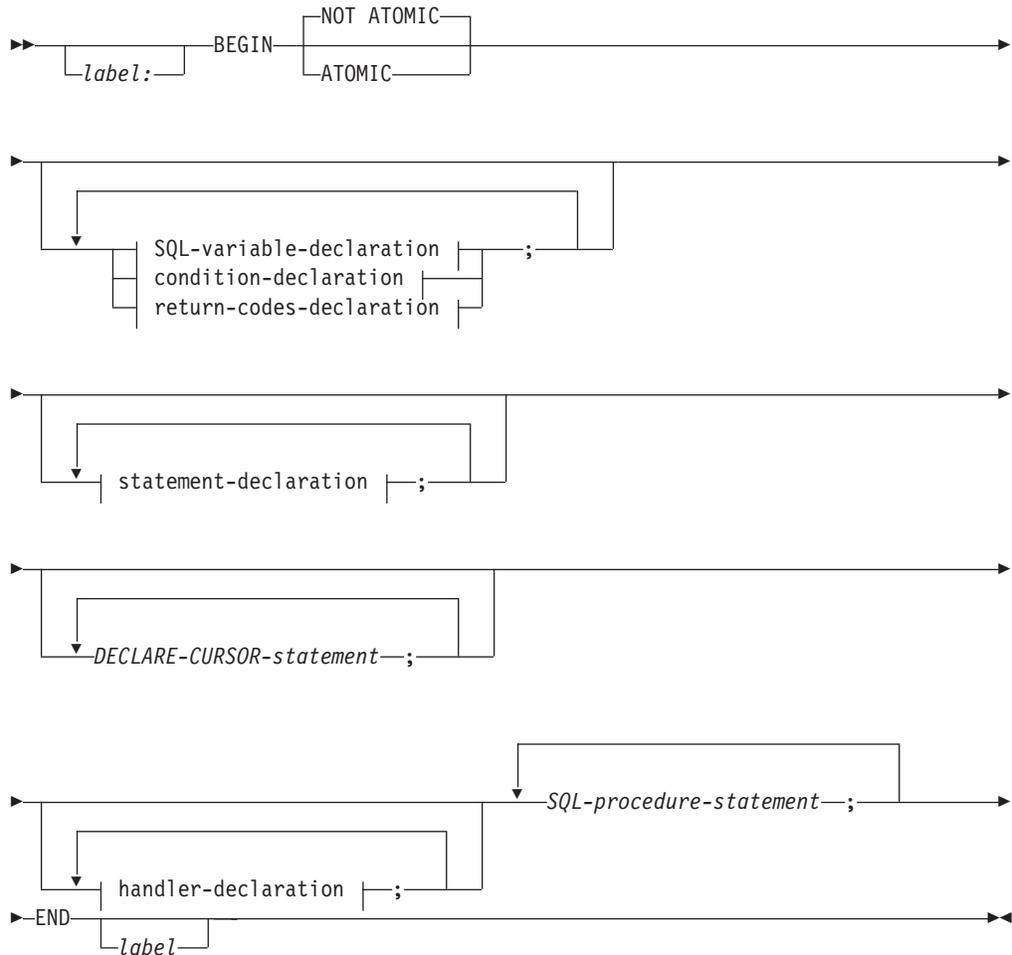
BEGIN
  DECLARE fullname CHAR(40);
  FOR v1 AS
    SELECT firstnme, midinit, lastname FROM employee
    DO
      SET fullname = lastname || ',' || firstnme || ' ' || midinit;
      INSERT INTO tnames VALUE (fullname);
  END FOR
END

```

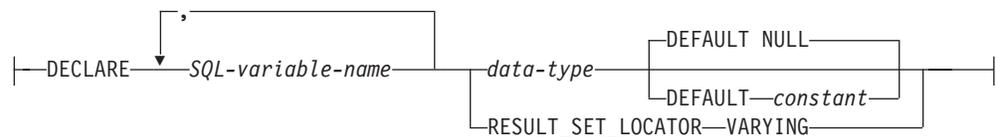
Compound Statement changes to Compound Statement (Procedure)

A procedure compound statement groups other statements together in an SQL procedure. You can declare SQL variables, cursors, and condition handlers within a compound statement.

The syntax diagram now has a title: procedure-compound-statement.

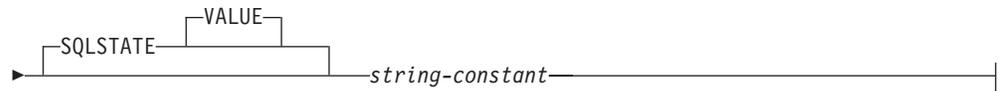


SQL-variable-declaration:



condition-declaration:





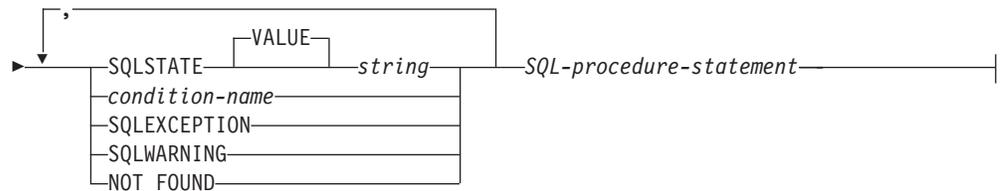
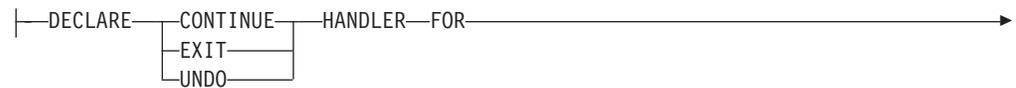
statement-declaration:



return-codes-declaration:



handler-declaration:



statement-declaration

A *statement-declaration* declares a list of one or more names that are local to the compound statement. A statement name cannot be the same as another statement name within the same compound statement.

Regeln:

- The execution of an SQL function or method must end with a RETURN (SQLSTATE 42632).
- In an SQL table or row function using a *dynamic-compound-statement*, the only RETURN statement allowed is the one at the end of the compound statement (SQLSTATE 429BD).

Anmerkungen:

- When a value is returned from a procedure, the caller may access the value using:
 - the GET DIAGNOSTICS statement to retrieve the RETURN_STATUS when the SQL procedure was called from another SQL procedure
 - the parameter bound for the return value parameter marker in the escape clause CALL syntax (?=CALL...) in a CLI application
 - directly from the SQLCA returned from processing the CALL of an SQL procedure by retrieving the value of SQLERRD[0] when the SQLCODE is not less than zero (assume a value of -1 when SQLCODE is less than zero).

Beispiele: Use a RETURN statement to return from an SQL stored procedure with a status value of zero if successful, and -200 if not.

```
BEGIN
...
  GOTO FAIL
...
  SUCCESS: RETURN 0
  FAIL: RETURN -200
END
```

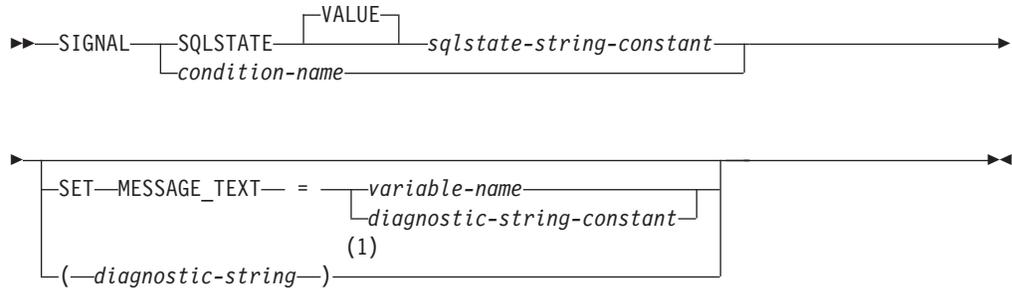
SIGNAL

The SIGNAL SQLSTATE Statement is no longer used, in favor of this usage.

SIGNAL

The SIGNAL statement is used to signal an error or warning condition. It causes an error or warning to be returned with the specified SQLSTATE, along with optional message text.

Syntax:



Anmerkungen:

- 1 This option is only provided within the scope of a CREATE TRIGGER statement for compatibility with older versions of DB2.

Beschreibung:

SQLSTATE VALUE *sqlstate-string-constant*

The specified string constant represents an SQLSTATE. It must be a character string constant with exactly 5 characters that follow the rules for SQLSTATES:

- Each character must be from the set of digits ('0' through '9') or non-accented upper case letters ('A' through 'Z').
- The SQLSTATE class (first two characters) cannot be '00', since this represents successful completion.

In the context of either a dynamic compound statement, trigger, SQL function, or SQL method, the following rules must also be applied:

- The SQLSTATE class (first two characters) cannot be '01' or '02', since these are not error classes.
- If the SQLSTATE class starts with the numbers '0' through '6' or the letters 'A' through 'H', then the subclass (the last three characters) must start with a letter in the range of 'I' through 'Z'.
- If the SQLSTATE class starts with the numbers '7', '8', '9', or the letters 'I' through 'Z', then the subclass can be any of '0' through '9' or 'A' through 'Z'.

If the SQLSTATE does not conform to these rules, an error is raised (SQLSTATE 428B3).

condition-name

Specifies the name of the condition. The condition name must be unique within the procedure and can only be referenced within the compound statement in which it is declared.

SET MESSAGE_TEXT=

Specifies a string that describes the error or warning. The string is returned in the SQLERRMC field of the SQLCA. If the actual string is longer than 70 bytes, it is truncated without warning. This clause can only be specified if a SQLSTATE or condition-name is also specified (SQLSTATE 42601).

variable-name

Identifies an SQL variable that must be declared within the compound statement. The SQL variable must be defined as a CHAR or VARCHAR data type.

diagnostic-string-constant

Specifies a character string constant that contains the message text.

diagnostic-string

An expression with a type of CHAR or VARCHAR that returns a character string of up to 70 bytes to describe the error condition. If the string is longer than 70 bytes, it will be truncated. This option is only provided within the scope of a CREATE TRIGGER statement, for compatibility with older versions of DB2. Regular use is not recommended.

Anmerkungen:

- If a SIGNAL statement is issued, the SQLCODE that is assigned is:
 - +438 if the SQLSTATE begins with '01' or '02'
 - 438 otherwise
- If the SQLSTATE or condition indicates that an exception (SQLSTATE class other than '01' or '02') is signaled:
 - Then the exception is handled and control is transferred to a handler, provided that a handler exists in the same compound statement (or an outer compound statement) as the signal statement, and the compound statement contains a handler for the specified SQLSTATE, condition-name, or SQLEXCEPTION;
 - If the exception cannot be handled, then control is immediately returned to the end of the compound statement.
- If the SQLSTATE or condition indicates that a warning (SQLSTATE class '01') or not found condition (SQLSTATE class '02') is signaled:
 - Then the warning or not found condition is handled and control is transferred to a handler, provided that a handler exists in the same compound statement (or an outer compound statement) as the signal statement, and the compound statement contains a handler for the specified SQLSTATE, condition-name, SQLWARNING (if the SQLSTATE class is '01'), or NOT FOUND (if the SQLSTATE class is '02');
 - If the warning cannot be handled, then processing continues with the next statement.

- SQLSTATE values are comprised of a two-character class code value, followed by a three-character subclass code value. Class code values represent classes of successful and unsuccessful execution conditions.

Any valid SQLSTATE value can be used in the SIGNAL statement. However, it is recommended that programmers define new SQLSTATEs based on ranges reserved for applications. This prevents the unintentional use of an SQLSTATE value that might be defined by the database manager in a future release.

- SQLSTATE classes that begin with the characters '7' through '9', or 'I' through 'Z' may be defined. Within these classes, any subclass may be defined.
- SQLSTATE classes that begin with the characters '0' through '6', or 'A' through 'H' are reserved for the database manager. Within these classes, subclasses that begin with the characters '0' through 'H' are reserved for the database manager. Subclasses that begin with the characters 'I' through 'Z' may be defined.

Beispiele: An SQL procedure for an order system that signals an application error when a customer number is not known to the application. The ORDERS table includes a foreign key to the CUSTOMER table, requiring that the CUSTNO exist before an order can be inserted.

```

CREATE PROCEDURE SUBMIT_ORDER
  (IN ONUM INTEGER, IN CNUM INTEGER,
   IN PNUM INTEGER, IN QNUM INTEGER)
  SPECIFIC SUBMIT_ORDER
  MODIFIES SQL DATA
  LANGUAGE SQL
  BEGIN
    DECLARE EXIT HANDLER FOR SQLSTATE VALUE '23503'
      SIGNAL SQLSTATE '75002'
      SET MESSAGE_TEXT = 'Customer number is not known';
    INSERT INTO ORDERS (ORDERNO, CUSTNO, PARTNO, QUANTITY)
      VALUES (ONUM, CNUM, PNUM, QNUM);
  END

```

Appendix A. SQL Limits

There is a change to Table 33, Database Manager Limits. With the registry variable DB2_INDEX_2BYTEVARLEN set to ON, the longest variable index key part (in bytes) can now be greater than 255.

Appendix D. Catalog Views

A new catalog view has been added:

SYSCAT.SEQUENCES

The view SYSCAT.SEQUENCES is automatically generated for databases created with FixPak 3 or later. For databases created prior to FixPak 3, run the **db2updv7** command in order to add the view to the database. See the Command Reference update in the Release Notes for details.

This catalog view is updated during normal operation, in response to SQL data definition statements, environment routines, and certain utilities. Data in the catalog view is available through normal SQL query facilities. Columns have consistent names based on the type of objects that they describe.

Tabelle 30. Columns in SYSCAT.SEQUENCES Catalog View

Column Name	Data Type	Nullable	Description
SEQSHEMA	VARCHAR(128)		Schema of the sequence.
SEQNAME	VARCHAR(128)		Sequence name (generated by DB2 for an identity column).
DEFINER	VARCHAR(128)		Definer of the sequence.
OWNER	VARCHAR(128)		Owner of the sequence.
SEQID	INTEGER		Internal ID of the sequence.
SEQTYPE	CHAR(1)		Sequence type S - Regular sequence
INCREMENT	DECIMAL(31,0)		Increment value.
START	DECIMAL(31,0)		Starting value.
MAXVALUE	DECIMAL(31,0)	Yes	Maximal value.
MINVALUE	DECIMAL(31,0)		Minimum value.
CYCLE	CHAR(1)		Whether cycling will occur when a boundary is reached: Y - cycling will occur N - cycling will not occur
CACHE	INTEGER		Number of sequence values to preallocate in memory for faster access. 0 indicates that values are not preallocated.
ORDER	CHAR(1)		Whether or not the sequence numbers must be generated in order of request: Y - sequence numbers must be generated in order of request N - sequence numbers are not required to be generated in order of request
DATATYPEID	INTEGER		For built-in types, the internal ID of the built-in type. For distinct types, the internal ID of the distinct type.

Tabelle 30. Columns in SYSCAT.SEQUENCES Catalog View (Forts.)

Column Name	Data Type	Nullable	Description
SOURCETYPEID	INTEGER		For a built-in type, this has a value of 0. For a distinct type, this is the internal ID of the built-in type that is the source type for the distinct type.
CREATE_TIME	TIMESTAMP		Time when the sequence was created.
ALTER_TIME	TIMESTAMP		Time when the last ALTER SEQUENCE statement was executed for this sequence.
PRECISION	SMALLINT		The precision defined for a sequence with a decimal or numeric type. Values are: 5 for a SMALLINT, 10 for INTEGER, and 19 for BIGINT.
ORIGIN	CHAR(1)		Sequence Origin U - User generated sequence S - System generated sequence
REMARKS	VARCHAR(254)	Yes	User supplied comments, or null.

DB2 Stored Procedure Builder

Java 1.2-Unterstützung für DB2 Stored Procedure Builder

DB2 Stored Procedure Builder unterstützt die Erstellung gespeicherter Java-Prozeduren mit Java 1.2-Funktionalität. Daneben unterstützt Stored Procedure Builder bidirektionale Sprachen wie Arabisch und Hebräisch mit Hilfe der Bidi-Unterstützung von Java 1.2.

Diese Unterstützung gilt nur für Windows NT-Plattformen.

Stored Procedure Builder kann die Java 1.2-Funktionalität nur dann erkennen und verwenden, wenn Java 1.2 installiert ist.

Gehen Sie wie folgt vor, um Java 1.2 zu installieren:

1. JDK 1.2.2 ist auf der DB2 UDB-CD im Verzeichnis DB2\bid\NT verfügbar. `ibm-inst-n122p-win32-x86.exe` ist das Installationsprogramm, `ibm-jdk-n122p-win32-x86.exe` die Datei für die Verteilung von JDK. Kopieren Sie beide Dateien in ein temporäres Verzeichnis auf Ihrer Festplatte und führen Sie das Installationsprogramm von dort aus.
2. Installieren Sie das Produkt unter `<DB2PATH>\java\Java12`. Dabei ist `<DB2PATH>` der Installationspfad von DB2.
3. Wählen Sie JDK/JRE nicht als System-VM aus, wenn Sie von der JDK/JRE-Installation dazu aufgefordert werden.

Nach der erfolgreichen Installation von Java 1.2 erfolgreich starten Sie Stored Procedure Builder auf die übliche Art.

Setzen Sie zum Ausführen gespeicherter Java-Prozeduren mit JDK 1.2-Unterstützung die Umgebungsvariable `DB2_USE_JDK12` des Datenbank-Servers mit folgendem Befehl auf `TRUE`:

```
DB2SET DB2_USE_JDK12=TRUE
```

Setzen Sie außerdem Ihren Pfad `JDK11_PATH` so, dass er auf das Verzeichnis zeigt, in dem Ihre Java 1.2-Unterstützung installiert ist. Setzen Sie diesen Pfad mit folgendem Befehl:

```
DB2 UPDATE DBM CFG USING JDK11_PATH
```

Wenn Sie Java 1.2 nicht mehr verwenden wollen, können Sie JDK/JRE im Verzeichnis `<DB2PATH>\java\Java12` deinstallieren oder das Unterverzeichnis `<DB2PATH>\java\Java12` umbenennen.

Achtung: Verwechseln Sie nicht `<DB2PATH>\java\Java12` und `<DB2PATH>\Java12`. `<DB2PATH>\Java12` ist Teil der DB2-Installation und enthält die JDBC-Unterstützung für Java 1.2.

Fernes Debug von gespeicherten DB2-Prozeduren

Wenn Sie auf UNIX- und Windows-Plattformen die Funktion für fernes Debug in gespeicherten Java- oder C-Prozeduren verwenden wollen, müssen Sie IBM Distributed Debugger installieren. IBM Distributed Debugger befindet sich auf der CD *Visual Age for Java Professional Edition*. Der Debugger-Client kann nur unter Windows ausgeführt werden. Folgende Server-Plattformen werden u. a. unterstützt: Windows, AIX und Solaris. Verwenden Sie die in Stored Procedure Builder integrierte SQL-Debug-Funktion, um Fehler in lokalen und fernen gespeicherten SQL-Prozeduren für Windows- und UNIX-Plattformen zu beheben. Derzeit ist keine Unterstützung der OS/2-Plattform verfügbar. Weitere Informationen zu Stored Procedure Builder von DB2 für OS/390 finden Sie auf folgender Website:
<http://www-4.ibm.com/software/data/db2/os390/spb/exciting>

Zur Fehlerbehebung in SQL-Prozeduren auf der OS/390-Plattform benötigen Sie außerdem IBM C/C++ Productivity Tools for OS/390 R1. Weitere Informationen zu IBM C/C++ Productivity Tools für OS/390 R1 finden Sie auf folgender Website:
<http://www.ibm.com/software/ad/c390/pt/>

Erstellen von SQL-Prozeduren auf Windows-, OS/2- oder UNIX-Plattformen

Wenn Sie mit Stored Procedure Builder in Ihrer Windows-, OS/2- oder UNIX-Datenbank erfolgreich SQL-Prozeduren erstellen wollen, müssen Sie Ihren Server zuvor für SQL-Prozeduren konfigurieren. Informationen zum Konfigurieren Ihres Servers für SQL-Prozeduren finden Sie unter „Chapter 4. Building Java Applets and Applications“ auf Seite 344.

Verwenden von DB2 Stored Procedure Builder auf der Solaris-Plattform

Gehen Sie wie folgt vor, um Stored Procedure Builder auf der Solaris-Plattform zu verwenden:

1. Laden Sie JDK 1.1.8 herunter und installieren Sie es. Sie können JDK 1.1.8 von der JavaSoft-Website herunterladen.
2. Setzen Sie die Umgebungsvariable `JAVA_HOME` auf die Speicherposition, auf der Sie JDK installiert haben.
3. Setzen Sie Ihre DB2-Variablen `JDK11_PATH` auf das Verzeichnis, in dem Sie JDK installiert haben.

Setzen Sie `JDK11_PATH` mit folgendem Befehl:

```
DB2 UPDATE DBM CFG USING JDK11_PATH
```

Bekanntere Probleme und Einschränkungen

- SQL-Prozeduren werden momentan nicht unter Windows 98 unterstützt.
- Bei gespeicherten Java-Prozeduren dürfen die JAR-ID, Klassennamen und Methodennamen nur ASCII-Zeichen enthalten.
- Bei AS/400 müssen die folgenden vorläufigen Programmkorrekturen von Version 4 Release 4 auf OS/400 V4R4 angewendet werden:
 - SF59674
 - SF59878

- Parameter für gespeicherte Prozeduren mit einem Zeichensubtyp von FOR MIXED DATA oder FOR SBCS DATA werden nicht im Quellcode im Editorfenster angezeigt, wenn die gespeicherte Prozedur aus der Datenbank wiederhergestellt wird.
- Derzeit tritt ein Problem auf, wenn Java-Quellcode aus einer Datenbank abgerufen wird. Beim Abrufen wurden die Kommentare im Code ausgeblendet. Dies betrifft Benutzer von DB2 Stored Procedure Builder, die mit Nicht-ASCII-Codepages arbeiten und deren Clients und Server unterschiedliche Codepages verwenden.

Using DB2 Stored Procedure Builder with Traditional Chinese Locale

There is a problem when using Java Development Kit or Java Runtime 1.1.8 with the Traditional Chinese locale. Graphical aspects of the Stored Procedure Builder program (including menus, editor text, messages, and so on) will not display properly. The solution is to make a change to the file `font.properties.zh_TW`, which appears in one or both of the following directories:

```
sqllib/java/jdk/lib
sqllib/java/jre/lib
```

Change:

```
monospaced.0=\u7d30\u660e\u9ad4,CHINESEBIG5_CHARSET,NEED_CONVERTED
```

to:

```
monospaced.0=Courier New,ANSI_CHARSET
```

UNIX-Installationen (AIX, Sun Solaris, Linux) und Stored Procedure Builder

Bei Sun Solaris-Installationen und wenn Sie ein anderes Java Development Kit oder Runtime als das unter AIX mit UDB installierte verwenden, müssen Sie die Umgebungsvariable `JAVA_HOME` auf den Pfad einstellen, in dem Java installiert ist (d. h. auf das Verzeichnis mit den Unterverzeichnissen `/bin` und `/lib`). Stored Procedure Builder wird unter Linux nicht unterstützt, kann jedoch auf unterstützten Plattformen zum Erstellen und Ausführen gespeicherter Prozeduren auf Systemen unter DB2 UDB für Linux verwendet werden.

Unterstützte Plattformen sind: AIX, Solaris und NT für den Client und AIX, Solaris, NT, Linux, OS/2, HP-UX und NUMA-Q für den Server.

Erstellen von gespeicherten SQL-Prozeduren unter OS/390

DB2 Stored Procedure Builder unterstützt die Erstellung von gespeicherten SQL-Prozeduren auf dem DB2 UDB-Server für OS390 Version 7.

Debug von gespeicherten SQL-Prozeduren

Das Debug von gespeicherten SQL-Prozeduren auf Windows- und UNIX-Plattformen ist jetzt direkt in DB2 Stored Procedure Builder integriert. Die Konfigurationsoption `KEEPDARI` des Datenbankmanagers kann beim Debug nicht abgeschirmter (gesicherter) SQL-Prozeduren auf YES oder NO gesetzt werden. Beim Debug abgeschirmter (ungesicherter) SQL-Prozeduren muss sie jedoch auf YES (Standardeinstellung) gesetzt sein. Weitere Informationen zur Verwendung des integrierten Debugger finden Sie in der Onlinehilfe von Stored Procedure Builder.

Exportieren gespeicherter Java-Prozeduren

DB2 Stored Procedure Builder unterstützt jetzt den Export gespeicherter Java-Prozeduren. Gehen Sie wie folgt vor, um eine gespeicherte Java-Prozedur zu exportieren:

1. Klicken Sie den Ordner für gespeicherte Prozeduren mit Maustaste 2 an, und klicken Sie danach **Gespeicherte Java-Prozeduren exportieren** an, um das Fenster **Gespeicherte Java-Prozeduren exportieren** zu öffnen.
2. Wählen Sie die gespeicherten Prozeduren aus, die Sie exportieren wollen, und verschieben Sie sie in die Spalte für die ausgewählten gespeicherten Prozeduren.
3. Wählen Sie Ihre gewünschten Optionen aus, und klicken Sie **OK** an.

Einfügen gespeicherter Prozeduren unter OS/390

Wenn Sie bei DB2 Stored Procedure Builder Version 5 und höher unter OS/390 den Assistenten zum Einfügen einer gespeicherten Prozedur verwenden und keine WLM-Umgebungsoptionen angeben, enthält der generierte Code folgenden Text: NO WLM ENVIRONMENT. Diese Codezeile veranlasst die gespeicherte Prozedur, wie erwartet im SPAS-Adressraum zu laufen. Dieser Fix behebt ein Problem, das ab DB2 Stored Procedure Builder Version 6 auftrat.

Der generierte Code nach dem Fix sieht wie folgt aus:

```
CREATE PROCEDURE SYSPROC.Proc2 ( )
  RESULT SETS 1
  LANGUAGE SQL
  MODIFIES SQL DATA
  COLLID TEST
  NO WLM ENVIRONMENT
  ASUTIME NO LIMIT
  RUN OPTIONS 'NOTEST(ALL,*,VADTCPPIP&9.112.14.91:*)'
-----
-- SQL Stored Procedure
-----
P1: BEGIN
  -- Declare cursor
  DECLARE cursor1 CURSOR WITH RETURN FOR
    SELECT * FROM SYSIBM.SYSPROCEDURES;

  -- Cursor left open for client application
  OPEN cursor1;

END P1
```

Einstellen von Erstellungsoptionen für gespeicherte SQL-Prozeduren auf einem Workstation-Server

Bei Verwendung von DB2 Stored Procedure Builder auf UNIX- und Windows-Plattformen können Sie Erstellungsoptionen für alle gespeicherten SQL-Prozeduren einstellen. Diese Erstellungsoptionen enthalten die folgenden DB2-Registrierungsvariablen für Compiler- und Precompiler:

- DB2_SQLROUTINE_PREOPTS
- DB2_SQLROUTINE_COMPILER_PATH
- DB2_SQLROUTINE_COMPILE_COMMAND
- DB2_SQLROUTINE_KEEP_FILES

Es ist zwar möglich, diese Registrierungsvariablen mit dem Befehl **db2set** einzustellen; durch die Verwendung von Stored Procedure Builder erübrigt sich jedoch der physische Zugriff auf den Datenbank-Server, um den Befehl abzusetzen oder um den Server zu stoppen und erneut zu starten, um die Änderungen wirksam zu machen.

Wenn Sie das Fenster **SQL-SP-Build-Optionen** öffnen wollen, klicken Sie mit Maustaste 2 eine Datenbankverbindung in Ihrer Projektsicht und anschließend **SQL-SP-Build-Optionen** an. Weitere Informationen zum Einstellen dieser Optionen finden Sie in der Hilfe von DB2 Stored Procedure Builder.

Automatisches Aktualisieren des WLM-Adressraums für gespeicherte Prozeduren unter OS/390

Nachdem Sie erfolgreich eine gespeicherte Prozedur unter OS/390 erstellt haben, die im WLM ausgeführt wird, aktualisiert DB2 Stored Procedure Builder den WLM-Adressraum automatisch.

Entwicklung von gespeicherten Java-Prozeduren unter OS/390

DB2 Stored Procedure Builder unterstützt die Entwicklung von gespeicherten Java-Prozeduren in DB2 UDB für OS/390 Version 6 und höher. Sie können neue gespeicherte Java-Prozeduren erstellen oder vorhandene ändern.

Aufbau einer benutzerdefinierten Funktion in einer DB2-Tabelle für MQSeries und OLE DB

DB2 Stored Procedure Builder enthält Assistenten, die Ihnen helfen, benutzerdefinierte Tabellenfunktionen für MQSeries und OLE DB zu erstellen. Sie können den Assistenten zum Erstellen von benutzerdefinierten Tabellenfunktionen in OLE-Datenbanken verwenden, um auf OLE DB-Datenanbieter zuzugreifen. Der Assistent erstellt eine benutzerdefinierte Funktion für eine OLE-Tabelle und optional eine Tabellensicht. Sie können den Assistenten zum Erstellen von benutzerdefinierten MQSeries-Tabellenfunktionen verwenden, um eine benutzerdefinierte Tabellenfunktion mit einer optionalen Tabellensicht für den Zugriff auf MQSeries-Nachrichten und die syntaktische Analyse der Daten in einem tabellarischen Format zu erstellen.

Unicode-Aktualisierungen

Einführung

Der Unicode-Standard ist das Universalschema für Codeumsetzung für geschriebene Zeichen und Text. In Unicode wird jedes Zeichen durch mehrere Bytes dargestellt. Dieser Standard definiert eine konsistente Methode für das Codieren von mehrsprachigen Texten, die den internationalen Austausch von Textdaten ermöglicht und die Grundlage für globale Software erstellt.

Unicode umfasst die beiden folgenden Schemata für Codeumsetzung. Das Standardschema für Codeumsetzung ist UTF-16, ein 16-Bit-Codierformat. UCS-2 ist eine Untergruppe von UTF-16 und verwendet für die Darstellung der einzelnen Zeichen jeweils 2 Byte. UCS-2 ist allgemein als Universal-Codepage anerkannt, die alle erforderlichen Zeichen aller vorhandenen SBCS- und DBCS-Codepages darstellen kann. UCS-2 ist bei IBM als Codepage 1200 registriert.

Das zweite Unicode-Codierformat ist UTF-8. Dieses Format ist byteorientiert und wurde für den benutzerfreundlichen Einsatz mit vorhandenen ASCII-basierten Systemen konzipiert. UTF-8 verwendet eine unterschiedliche Anzahl Bytes (idR. 1-3, manchmal 4) zum Speichern der einzelnen Zeichen. Die unveränderlichen ASCII-Zeichen werden als einzelne Bytes gespeichert. Alle anderen Zeichen werden mit mehreren Bytes gespeichert. Im Allgemeinen können UTF-Daten von Code, der nicht für Mehrbytecodepages konzipiert wurde, wie erweiterte ASCII-Daten behandelt werden. UTF-8 ist bei IBM als Codepage 1208 registriert.

Bei der Konvertierung von Daten zwischen der lokalen Codepage und den Codepages UCS-2 und UTF-8 müssen Anwendungen die Anforderungen der Daten berücksichtigen. In UCS-2 beispielsweise sind für 20 Zeichen genau 40 Bytes erforderlich und in UTF-8 abhängig von der Originalcodepage und den verwendeten Zeichen 20-60 Bytes.

DB2-Unicode-Datenbanken und -Anwendungen

Eine Datenbank unter DB2 Universal Database für UNIX, Windows oder OS/2, die mit dem codierten Zeichensatz UTF-8 erstellt wurde, kann Daten sowohl im UCS-2-Format als auch im UTF-8-Format speichern. Eine solche Datenbank wird als Unicode-Datenbank bezeichnet. SQL CHAR-Daten werden mit UTF-8 verschlüsselt, SQL GRAPHIC mit UCS-2. Dies entspricht dem Speichern von codierten Einzelbytezeichensätzen in CHAR-Spalten und dem Speichern von codierten Doppelbytezeichensätzen in GRAPHIC-Spalten.

Die Codepage einer Anwendung stimmt möglicherweise nicht mit der Codepage überein, in der DB2 Daten speichert. Stimmen die Codepages bei einer Nicht-Unicode-Datenbank nicht überein, konvertiert der Datenbankmanager Zeichen- und Grafikdaten (reine DBCS-Daten), die zwischen dem Client und dem Server übertragen werden. In einer Unicode-Datenbank konvertiert der Datenbankmanager die Zeichendaten automatisch zwischen der Client-Codepage und UTF-8, aber alle Grafikdaten (UCS-2) werden ohne Konvertierung zwischen dem Client und dem Server übertragen.

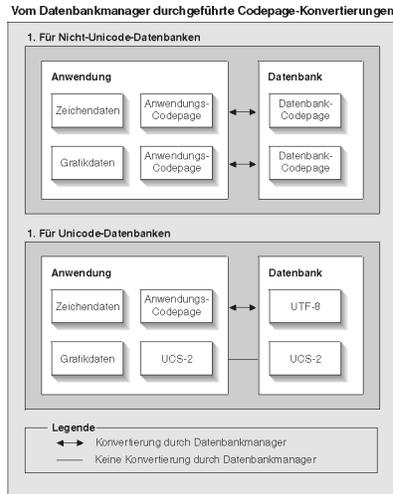


Abbildung 7. Vom Datenbankmanager durchgeführte Codepage-Konvertierungen

Anmerkungen:

1. Gibt eine Anwendung beim Herstellen einer Verbindung zu einer Unicode-Datenbank `DB2CODEPAGE=1208` an, wird als lokale Codepage UTF-8 verwendet, d. h. es ist keine Codepage-Konvertierung erforderlich.
2. Bei Verbindungen zu einer Unicode-Datenbank können Anwendungen der Befehlszeilenschnittstelle Zeichendaten als Grafikdaten empfangen und umgekehrt.

Eine Anwendung kann als Codepage UTF-8 angeben, d. h. diese Anwendung sendet und empfängt Grafikdaten im UCS-2-Format und Zeichendaten im UTF-8-Format. Diese Anwendungs-Codepage wird nur für Unicode-Datenbanken unterstützt.

Außerdem müssen die folgenden Punkte bei der Verwendung von Unicode berücksichtigt werden:

1. Die Datenbank-Codepage wird zum Zeitpunkt der Datenbankeinstellung festgelegt und standardmäßig aus den Ländereinstellungen (bzw. der Codepage) des Betriebssystems abgeleitet. Die Schlüsselwörter `CODESET` und `TERRITORY` können verwendet werden, um explizit eine Unicode-DB2-Datenbank zu erstellen. Beispiel:


```
CREATE DATABASE unidb USING CODESET UTF-8 TERRITORY US
```
2. Standardmäßig wird als Anwendungs-Codepage ebenfalls die lokale Codepage verwendet; diese Einstellung kann aber mit zwei Methoden in UTF-8 geändert werden:
 - Setzen der Anwendungs-Codepage auf UTF-8 (1208) mit dem folgenden Befehl:


```
db2set DB2CODEPAGE=1208
```
 - Für CLI/ODBC-Anwendungen durch Aufrufen von `SQLSetConnectAttr()` und Angeben von `SQL_AA_FALSE` für `SQL_ATTR_ANSI_APP`. Die Standardeinstellung ist `SQL_AA_TRUE`.

3. Bei Daten in GRAPHIC-Spalten wird jedes Unicode-Zeichen durch genau zwei Bytes dargestellt, während für Daten in CHAR-Spalten für jedes Unicode-Zeichen 1-3 Bytes benötigt werden. Bei SQL dürfen für GRAPHIC-Spalten idR. halb so viele Zeichen wie in CHAR-Spalten verwendet werden, aber in Bezug auf Bytes werden beide Spaltentypen gleich behandelt. Die maximale Zeichenlänge für eine CHAR-Spalte ist 254. Die maximale Zeichenlänge für eine Spalte mit Grafikdaten ist 127. Weitere Informationen finden Sie unter MAX im Kapitel „Functions“ des Handbuchs *SQL Reference*.
4. Die Differenzierung zwischen Grafikliterals und Zeichenliterals erfolgt durch die Angabe des Präfix G für Grafikliterals. Beispiel:

```
SELECT * FROM meintab WHERE meinzeich = 'utf-8-daten' AND meingraf = G'ucs-2-daten'
```

Anmerkung: Bei Unicode-Datenbanken ist das Präfix G nicht erforderlich. Weitere Informationen und aktuelle Angaben zur Unterstützung finden Sie in „Literals in Unicode Databases“ auf Seite 479.

5. Die Unterstützung von CLI/ODBC- und JDBC-Anwendungen unterscheidet sich von der Unterstützung für eingebettete Anwendungen. Informationen zur CLI/ODBC-Unterstützung finden Sie in „CLI Guide and Reference“ auf Seite 476.
6. Die Bytereihenfolge von UCS-2-Daten kann je nach Plattform unterschiedlich sein. Intern verwendet DB2 das Big-Endian-Format.

Aktualisierung der Dokumentation

Dieses Dokument aktualisiert die folgenden Informationen zur Verwendung von Unicode mit DB2 Version 7.1:

- *SQL Reference*:
 - Chapter 3 Language Elements
 - Chapter 4 Functions
- *CLI Guide and Reference*:
 - Chapter 3. Using Advanced Features
 - Appendix C. DB2 CLI and ODBC
- *Versetzen von Daten Dienstprogramme und Referenz*, Anhang C, Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD

Weitere Informationen zur Verwendung von Unicode mit DB2 finden Sie in *Systemverwaltung*, Anhang J, unter „Unicode/UCS-2- und UTF-8-Unterstützung in DB2 UDB“.

SQL Reference

Chapter 3 Language Elements

Promotion of Data Types

In this section table 5 shows the precedence list for each data type. Please note:

1. For a Unicode database, the following are considered to be equivalent data types:
 - CHAR and GRAPHIC
 - VARCHAR and VARGRAPHIC
 - LONG VARCHAR and LONG VARGRAPHIC
 - CLOB and DBCLOB

- In a Unicode database, it is possible to create functions where the only difference in the function signature is between equivalent CHAR and GRAPHIC data types, for example, foo(CHAR(8)) and foo(GRAPHIC(8)). We strongly recommend that you do not define such duplicate functions since migration to a future release will require one of them to be dropped before the migration will proceed.

If such duplicate functions do exist, the choice of which one to invoke is determined by a two pass algorithm. The first pass attempts to find a match using the same algorithm as is used for resolving functions in a non-Unicode database. If no match is found, then a second pass will be done taking into account the following promotion precedence for CHAR and GRAPHIC strings:

GRAPHIC->CHAR->VARGRAPHIC->VARCHAR->LONG VARGRAPHIC->LONG VARCHAR->DBCLOB->CLOB

Casting Between Data Types

The following entry has been added to the list introduced as: "The following casts involving distinct types are supported":

- for a Unicode database, cast from a VARCHAR or VARGRAPHIC to distinct type DT with a source data type CHAR or GRAPHIC.

The following are updates to *Table 6. Supported Casts between Built-in Data Types*. Only the affected rows of the table are included.

Table 31. Supported Casts between Built-in Data Types

Target Data Type →	C	V	L	C	G	V	L	D
	H	A	O	L	R	A	O	B
	A	R	N	O	A	R	N	C
	R	C	G	B	P	G	G	L
		H	V		H	R	V	O
		A	A		I	A	A	B
		R	R		C	P	R	
Source Data Type ↓			C			H	G	
			H			I	R	
			A			C	A	
			R				P	
							H	
							I	
							C	
CHAR	Y	Y	Y	Y	Y ¹	Y ¹	-	-
VARCHAR	Y	Y	Y	Y	Y ¹	Y ¹	-	-
LONGVARCHAR	Y	Y	Y	Y	-	-	Y ¹	Y ¹
CLOB	Y	Y	Y	Y	-	-	-	Y ¹
GRAPHIC	Y ¹	Y ¹	-	-	Y	Y	Y	Y
VARGRAPHIC	Y ¹	Y ¹	-	-	Y	Y	Y	Y
LONGVARGRAPHIC	-	-	Y ¹	Y ¹	Y	Y	Y	Y
DBCLOB	-	-	-	Y ¹	Y	Y	Y	Y

¹ Cast is only supported for Unicode databases.

Assignments and Comparisons

Assignments and comparisons involving both character and graphic data are only supported when one of the strings is a literal. For function resolution, graphic literals and character literals will both match character and graphic function parameters.

The following are updates to *Table 7. Data Type Compatibility for Assignments and Comparisons*. Only the affected rows of the table, and the new footnote ⁶, are included:

Operands	Binary Integer	Decimal Number	Floating Point	Character String	Graphic String	DATE	TIME	Time-stamp	Binary String	UDT
Character String	No	No	No	Yes	Yes ⁶	¹	¹	¹	No ³	²
Graphic String	No	No	No	Yes ⁶	Yes	No	No	No	No	²

⁶ Only supported for Unicode databases.

String Assignments:

Storage Assignment

The last paragraph of this sub-section is modified as follows:

When a string is assigned to a fixed-length column and the length of the string is less than the length attribute of the target, the string is padded to the right with the necessary number of single-byte, double-byte, or UCS-2² blanks. The pad character is always a blank even for columns defined with the FOR BIT DATA attribute.

Retrieval Assignment

The third paragraph of this sub-section is modified as follows:

When a character string is assigned to a fixed-length variable and the length of the string is less than the length attribute of the target, the string is padded to the right with the necessary number of single-byte, double-byte, or UCS-2² blanks. The pad character is always a blank even for strings defined with the FOR BIT DATA attribute.

² UCS-2 defines several SPACE characters with different properties. For a Unicode database, the database manager always uses the ASCII SPACE at position x'0020' as UCS-2 blank. For an EUC database, the IDEOGRAPHIC SPACE at position x'3000' is used for padding GRAPHIC strings.

Conversion Rules for String Assignments

The following paragraph has been added to the end of this sub-section:

For Unicode databases, character strings can be assigned to a graphic column, and graphic strings can be assigned to a character column.

DBCS Considerations for Graphic String Assignments

The first paragraph of this sub-section has been modified as follows:

Graphic string assignments are processed in a manner analogous to that for character strings. For non-Unicode databases, graphic string data types are compatible only with other graphic string data types, and never with numeric, character string, or datetime data types. For Unicode databases, graphic string data types are compatible with character string data types.

String Comparisons:

Conversion Rules for Comparison

This sub-section has been modified as follows:

When two strings are compared, one of the strings is first converted, if necessary, to the encoding scheme and/or code page of the other string. For details, see the "Rules for String Conversions" section of Chapter 3 Language Elements in the *SQL Reference*.

Rules for Result Data Types

Character and Graphic Strings in a Unicode Database: This is a new sub-section inserted after the sub-section "Graphic Strings".

In a Unicode database, character strings and graphic strings are compatible.

If one operand is...	And the other operand is...	The data type of the result is...
GRAPHIC(x)	CHAR(y) or GRAPHIC(y)	GRAPHIC(z) where $z = \max(x,y)$
VARGRAPHIC(x)	CHAR(y) or VARCHAR(y)	VARGRAPHIC(z) where $z = \max(x,y)$
VARCHAR(x)	GRAPHIC(y) or VARGRAPHIC	VARGRAPHIC(z) where $z = \max(x,y)$
LONG VARGRAPHIC	CHAR(y) or VARCHAR(y) or LONG VARCHAR	LONG VARGRAPHIC
LONG VARCHAR	GRAPHIC(y) or VARGRAPHIC(y)	LONG VARGRAPHIC
DBCLOB(x)	CHAR(y) or VARCHAR(y) or CLOB(y)	DBCLOB(z) where $z = \max(x,y)$
DBCLOB(x)	LONG VARCHAR	DBCLOB(z) where $z = \max(x,16350)$
CLOB(x)	GRAPHIC(y) or VARGRAPHIC(y)	DBCLOB(z) where $z = \max(x,y)$
CLOB(x)	LONG VARGRAPHIC	DBCLOB(z) where $z = \max(x,16350)$

Rules for String Conversions

The third point has been added to the following list in this section:

For each pair of code pages, the result is determined by the sequential application of the following rules:

- If the code pages are equal, the result is that code page.
- If either code page is BIT DATA (code page 0), the result code page is BIT DATA.
- In a Unicode database, if one code page denotes data in an encoding scheme different from the other code page, the result is UCS-2 over UTF-8 (that is, the graphic data type over the character data type).¹
- Otherwise, the result code page is determined by Table 8 of the “Rules for String Conversions” section of Chapter 3 Language Elements in the *SQL Reference*. An entry of ‘first’ in the table means the code page from the first operand is selected and an entry of ‘second’ means the code page from the second operand is selected.

¹ In a non-Unicode database, conversion between different encoding schemes is not supported.

Expressions

The following has been added:

In a Unicode database, an expression that accepts a character or graphic string will accept any string types for which conversion is supported.

With the Concatenation Operator: The following has been added to the end of this sub-section:

In a Unicode database, concatenation involving both character string operands and graphic string operands will first convert the character operands to graphic operands. Note that in a non-Unicode database, concatenation cannot involve both character and graphic operands.

Predicates

The following entry has been added to the list introduced by the sentence: “The following rules apply to all types of predicates”:

- In a Unicode database, all predicates that accept a character or graphic string will accept any string types for which conversion is supported.

Chapter 4 Functions

Scalar Functions

The following sentence has been added to the end of this section:

In a Unicode database, all scalar functions that accept a character or graphic string will accept any string types for which conversion is supported.

Chapter 3. Using Advanced Features

The following is a new section for this chapter.

Writing a DB2 CLI Unicode Application

There are two main areas of support for DB2 CLI Unicode Applications:

1. The addition of a set of functions that can accept Unicode string arguments in place of ANSI string arguments.
2. The addition of new C and SQL data types to describe data as ANSI or Unicode data. The following sections provide more information for both of these areas. To be considered a Unicode application, the application must set the `SQL_ATTR_ANSI_APP` connection attribute to `SQL_AA_FALSE`, before a connection is made. This will ensure that CLI will connect as a Unicode client, and all Unicode data will be sent in either UTF-8 for CHAR data or UCS-2 for GRAPHIC data.

Unicode Functions: The following is a list of the ODBC API functions that support both Unicode (W) and ANSI (A) versions (the function name will have a W for Unicode):

SQLBrowseConnect	SQLForeignKeys	SQLPrimaryKeys
SQLColAttribute	SQLGetConnectAttr	SQLProcedureColumns
SQLColAttributes	SQLGetConnectOption	SQLProcedures
SQLColumnPrivileges	SQLGetCursorName	SQLSetConnectAttr
SQLColumns	SQLGetDescField	SQLSetConnectOption
SQLConnect	SQLGetDescRec	SQLSetCursorName
SQLDataSources	SQLGetDiagField	SQLSetDescField
SQLDescribeCol	SQLGetDiagRec	SQLSetStmtAttr
SQLDriverConnect	SQLGetInfo	SQLSpecialColumns
SQLDrivers	SQLGetStmtAttr	SQLStatistics
SQLError	SQLNativeSQL	SQLTablePrivileges
SQLExecDirect	SQLPrepare	SQLTables

Unicode functions that always return, or take, string length arguments are passed as count-of-characters. For functions that return length information for server data, the display size and precision are described in number of characters. When the length (transfer size of the data) could refer to string or nonstring data, the length is described in octet lengths. For example, `SQLGetInfoW` will still take the length as count-of-bytes, but `SQLExecDirectW` will use count-of-characters. CLI will return result sets in either Unicode or ANSI, depending on the application's binding. If an application binds to `SQL_C_CHAR`, the driver will convert `SQL_WCHAR` data to `SQL_CHAR`. The driver manager maps `SQL_C_WCHAR` to `SQL_C_CHAR` for ANSI drivers but does no mapping for Unicode drivers.

New datatypes and valid conversions: There are two new CLI or ODBC defined data types, SQL_C_WCHAR and SQL_WCHAR. SQL_C_WCHAR indicates that the C buffer contains UCS-2 data. SQL_WCHAR indicates that a particular column or parameter marker contains Unicode data. For DB2 Unicode Servers, graphic columns will be described as SQL_WCHAR. Conversion will be allowed between SQL_C_WCHAR and SQL_CHAR, SQL_VARCHAR, SQL_LONGVARCHAR and SQL_CLOB, as well as between the graphic data types.

Tabelle 32. Supported Data Conversions

SQL Data Type	SQL_C_CHAR	SQL_C_WCHAR	SQL_C_LONG	SQL_C_SHORT	SQL_C_TINYINT	SQL_C_FLOAT	SQL_C_DOUBLE	SQL_C_TYPE_DATE	SQL_C_TYPE_TIME	SQL_C_TYPE_TIMESTAMP	SQL_C_BINARY	SQL_C_DBCHAR	SQL_C_CLOB	SQL_C_BLOB	SQL_C_DBCLOB	SQL_C_INTEGER	SQL_C_NUMERIC	
BLOB	X	X									D				X			
CHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
CLOB	D	X									X		X					
DATE	X	X						D		X								
DBCLOB		X									X		D			X		
DECIMAL	D	X	X	X	X	X	X				X	X					X	X
DOUBLE	X	X	X	X	X	X	D					X					X	X
FLOAT	X	X	X	X	X	X	D					X					X	X
GRAPHIC (Non-Unicode)	X	X											D					
GRAPHIC (Unicode)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	
INTEGER	X	X	D	X	X	X	X					X					X	X
LONG VARCHAR	D	X									X							
LONG VARGRAPHIC (Non-Unicode)	X	X									X		D					
LONG VARGRAPHIC (Unicode)	X	X									X		D					
NUMERIC	D	X	X	X	X	X	X					X						X
REAL	X	X	X	X	X	D	X					X						X

Tabelle 32. Supported Data Conversions (Forts.)

SQL Data Type	SQL_C_CHAR	SQL_C_WCHAR	SQL_C_LONG	SQL_C_SHORT	SQL_C_TINYINT	SQL_C_FLOAT	SQL_C_DOUBLE	SQL_C_TYPE_DATE	SQL_C_TYPE_TIME	SQL_C_TYPE_TIMESTAMP	SQL_C_BINARY	SQL_C_BIT	SQL_C_DBCHAR	SQL_C_CLOB_LOCCATOR	SQL_C_BLOB_LOCCATOR	SQL_C_DBCLOB_LOCCATOR	SQL_C_BIGINT	SQL_C_NUMERIC
SMALLINT	X	X	X	D	X	X	X					X					X	X
BIGINT	X	X	X	X	X	X	X				X	X					D	X
TIME	X	X							D	X								
TIMESTAMP	X	X						X	X	D								
VARCHAR	D	X	X	X	X	X	X	X	X	X	X	X					X	X
VARGRAPHIC (Non-Unicode)	X	X											D					
VARGRAPHIC (Unicode)	X	X	X	X	X	X	X	X	X	X	X	X	D				X	

Anmerkung:

- D** Conversion is supported. This is the default conversion for the SQL data type.
- X** All IBM DBMSs support the conversion.
- blank** No IBM DBMS supports the conversion.
- Data is not converted to LOB Locator types, rather locators represent a data value, refer to Using Large Objects for more information.
- SQL_C_NUMERIC is only available on 32-bit Windows operating systems.

Obsolete Keyword/Patch Value: Before Unicode applications were supported, applications that were written to work with single byte character data could be made to work with double byte graphic data by a series of `cli.ini` file keywords, such as `GRAPHIC=1,2` or `3`, `Patch2=7` etc. These workarounds presented graphic data as character data, and also affected the reported length of the data.

These keywords are no longer required for Unicode applications, and in fact should not be used otherwise there could be serious side effects. If it is not known if a particular application is a Unicode application, we suggest you try without any of the keywords that affect the handling of graphic data.

Literals in Unicode Databases: In Non-Unicode databases, data in `LONG VARCHAR` and `LONG GRAPHIC` columns cannot be compared. Data in `GRAPHIC/VARGRAPHIC` and `CHAR/VARCHAR` columns can only be compared, or assigned to each other, using explicit cast functions since no implicit code page conversion is supported. This includes `GRAPHIC/VARGRAPHIC` and `CHAR/VARCHAR` literals where a `GRAPHIC/VARGRAPHIC` literal is differentiated from a `CHAR/VARCHAR` literal by a `G` prefix.

For Unicode databases, casting between `GRAPHIC/VARGRAPHIC` and `CHAR/VARCHAR` literals is not required. Also, a `G` prefix is not required in front of a `GRAPHIC/VARGRAPHIC` literal. Provided at least one of the arguments is a literal, implicit conversions occur. This allows literals with or without the `G` prefix to be used within statements that use either `SQLPrepareW()` or `SQLExecDirect().`Literals for `LONG VARGRAPHIC`s still must have a `G` prefix.

For more information, please see "Casting Between Data Types" in Chapter 3 Language Elements of the *SQL Reference*.

New CLI Configuration Keywords: The following three keywords have been added to avoid any extra overhead when Unicode applications connect to a database.

1. **DisableUnicode**

Keyword Description:

Disables the underlying support for Unicode.

db2cli.ini Keyword Syntax:

`DisableUnicode = 0 | 1`

Default Setting:

0 (false)

DB2 CLI/ODBC Settings Tab:

This keyword cannot be set using the CLI/ODBC Settings notebook. The `db2cli.ini` file must be modified directly to make use of this keyword.

Usage Notes:

With Unicode support enabled, and when called by a Unicode application, CLI will attempt to connect to the database using the best client codepage possible to ensure there is no unnecessary data loss due to codepage conversion. This may increase the connection time as codepages are exchanged, or may cause codepage conversions on the client that did not occur before this support was added.

Setting this keyword to True will cause all Unicode data to be converted to the application's local codepage first, before the data is sent to the server. This can cause data loss for any data that cannot be represented in the local codepage.

2. **ConnectCodepage**

Keyword Description:

Specifies a specific codepage to use when connecting to the data source to avoid extra connection overhead.

db2cli.ini Keyword Syntax:

ConnectCodepage = 0 | 1 <any valid db2 codepage>

Default Setting:

0

DB2 CLI/ODBC Settings Tab:

This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

Non-Unicode applications always connect to the database using the application's local codepage, or the DB2Codepage environment setting. By default, CLI will ensure that Unicode applications will connect to Unicode databases using UTF-8 and UCS-2 codepages, and will connect to non-Unicode databases using the database's codepage. This ensures there is no unnecessary data loss due to codepage conversion. This keyword allows the user to specify the database's codepage when connecting to a non-Unicode database in order to avoid any extra overhead on the connection. Specify a value of 1 to cause SQLDriverConnect() to return the correct value in the output connection string, so the value can be used on future SQLDriverConnect() calls.

3. **UnicodeServer**

Keyword Description:

Indicates that the data source is a Unicode Server. Equivalent to setting ConnectCodepage=1208.

db2cli.ini Keyword Syntax:

UnicodeServer = 0 | 1

Default Setting:

0

DB2 CLI/ODBC Settings Tab:

This keyword cannot be set using the CLI/ODBC Settings notebook. The db2cli.ini file must be modified directly to make use of this keyword.

Usage Notes:

This keyword is equivalent to ConnectCodepage=1208, and is added only for convenience. Set this keyword to avoid extra connect overhead when connecting to DB2 for OS/390 Version 7 or higher. There is no need to set this keyword for DB2 for Windows, DB2 for Unix or DB2 for OS/2 databases, since there is no extra processing required.

Appendix C. DB2 CLI and ODBC

The following is a new section added to this appendix.

ODBC Unicode Applications

A Unicode ODBC application sends and retrieves character data primarily in UCS-2. It does this by calling Unicode versions of the ODBC functions ('W' suffix) and by indicating Unicode data types. The application does not explicitly specify a local code page. The application can still call the ANSI functions and pass local code page strings.

For example, the application may call `SQLConnectW()` and pass the DSN, User ID and Password as Unicode arguments. It may then call `SQLExecDirectW()` and pass in a Unicode SQL statement string, and then bind a combination of ANSI local code page buffers (`SQL_C_CHAR`) and Unicode buffers (`SQL_C_WCHAR`). The database data types may be local code page or UCS-2 and UTF-8.

If a CLI application calls `SQLConnectW` or calls `SQLSetConnectAttr` with `SQL_ATTR_ANSI_APP` set to `SQL_AA_FALSE`, the application is considered a Unicode application. This means all CHAR data is sent and received from the database in UTF-8 format. The application can then fetch CHAR data into `SQL_C_CHAR` buffers in local code page (with possible data loss), or into `SQL_C_WCHAR` buffers in UCS-2 without any data loss.

If the application does not do either of the two calls above, CHAR data is converted to the applications local codepage at the server. This means CHAR data fetched into `SQL_C_WCHAR` may suffer data loss.

If the `DB2CODEPAGE` instance variable is set (using `db2set`) to code page 1208 (UTF-8), the application will receive all CHAR data in UTF-8 since this is now the local code page. The application must also ensure that all CHAR input data is also in UTF-8. ODBC also assumes that all `SQL_C_WCHAR` data is in the native endian format. CLI will perform any required byte-reversal for `SQL_C_WCHAR`.

ODBC Unicode Versus Non-Unicode Applications: This release of DB2 Universal Database contains the `SQLConnectW()` API. A Unicode driver must export `SQLConnectW` in order to be recognized as a Unicode driver by the driver manager. It is important to note that many ODBC applications (such as Microsoft Access and Visual Basic) call `SQLConnectW()`. In previous releases of DB2 Universal Database, DB2 CLI has not supported this API, and thus was not recognized as a Unicode driver by the ODBC driver manager. This caused the ODBC driver manager to convert all Unicode data to the application's local code page. With the added support of the `SQLConnectW()` function, these applications will now connect as Unicode applications and DB2 CLI will take care of all required data conversion.

DB2 CLI now accepts Unicode APIs (with a suffix of "W") and regular ANSI APIs. ODBC defines a set of functions with a suffix of "A", but the driver manager does not pass ANSI functions with the "A" suffix to the driver. Instead, it converts these functions to ANSI function calls without the suffix, and then passes them to the driver.

An ODBC application that calls the SQLConnectW() API is considered a Unicode application. Since the ODBC driver manager will always call the SQLConnectW() API regardless of what version the application called, ODBC introduced the SQL_ATTR_ANSI_APP connect attribute to notify the driver if the application should be considered an ANSI or UNICODE application. If SQL_ATTR_ANSI_APP is not set to SQL_AA_FALSE, DB2 CLI converts all Unicode data to the local code page before sending it to the server.

Versetzen von Daten Dienstprogramme und Referenz

Anhang C, Dateiformate der Dienstprogramme EXPORT/IMPORT/LOAD

Dieser Abschnitt wurde wie folgt aktualisiert:

Die Dienstprogramme EXPORT, IMPORT und LOAD werden nicht unterstützt, wenn Sie mit einem Unicode-Client eingesetzt werden, der mit einer Nicht-Unicode-Datenbank verbunden ist. Dateien von Unicode-Clients werden nur unterstützt, wenn der Unicode-Client mit einer Unicode-Datenbank verbunden ist.

Teil 7. Verbindung zu Host-Systemen

Konnektivität: Ergänzung	485
Konfigurieren des Anwendungs-Servers in einer VM-Umgebung.	485
Einstellungen von PATCH1 und PATCH2 bei der CLI/ODBC/JDBC-Konfiguration	485

Konnektivität: Ergänzung

Konfigurieren des Anwendungs-Servers in einer VM-Umgebung

Fügen Sie nach dem ersten (und einzigen) Satz im Unterabschnitt "Definieren des Anwendungs-Servers" des Abschnitts "Bereitstellen von Netzwerkinformationen" folgenden Satz hinzu:

Der RDB_NAME-Wert wird in der Prozedur SQLSTART EXEC als Parameter DBNAME bereitgestellt.

Einstellungen von PATCH1 und PATCH2 bei der CLI/ODBC/JDBC-Konfiguration

Der CLI/ODBC/JDBC-Treiber kann über "Client-Konfiguration - Unterstützung" oder über den ODBC-Treiber-Manager (wenn dieser auf dem System installiert ist) konfiguriert werden bzw. durch manuelles Editieren der Datei `db2cli.ini`. Weitere Informationen finden Sie in *Installation und Konfiguration Ergänzung* oder *CLI Guide and Reference*.

Das Standardverhalten des DB2 CLI/ODBC-Treibers kann geändert werden, indem Sie Werte für die beiden Schlüsselwörter PATCH1 und PATCH2 angeben. Die Eingabe dieser Werte kann über die Datei `db2cli.ini` oder die CLI-API `SQLDriverConnect()` bzw. `SQLBrowseConnect()` erfolgen.

Das Schlüsselwort PATCH1 wird angegeben, indem alle Schlüsselwerte zusammengezählt werden, die der Benutzer festlegen möchte. Wenn z. B. die Programmkorrekturen (Patches) 1, 2 und 8 angegeben würden, hätte PATCH1 den Wert 11. Nachfolgend finden Sie eine Beschreibung der einzelnen Schlüsselwortwerte und ihrer Auswirkung auf den Treiber:

- 1 - Veranlasst den Treiber, nach "count(exp)" zu suchen und durch "count(distinct exp)" zu ersetzen. Das ist erforderlich, da einige DB2-Versionen die Syntax "count(exp)" unterstützen, die auch von einigen ODBC-Anwendungen generiert wird. Für Microsoft-Anwendungen erforderlich, wenn der Server die Syntax "count(exp)" nicht unterstützt.
- 2 - Einige ODBC-Anwendungen werden abgefangen, wenn `SQL_NULL_DATA` in der Funktion `SQLGetTypeInfo()` für die Spalte `LITERAL_PREFIX` oder `LITERAL_SUFFIX` zurückgegeben wird. Zwingt den Treiber, statt dessen eine leere Zeichenfolge zurückzugeben. Erforderlich für Impromptu 2.0.
- 4 - Zwingt den Treiber, die Daten für die Eingabezeitmarke wie Datumsdaten zu behandeln, wenn Zeit- und Bruchkomponente der Zeitmarke den Wert Null haben. Erforderlich für Microsoft Access.
- 8 - Zwingt den Treiber, die Daten für die Eingabezeitmarke wie Zeitdaten zu behandeln, wenn die Datumskomponente der Zeitmarke den Wert 1899-12-30 hat. Erforderlich für Microsoft Access.
- 16 - Nicht verwendet.
- 32 - Zwingt den Treiber, keine Informationen zu Spalten des Typs `SQL_LONGVARCHAR`, `SQL_LONGVARBINARY` und `SQL_LONGVARGRAPHIC` zurückzugeben. Wirkt auf die Anwendung, als würden Langfelder

- nicht unterstützt. Erforderlich für Lotus 1-2-3.
- 64 - Zwingt den Treiber, Grafikausgabezeichenfolgen mit NULL zu beenden. Erforderlich für Microsoft Access in einer Doppelbyteumgebung.
 - 128 - Zwingt den Treiber, die Abfrage "SELECT Config, nValue FROM MSysConf" an den Server durchzulassen. Derzeit gibt der Treiber einen Fehler mit SQLSTATE-Wert S0002 (Tabelle nicht gefunden) zurück. Erforderlich, wenn der Benutzer diese Konfigurationstabelle in der Datenbank erstellt hat und die Anwendung darauf zugreifen soll.
 - 256 - Zwingt den Treiber, die Primärschlüsselspalten als erstes beim Aufruf SQLStatistics() zurückzugeben. Derzeit gibt der Treiber die Indizes nach Indexname sortiert zurück. Dies ist das ODBC-Standardverhalten.
 - 512 - Zwingt den Treiber, bei SQLGetFunctions() für SQL_API_SQLTABLEPRIVILEGES für SQL_API_SQLTABLEPRIVILEGES und SQL_API_SQLCOLUMNPRIVILEGES den Wert FALSE zurückzugeben.
 - 1024 - Zwingt den Treiber, SQL_SUCCESS statt SQL_NO_DATA_FOUND in SQLExecute() oder SQLExecDirect() zurückzugeben, wenn die ausgeführte Anweisung UPDATE oder DELETE keine Zeilen betrifft. Erforderlich für Visual Basic-Anwendungen.
 - 2048 - Nicht verwendet.
 - 4096 - Zwingt den Treiber, nach dem Schließen eines Cursors keine COMMIT-Anweisung abzusetzen, wenn er sich im Modus für automatisches Festschreiben befindet.
 - 8192 - Zwingt den Treiber, nach dem Aufrufen einer gespeicherten Prozedur eine zusätzliche Ergebnismenge zurückzugeben. Diese besteht aus einer einzeiligen Ergebnismenge mit den Ausgabewerten der gespeicherten Prozedur. Für den Zugriff durch Powerbuild-Anwendungen.
 - 32768 - Zwingt den Treiber, die Verarbeitung von Microsoft Query-Anwendungen mit DB2 MVS-Synonymen zu ermöglichen.
 - 65536 - Zwingt den Treiber, manuell ein "G" vor Zeichenliterals zu setzen, bei denen es sich eigentlich um Grafikliterals handelt. Diese Programmkorrektur (Patch) muss immer bei der Arbeit in einer Doppelbyteumgebung angegeben werden.
 - 131072 - Zwingt den Treiber, eine Zeitmarkenspalte als CHAR(26)-Spalte zu beschreiben, wenn diese Teil eines eindeutigen Index ist. Erforderlich für Microsoft-Anwendungen.
 - 262144 - Zwingt den Treiber, die Pseudo-Katalogtabelle db2cli.procedures statt der Tabellen SYSCAT.PROCEDURES und SYSCAT.PROCPARMS zu verwenden.
 - 524288 - Zwingt den Treiber, SYSTEM_TABLE_SCHEMA statt TABLE_SCHEMA bei der Bearbeitung einer Systemtabellenabfrage eines Systems mit DB2/400 Version 3.x zu verwenden. Führt zu einer Leistungsverbesserung.
 - 1048576 - Zwingt den Treiber, eine Zeichenfolge der Länge Null in SQLPutData() wie SQL_NULL_DATA zu behandeln.

Das Schlüsselwort PATCH2 unterscheidet sich von PATCH1. Bei diesem Schlüsselwort werden mehrere Programmkorrekturen (Patches) durch Kommas getrennt angegeben. Wenn z. B. die Programmkorrekturen 1, 4 und 5 angegeben wären, hätte PATCH2 den Wert "1,4,5". Nachfolgend finden Sie eine Beschreibung der einzelnen Schlüsselwortwerte und ihrer Auswirkung auf den Treiber:

- 1 - Zwingt den Treiber, den Namen einer gespeicherten Prozedur in einer Anweisung CALL in Großschreibung umzusetzen.
- 2 - Nicht verwendet.
- 3 - Zwingt den Treiber, alle Argumente in Schemenaufrufen in Großbuchstaben umzusetzen.
- 4 - Zwingt den Treiber, für Schemenaufrufe (d. h. SQLColumns(), SQLProcedureColumns() usw.), die LIKE-Ergebnismenge der Version 2.1.2 statt einer LIKE-Ergebnismenge der Version 5 zurückzugeben.
- 5 - Zwingt den Treiber, die Verarbeitung von VARCHAR-Eingabespalten, bei denen die Zeiger auf Daten und Länge im Speicher aufeinander folgen, nicht zu optimieren.
- 6 - Zwingt den Treiber, eine Nachricht zurückzugeben, dass verschiebbare Cursor nicht unterstützt werden. Erforderlich für Visual Basic-Programme, wenn der DB2-Client Version 5 und der Server DB2 UDB Version 5 ist.
- 7 - Zwingt den Treiber, alle GRAPHIC-Spaltendatentypen dem Spaltendatentyp CHAR zuzuordnen. Erforderlich für eine Doppelbyteumgebung.
- 8 - Zwingt den Treiber, Katalogsuchargumente in Schemenaufrufen zu ignorieren.
- 9 - Keine COMMIT-Operation bei frühzeitigem Schließen des Cursors (Early Close).
- 10 - Nicht verwendet.
- 11 - Ausgeben, dass der Katalogname unterstützt wird (gespeicherte VB-Prozeduren).
- 12 - Entfernt doppelte Anführungszeichen aus Argumenten eines Schemenaufrufs (Visual Interdev).
- 13 - Kein Anhängen von Schlüsselwörtern aus db2cli.ini an Ausgabeverbindungszeichenfolgen.
- 14 - Ignoriert den Schemennamen bei SQLProcedures() und SQLProcedureColumns().
- 15 - Verwendet den Punkt als Dezimaltrennzeichen bei der Zeichenausgabe.
- 16 - Rückgabe von beschreibenden Informationen bei jedem OPEN erzwingen.
- 17 - Keine Rückgabe von Spaltennamen bei DESCRIBE.
- 18 - Versuch, Literale durch Parametermarken zu ersetzen.

- 19 - Derzeit unterstützt DB2 MVS Version 4.1 nicht die ODBC-Syntax, in der runde Klammern in der Klausel ON einer erweiterten Verknüpfungsklausel zulässig sind. Wenn Sie PATCH2 aktivieren, entfernt der IBM DB2 ODBC-Treiber die runde Klammer, wenn sich die erweiterte Verknüpfungsklausel in einer ODBC-Escape-Zeichenfolge befindet. PATCH2 sollte nur für DB2 MVS 4.1 verwendet werden.
- 20 - Derzeit unterstützt DB2 unter MVS nicht das Vergleichselement BETWEEN mit Parametermarken als zwei Operanden (Ausdruck ? BETWEEN ?). Wenn Sie diese Programmkorrektur aktivieren, schreibt der IBM ODBC-Treiber das Vergleichselement um (Ausdruck >= ? und Ausdruck <= ?).
- 21 - Setzt alle Parameter OUTPUT für gespeicherte Prozeduren auf SQL_NULL_DATA.
- 22 - Veranlasst den IBM ODBC-Treiber, die erweiterte Verknüpfung als nicht unterstützt zu melden. Für Anwendungen, die SELECT DISTINCT coll oder ORDER BY coll generieren, wenn eine erweiterte Verknüpfungsanweisung verwendet wird, wobei coll länger ist als 254 Zeichen, so dass DB2 UDB einen Fehler zurückgibt (DB2 UDB unterstützt bei dieser Syntax keine Spalten, die größer sind als 254 Byte).
- 23 - Keine Eingabeoptimierung von Parametern, die mit cbColDef=0 gebunden wurden.
- 24 - Problemumgehung bei Access für die Zuordnung von Zeitwerten als Zeichen.
- 25 - Problemumgehung bei Access für Dezimalspalten - entfernt abschließende Nullen bei der Zeichendarstellung.
- 26 - Keine Rückgabe des SQLCODE-Werts 464 an die Anwendung - gibt an, dass Ergebnismengen zurückgegeben werden.
- 27 - Zwingt SQLTables, den Wert des Schlüsselworts TABLETYPE zu verwenden, selbst wenn die Anwendung einen gültigen Wert angibt.
- 28 - Beschreibt reale Spalten als Doppelspalten.
- 29 - Problemumgehung bei ADO für Dezimalspalten - entfernt führende Nullen für x-Werte; dabei gilt: 1 > x > -1 (nur für einige MDAC-Versionen erforderlich).
- 30 - Inaktiviert die Cache-Optimierung der gespeicherten Prozedur.
- 31 - Meldet Statistikdaten für Aliasnamen bei einem Aufruf von SQLStatistics.
- 32 - Überschreibt die Verarbeitung für SQLCODE-Wert -727, Ursachencode 4.
- 33 - Gibt nach dem Konvertieren der Zeitmarke in den CHAR-Datentyp die ISO-Version zurück (nicht die ODBC-Version).
- 34 - Meldet CHAR FOR BIT DATA-Spalten als CHAR-Spalten.
- 35 - Meldet einen ungültigen Tabellennamen, wenn SQL_DESC_BASE_TABLE_NAME angefordert wird - Optimierung bei ADO für den Lesezugriff.
- 36 - Reserviert.
- 37 - Reserviert.

Teil 8. Allgemeine Informationen

Allgemeine Informationen	491
DB2 Universal Database Informationsmanagement Kurzüberblick	491
DB2 Everywhere jetzt DB2 Everyplace	492
Maus erforderlich	492
Fehler "Bindedateien nicht gefunden" beim Ver- such, von DB2 Run-Time Client aus zu binden	492
Discovery-Modus SEARCH.	492
Speicherfenster für HP-UX 11	492
Benutzeraktion für "dlfm client_conf" schlägt fehl	493
Wenn der Kopierdämon bei "dlfm stop" nicht stop- pen sollte.	494
DB2 DFS Client Enabler deinstallieren	494
Client-Authentifizierung unter Windows NT	494
AutoLoader kann während einer Aufspaltung blo- ckieren	494
DATALINK-Wiederherstellung	495
Definieren von Benutzer-ID und Kennwort in IBM Communications Server für Windows NT (CS/NT). 495	
Knotendefinition	495
Einschränkungen bei Systemen zusammengeschos- sener Datenbanken	496
Einschränkung bei DataJoiner	496
Hebrew Information Catalog Manager for Win- dows NT.	496
Unterstützung für Microsoft SNA Server und SNA- Aktualisierung an mehreren Standorten (zweipha- sige Festschreibung)	497
DB2 SNA SPM kann nach dem Booten von Win- dows nicht gestartet werden	497
Festlegen der Ländereinstellung für den DB2- Verwaltungsserver.	498
Nicht funktionierende Tastenkombinationen	498
Anforderungen an Servicebenutzereinträge für DB2 unter Windows NT und Windows 2000	498
Verlorenes Zugriffsrecht EXECUTE für in Version 6 erstellte Benutzer von Query Patroller	499
Query Patroller - Einschränkungen	499
Notwendigkeit zum Festschreiben aller benutzer- definierten Programme, die in der Data Warehouse-Zentrale verwendet werden.	499
Neue Option für Export über die Befehlszeile der Data Warehouse-Zentrale	500
Sicherungsservice-APIs (XBSA)	500
OS/390-Agent	501
Installationsübersicht	501
Installationsdetails.	502
Konfigurieren zusätzlicher Agentenfunktionen	505
Planen von Warehouse-Schritten mit dem Auslöserprogramm (XTClient).	507
Umsetzungsprogramme	508
Zugreifen auf Datenbanken außerhalb der DB2- Systemfamilie	511
Ausführen von Dienstprogrammen von DB2 für OS/390	513
Replikation	515
Agentenprotokollierung	516
Client-seitiges Caching in Windows NT	517
Testprodukte auf den CD-ROMs von Enterprise Edition für UNIX	517
Testprodukte auf den CD-ROMs von DB2 Connect Enterprise Edition für UNIX	518
Löschen von Data Links Manager	519
Deinstallation von DLFM-Komponenten mit SMIT entfernt unter Umständen zusätzliche Datei- gruppen	519
Fehler SQL1035N bei der Verwendung des Befehls- zeilenprozessors unter Windows 2000	519
Verbesserung bei SQL Assist	519
DB2 für Linux - Integration der Gnome- und KDE- Desktops	520
Ausführen von DB2 unter einem Windows 2000- Terminalserver (im Verwaltungsmodus)	521
Onlinehilfefunktion für Befehle zum Sichern und Wiederherstellen	521
"Warehouse Manager" muss "DB2 Warehouse Manager" heißen	521

Allgemeine Informationen

DB2 Universal Database Informationsmanagement Kurzüberblick

Der Kurzüberblick ist unter DB2 für Linux oder Linux/390 nicht verfügbar.

Der Kurzüberblick ist für die Ausführung mit kleinen Systemschriftarten optimiert. Sie müssen möglicherweise die Schriftgröße Ihres Web-Browsers anpassen, damit Sie den Kurzüberblick unter OS/2 korrekt anzeigen können. Informationen zur Anpassung der Schriftgröße finden Sie in der Hilfe Ihres Web-Browsers. Zur korrekten Anzeige des Kurzüberblicks (nur Einzelbytezeichensätze) empfiehlt es sich, eine Helvetica-Schrift mit 8 Punkt zu verwenden. Japanische und koreanische Kunden sollten eine Mincho-Schrift mit 8 Punkt verwenden. Sie müssen die Option für die Verwendung der Standardschriftarten anstelle der dokumentspezifischen Schriftarten in den Einstellungen auswählen.

Es kann vorkommen, dass der Kurzüberblick hinter einem zweiten Browser-Fenster gestartet wird. Schließen Sie den Kurzüberblick, und führen Sie die Schritte in „Fehlernachrichten beim Versuch, Netscape zu starten“ auf Seite 12, aus, um dieses Problem zu beheben.

Beim Starten des Kurzüberblicks erhalten Sie möglicherweise einen JavaScript-Fehler ähnlich dem Folgenden:

```
file:/C:/Program Files/SQLLIB/doc/html/db2qt/index4e.htm, line 65:
```

```
Window is not defined.
```

Dieser JavaScript-Fehler verhindert, dass die Startseite des Kurzüberblicks, index4e.htm, automatisch geschlossen wird, nachdem der Kurzüberblick gestartet wurde. Sie können die Startseite des Kurzüberblicks schließen, indem Sie das Browser-Fenster schließen, in dem index4e.htm angezeigt wird.

Im Abschnitt "Neue Funktionen" zum Thema Datenverwaltung wird angegeben, dass die Unterstützung der Protokollarchive nach Bedarf in Version 7.1 unterstützt wird. Dies ist nicht der Fall. Außerdem heißt es:

Die Größe von Protokolldateien wurde von 4 GB auf 32 GB erhöht.

Dieser Satz muss wie folgt lauten:

Der insgesamt aktive Protokollbereich wurde von 4 GB auf 32 GB erhöht.

Der Abschnitt zu DB2 Data Links Manager enthält folgenden Satz:

Außerdem wird nun die Verwendung der Veritas-Schnittstelle XBSA für die Sicherung und Wiederherstellung mit NetBackup unterstützt.

Dieser Satz muss wie folgt lauten:

Außerdem wird nun die XBSA-Schnittstelle für die Dateiarchivierung und -wiederherstellung unterstützt. Speichermanager, die die XBSA-Schnittstelle unterstützen, sind z. B. Legato NetWorker und Veritas NetBackup.

DB2 Everywhere jetzt DB2 Everyplace

Der Name von DB2 Everywhere wurde in DB2 Everyplace geändert.

Maus erforderlich

Bei allen Plattformen mit Ausnahme von Windows ist eine Maus zur Verwendung der Tools erforderlich.

Fehler "Bindedateien nicht gefunden" beim Versuch, von DB2 Run-Time Client aus zu binden

Da DB2 Run-Time Client keinen vollständigen Satz von Bindedateien hat, kann das Binden von GUI-Tools nicht von DB2 Run-Time Client aus erfolgen, sondern nur von DB2 Administration Client.

Discovery-Modus SEARCH

Der Discovery-Modus SEARCH wird nur für Datenträger im Broadcast-Betrieb unterstützt. Der Discovery-Modus SEARCH funktioniert z. B. nicht über einen ATM-Adapter. Diese Einschränkung wirkt sich jedoch nicht auf den Discovery-Modus KNOWN aus.

Speicherfenster für HP-UX 11

Speicherfenster sind für Benutzer konzipiert, die mit großen HP-64-Bit-Maschinen arbeiten und mehr als 1,75 GB gemeinsam benutzten Speicher für 32-Bit-Anwendungen nutzen wollen. Speicherfenster sind nicht erforderlich, wenn Sie die 64-Bit-Version von DB2 ausführen. Speicherfenster stellen einen separaten gemeinsam benutzten Speicher in der Größe von 1 GB pro Prozess oder Prozessgruppe zur Verfügung. Dadurch kann ein Exemplar seinen eigenen 1 GB großen gemeinsam benutzten Speicher plus 0,75 GB globalen gemeinsam benutzten Speicher verwenden. Wenn Benutzer dies nutzen wollen, können sie mehrere Exemplare jeweils in einem eigenen Fenster ausführen. Es gibt folgende Vorbedingungen und Voraussetzungen für die Verwendung von Speicherfenstern:

- DB2 EE-Umgebung
 - Programmkorrekturen: Erweiterungssoftware 12/98 und PHKL_17795
 - Die Variable \$DB2INSTANCE muss für das Exemplar gesetzt sein.
 - Die Datei /etc/services.window muss für jedes DB2-Exemplar, das Sie unter Speicherfenstern ausführen wollen, einen Eintrag enthalten. Beispiel:

```
db2instance1 50
db2instance2 60
```

Anmerkung: Zwischen dem Namen und der ID darf nur ein Leerzeichen stehen.

- Alle DB2-Befehle, die Sie auf dem Server ausführen wollen und die mehrere Anweisungen benötigen, müssen mit einer TCP/IP-Schleifenmethode ausgeführt werden. Dies ist notwendig, weil die Shell beendet wird, wenn Speicherfenster die Verarbeitung der ersten Anweisung beenden. Der DB2-Service kann angeben, wie dies erreicht werden kann.
- Allen DB2-Befehlen, die Sie für ein Exemplar in Speicherfenstern ausführen wollen, muss „db2win“ (in sqllib/bin) vorangestellt werden. Beispiel:

```
db2win db2start
db2win db2stop
```

- Alle DB2-Befehle, die außerhalb von Speicherfenstern ausgeführt werden (wenn Speicherfenster aktiv sind), sollten die Nachricht SQL1042 ausgeben.
Beispiel:

```
db2win db2start <== OK
db2 connect to db <==SQL1042
db2stop <==SQL1042
db2win db2stop <== OK
```

- DB2 EEE-Umgebung

- Programmkorrekturen: Erweiterungssoftware 12/98 und PHKL_17795
- Die Variable \$DB2INSTANCE muss für das Exemplar gesetzt sein.
- Die Registrierungsvariable DB2_ENABLE_MEM_WINDOWS muss auf TRUE gesetzt sein.
- Die Datei /etc/services.window muss für jeden logischen Knoten jedes Exemplars, das Sie unter Speicherfenstern ausführen wollen, einen Eintrag enthalten. Das erste Feld jedes Eintrags sollte der mit der Anschlussnummer verknüpfte Exemplarname sein. Beispiel:

```
=== $HOME/sql1lib/db2nodes.cfg für db2instance1 ===
5 host1 0
7 host1 1
9 host2 0
```

```
=== $HOME/sql1lib/db2nodes.cfg für db2instance2 ===
1 host1 0
2 host2 0
3 host2 1
```

```
=== /etc/services.window auf host1 ===
db2instance10 50
db2instance11 55
db2instance20 60
```

```
=== /etc/services.window auf host2 ===
db2instance10 30
db2instance20 32
db2instance21 34
```

- Sie dürfen einem Befehl nicht „db2win“ voranstellen. „db2win“ darf nur in einer EE-Umgebung verwendet werden.

Benutzeraktion für "dlfm client_conf" schlägt fehl

Wenn bei einem DLFM-Client der Befehl **dlfm client_conf** aus irgendeinem Grund fehlschlägt, können „alte“ Einträge in DB2-Katalogen die Ursache hierfür sein. Dieses Problem kann mit folgenden Befehlen behoben werden:

```
db2 uncatalog db <datenbankname>
db2 uncatalog node <aliasname-des-knotens>
db2 terminate
```

Führen Sie dann **dlfm client_conf** erneut aus.

Wenn der Kopierdämon bei "dlfm stop" nicht stoppen sollte

In sehr seltenen Fällen stoppt der Kopierdämon, **dlfm_copyd**, nicht, wenn ein Benutzer den Befehl **dlfm stop** absetzt oder wenn das System abnormal heruntergefahren wird. Wenn dieser Fehler auftritt, setzen Sie den Befehl **dlfm shutdown** ab, bevor Sie dlfm neu starten.

DB2 DFS Client Enabler deinstallieren

Bevor DB2 DFS Client Enabler deinstalliert werden kann, sollte der Root-Benutzer sicherstellen, dass keine DFS-Dateien mehr in Benutzung sind und dass kein Benutzer mehr eine Shell im DFS-Dateibereich offen hat. Setzen Sie als Root-Benutzer den folgenden Befehl ab:

```
stop.dfs dfs_cl
```

Prüfen Sie, dass /... nicht mehr angehängt ist:

```
mount | grep -i dfs
```

Wenn dies nicht erfolgt ist und DB2 DFS Client Enabler deinstalliert wird, muss die Maschine erneut gestartet werden.

Client-Authentifizierung unter Windows NT

Es gibt eine neue DB2-Registrierungsvariable, DB2DOMAINLIST, die den bestehenden Client-Authentifizierungsmechanismus in der Windows NT-Umgebung ergänzt. Diese Variable wird auf dem Server unter DB2 für Windows NT verwendet, um mindestens eine Windows NT-Domäne zu definieren. Nur Verbindungs- oder Anschlussanforderungen von Benutzern, die zu den in dieser Liste definierten Domänen gehören, werden akzeptiert.

Diese Registrierungsvariable sollte nur in einer reinen Windows NT-Domänenumgebung mit DB2-Servern und -Clients mit Version 7 (oder höher-) verwendet werden.

Informationen zur Einstellung dieser Registrierungsvariablen finden Sie im Abschnitt "DB2-Registrierungsvariablen und DB2-Umgebungsvariablen" des Handbuchs *Systemverwaltung: Optimierung*.

AutoLoader kann während einer Aufspaltung blockieren

AIX 4.3.3 enthält eine Programmkorrektur für ein libc-Problem, das AutoLoader bei einer Aufspaltung zum Blockieren bringen könnte. AutoLoader ist ein Multi-Thread-Programm. Einer der Threads spaltet mit fork einen anderen Prozess ab. Durch das Abspalten eines Kindprozesses wird ein Image des Elterprozess-Speichers im Kindprozess erstellt.

Sperren, die von libc.a für die Verwaltung mehrerer Threads verwendet werden, die wiederum Speicher aus dem Freispeicher zuordnen, können im selben Prozess gleichzeitig von einem nicht aufspaltenden Thread definiert sein. Da der nicht aufspaltende Thread im Kindprozess nicht vorhanden ist, wird diese Sperre im Kindprozess nie freigegeben, so dass der Elterprozess manchmal blockiert wird.

DATALINK-Wiederherstellung

Bei der Wiederherstellung jeglicher Offlinesicherungen, die nach einer Datenbankwiederherstellung mit oder ohne aktualisierender Wiederherstellung durchgeführt wurden, erfolgt keine schnelle Abstimmung. In solchen Fällen werden alle Tabellen mit DATALINK-Spalten unter Dateiverbindungssteuerung in den Status DRP (DataLink Reconcile Pending) gesetzt.

Definieren von Benutzer-ID und Kennwort in IBM Communications Server für Windows NT (CS/NT)

Wenn Sie APPC als Kommunikationsprotokoll für ferne DB2-Clients verwenden, um eine Verbindung zu Ihrem DB2-Server herzustellen, und wenn Sie CS/NT als SNA-Produkt verwenden, stellen Sie sicher, dass die folgenden Schlüsselwörter in der CS/NT-Konfigurationsdatei korrekt eingestellt ist. Diese Datei befindet sich im allgemeinen im Verzeichnis `x:\ibmcs\private`.

Knotendefinition

TG_SECURITY_BEHAVIOR

Mit diesem Parameter gibt der Benutzer an, wie der Knoten Sicherheitsinformationen bei der Verbindungsherstellung (ATTACH) handhaben soll, wenn das Transaktionsprogramm nicht für Sicherheit konfiguriert ist.

IGNORE_IF_NOT_DEFINED

Mit diesem Parameter gibt der Benutzer an, ob Sicherheitsparameter bei der Verbindungsherstellung (ATTACH) vorhanden sind und ob sie ignoriert werden sollen, wenn das Transaktionsprogramm nicht für Sicherheit konfiguriert ist.

Wenn Sie IGNORE_IF_NOT_DEFINED verwenden, müssen Sie in CS/NT keine Benutzer-ID und kein Kennwort definieren.

VERIFY_EVEN_IF_NOT_DEFINED

Mit diesem Parameter gibt der Benutzer an, ob Sicherheitsparameter bei der Verbindungsherstellung (ATTACH) vorhanden sind. Außerdem wird festgelegt, dass diese Parameter überprüft werden, selbst wenn das Transaktionsprogramm nicht für Sicherheit konfiguriert ist. Dies ist die Standardeinstellung.

Wenn Sie VERIFY_EVEN_IF_NOT_DEFINED verwenden, müssen Sie in CS/NT eine Benutzer-ID und ein Kennwort definieren.

Führen Sie die folgenden Schritte aus, um die Benutzer-ID und das Kennwort für CS/NT zu definieren:

1. Wählen Sie **Start** —> **Programme** —> **IBM Communications Server** —> **SNA Node Configuration** aus. Das Fenster **Welcome to Communications Server Configuration** wird geöffnet.
2. Wählen Sie die gewünschte Konfigurationsdatei aus. Klicken Sie **Next** an. Das Fenster **Choose a Configuration Scenario** wird geöffnet.
3. Markieren Sie **CPI-C-, APPC- or 5250-Emulation**. Klicken Sie **Finish** an. Das Fenster **Communications Server SNA Node** wird geöffnet.

4. Klicken Sie [+] neben **CPI-C** und **APPC** an.
5. Klicken Sie [+] neben **LU6.2 Security** an.
6. Klicken Sie **User Passwords** mit Maustaste 2 an, und wählen Sie **Create** aus. Das Fenster **Define a User ID Password** wird geöffnet.
7. Geben Sie die Benutzer-ID und das Kennwort ein. Klicken Sie **OK** an. Klicken Sie **Finish** an, damit die Änderungen wirksam werden.

Einschränkungen bei Systemen zusammenschlossener Datenbanken

Bei Systemen zusammenschlossener Datenbanken gelten folgende Einschränkungen:

- Die Oracle-Datentypen NCHAR, NVARCHAR2, NCLOB und BFILE werden in Abfragen mit Kurznamen nicht unterstützt.
- Die Befehle CREATE SERVER OPTION, ALTER SERVER OPTION und DROP SERVER OPTION werden von der Steuerzentrale nicht unterstützt. Zur Ausgabe eines dieser Befehle müssen Sie den Befehlszeilenprozessor verwenden.
- Bei Abfragen mit Kurznamen hält DB2 UDB sich nicht immer an die Datenbank-Konfigurationsoption DFT_SQLMATHWARN. Stattdessen gibt DB2 UDB die arithmetischen Fehler oder Warnungen unabhängig von der Einstellung von DFT_SQLMATHWARN direkt von der fernen Datenquelle aus.
- Bei der Anweisung CREATE SERVER OPTION darf die Server-Option COLSEQ für Datenquellen mit einer Sortierfolge, bei der Groß-/Kleinschreibung nicht unterschieden wird, nicht auf 'Y' gesetzt werden.
- Die Anweisung ALTER NICKNAME gibt SQL0901N zurück, wenn eine ungültige Option angegeben wird.
- Bei Oracle-, Microsoft SQL Server- und Sybase-Datenquellen können numerische Datentypen nicht dem DB2-Datentyp BIGINT zugeordnet werden. Standardmäßig wird der Oracle-Datentyp **number(p,s)** dem DB2-Datentyp DECIMAL zugeordnet. Dabei gilt: $10 \leq p \leq 18$ und $s = 0$.

Einschränkung bei DataJoiner

Verteilte Anforderungen, die in einer zusammenschlossenen Umgebung abgesetzt werden, sind auf Nur-Lese-Operationen beschränkt.

Hebrew Information Catalog Manager for Windows NT

The Information Catalog Manager component is available in Hebrew and is provided on the DB2 Warehouse Manager for Windows NT CD.

The Hebrew translation is provided in a zip file called IL_ICM.ZIP and is located in the DB2\IL directory on the DB2 Warehouse Manager for Windows NT CD.

To install the Hebrew translation of Information Catalog Manager, first install the English version of DB2 Warehouse Manager for Windows NT and all prerequisites on a Hebrew Enabled version of Windows NT.

After DB2 Warehouse Manager for Windows NT has been installed, unzip the IL_ICM.ZIP file from the DB2\IL directory into the same directory where DB2 Warehouse Manager for Windows NT was installed. Ensure that the correct options are supplied to the unzip program to create the directory structure in the zip file.

After the file has been unzipped, the global environment variable LC_ALL must be changed from En_US to Iw_IL. To change the setting:

1. Open the Windows NT Control Panel and double click on the **System** icon.
2. In the **System Properties** window, click on the **Environment** tab, then locate the variable LC_ALL in the **System Variables** section.
3. Click on the variable to display the value in the **Value** edit box. Change the value from **En_US** to **Iw_IL**.
4. Click on the **Set** button.
5. Close the **System Properties** window and the **Control Panel**.

The Hebrew version of Information Catalog Manager should now be installed.

Unterstützung für Microsoft SNA Server und SNA-Aktualisierung an mehreren Standorten (zweiphasige Festschreibung)

Host- und AS/400-Anwendungen können nicht mit zweiphasiger SNA-Festschreibung auf DB2 UDB-Server zugreifen, wenn das verwendete SNA-Produkt Microsoft SNA Server ist. DB2 UDB-Veröffentlichungen, in denen dies angegeben wird, sind nicht korrekt. IBM Communications Server für Windows NT Version 5.02 oder höher ist erforderlich.

Anmerkung: Anwendungen, die auf Host- und AS/400-Datenbank-Server mit DB2 UDB für Windows zugreifen, *können* zweiphasige SNA-Festschreibung in Verbindung mit Microsoft SNA Server Version 4 Service Pack 3 oder höher verwenden.

DB2 SNA SPM kann nach dem Booten von Windows nicht gestartet werden

Wenn Sie Microsoft SNA Server Version 4 SP3 oder später verwenden, stellen Sie bitte sicher, dass DB2 SNA SPM nach einem Warmstart korrekt gestartet wird. Prüfen Sie die Datei \sqlllib\<exemplarname>\db2diag.log auf Einträge ähnlich den Folgenden:

```
2000-04-20-13.18.19.958000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:316 Appid:none
common_communication sqlccspmconnmgr_APPC_init Probe:19
SPM0453C Der Synchronisationspunktmanager wurde nicht gestartet, da
Microsoft SNA Server nicht gestartet wurde.
2000-04-20-13.18.23.033000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:302 Appid:none
common_communication sqlccsna_start_listen Probe:14
DIA3001E "SNA SPM"-Protokollunterstützung nicht erfolgreich gestartet.
```

```
2000-04-20-13.18.23.603000 Instance:DB2 Node:000
PID:291(db2syscs.exe) TID:316 Appid:none
common_communication sqlccspmconnmgr_listener Probe:6
DIA3103E Fehler in APPC-Protokollunterstützung festgestellt. APPC-Verb
"APPC(DISPLAY 1 BYTE)". Primärer RC: "F004". Sekundärer RC: "00000000".
```

Sind solche Einträge in Ihrer db2diag.log vorhanden und stimmen die Zeitmarken mit der Zeit des letzten Warmstarts überein, gehen Sie wie folgt vor:

1. Rufen Sie db2stop auf.
2. Starten Sie den SnaServer-Service (falls dieser noch nicht gestartet wurde).
3. Rufen Sie db2start auf.

Überprüfen Sie die Datei db2diag.log erneut, um sicherzustellen, dass die Einträge nicht mehr hinzugefügt werden.

Festlegen der Ländereinstellung für den DB2-Verwaltungsserver

Stellen Sie sicher, dass die Ländereinstellung des DB2-Verwaltungsserver-Exemplars kompatibel mit der Ländereinstellung des DB2-Exemplars ist. Andernfalls kann das DB2-Exemplar nicht mit dem DB2-Verwaltungsserver kommunizieren.

Wenn die Umgebungsvariable LANG im Benutzerprofil des DB2-Verwaltungsservers nicht gesetzt ist, wird der DB2-Verwaltungsserver mit den Standardeinstellungen für die Ländereinstellung des Systems gestartet. Wenn die Standardeinstellungen für die Ländereinstellung des Systems nicht definiert sind, wird der DB2-Verwaltungsserver mit Codepage 819 gestartet. Wenn das DB2-Exemplar eine DBCS-Ländereinstellung verwendet und der DB2-Verwaltungsserver mit Codepage 819 gestartet wird, kann das Exemplar nicht mit dem DB2-Verwaltungsserver kommunizieren. Die Ländereinstellung des DB2-Verwaltungsservers und die des DB2-Exemplars müssen kompatibel sein. Auf einem Linux-System für vereinfachtes Chinesisch z. B. sollte LANG=zh_CN im Benutzerprofil des DB2-Verwaltungsservers gesetzt sein.

Nicht funktionierende Tastenkombinationen

In einigen Sprachen funktionieren einige Tastenkombinationen für die Steuerzentrale auf UNIX-basierten Systemen und unter OS/2 nicht. Verwenden Sie bitte die Maus zur Auswahl von Optionen.

Anforderungen an Servicebenutzereinträge für DB2 unter Windows NT und Windows 2000

Während der Installation von DB2 für Windows NT oder Windows 2000 erstellt das Installationsprogramm mehrere Windows-Services und ordnet jedem Service einen Benutzereintrag zu. Das Installationsprogramm erteilt dem Servicebenutzereintrag, der dem jeweiligen DB2-Service zugeordnet wird, die folgenden Benutzerrechte, damit DB2 ordnungsgemäß ausgeführt werden kann:

- Als Teil des Betriebssystems handeln
- Erstellen eines Tokenobjekts
- Anheben einer Quote
- Anmelden als Dienst
- Ersetzen eines Tokens auf Prozessebene

Wenn Sie einen anderen Servicebenutzereintrag für die DB2-Services verwenden möchten, müssen Sie diesem Servicebenutzereintrag die o. g. Benutzerrechte erteilen.

Außerdem muss der Servicebenutzereintrag über Schreibzugriff für das Verzeichnis verfügen, in dem das DB2-Produkt installiert ist.

Der DB2-Service DB2DAS00 (Verwaltungsserver) muss außerdem über die Berechtigung zum Starten und Stoppen anderer DB2-Services verfügen (d. h. der Service muss zur Gruppe der Hauptbenutzer gehören) und die DB2-Berechtigung SYS-ADM für alle DB2-Exemplare besitzen, die er verwaltet.

Verlorenes Zugriffsrecht EXECUTE für in Version 6 erstellte Benutzer von Query Patroller

Aufgrund einiger neuer gespeicherter Prozeduren (IWM.DQPGROUP, IWM.DQP-VALUR, IWM.DQPCALCT und IWM.DQPINJOB), die Query Patroller Version 7 hinzugefügt wurden, verlieren Benutzer, die mit Query Patroller Version 6 erstellt wurden, das Zugriffsrecht EXECUTE für diese Pakete. FixPak 1 enthält eine Anwendung, die dieses Problem automatisch löst.

Wenn Sie Benutzerinformationen mit DQP Query Admin modifizieren, versuchen Sie bitte nicht, vorhandene Benutzer aus der Liste zu entfernen.

Query Patroller - Einschränkungen

Aufgrund von Einschränkungen der JVM-Plattform (Java Virtual Machine) wird Query Enabler unter HP-UX und NUMA-Q nicht unterstützt. Außerdem wird Query Patroller Tracker unter NUMA-Q nicht unterstützt. Sind alle Query Patroller-Client-Tools erforderlich, wird die Verwendung einer anderen Plattform (z. B. Windows NT) für die Ausführung dieser Tools auf einem HP-UX- oder NUMA-Q-Server empfohlen.

Notwendigkeit zum Festschreiben aller benutzerdefinierten Programme, die in der Data Warehouse-Zentrale verwendet werden

Wenn Sie eine gespeicherte Prozedur verwenden wollen, die von DB2 Stored Procedure Builder als benutzerdefiniertes Programm in der Data Warehouse-Zentrale erstellt wurde, müssen Sie die folgende Anweisung in die gespeicherte Prozedur vor der Anweisung `con.close();` einfügen:

```
con.commit();
```

Wird diese Anweisung nicht eingefügt, werden Änderungen an der gespeicherten Prozedur zurückgesetzt, sobald die gespeicherte Prozedur von der Data Warehouse-Zentrale aus ausgeführt wird.

Bei allen benutzerdefinierten Programmen in der Data Warehouse-Zentrale ist es notwendig, alle enthaltenen DB2-Funktionen *explizit* festzuschreiben, damit die Änderungen in der Datenbank wirksam werden. Das bedeutet, Sie müssen die COMMIT-Anweisungen den benutzerdefinierten Programmen hinzufügen.

Neue Option für Export über die Befehlszeile der Data Warehouse-Zentrale

Der Export über die Befehlszeile in Befehlsdateien hat eine neue Option, /B. Diese Option ist über die Schnittstelle der Data Warehouse-Zentrale nicht verfügbar.

Die neue Syntax für den Befehl **iwh2exp2** lautet:

```
iwh2exp2 dateiname.INP dbname benutzer-id kennwort [PREFIX=tabellenschema]
[/S] [/R] [/B]
```

Dabei gilt Folgendes:

- *dateiname.INP* ist der vollständige Pfadname der INP-Datei.
- *dbname* ist der Name der Steuerungsdatenbank der Data Warehouse-Zentrale.
- *benutzer-id* ist die Benutzer-ID, die zur Anmeldung an die Datenbank verwendet wird.
- *kennwort* ist das Kennwort, das zur Anmeldung an die Datenbank verwendet wird.
- Optionale Parameter:
 - PREFIX=*tabellenschema* ist das Tabellenschema für die Steuerungsdatenbanktabellen (der Standardwert ist IWH).
 - /S exportiert Zeitpläne mit ausgewählten Schritten.
 - /R exportiert Warehouse-Quellen mit ausgewählten Schritten nicht.
 - /B exportiert mitwirkenden Schritte mit ausgewählten Schritten nicht.

Anmerkung: Wenn /R oder /B angegeben wird, müssen die Warehouse-Quellen oder mitwirkenden Schritte bereits vorhanden sein, wenn die resultierende Befehlsdatei importiert wird. Ansonsten wird ein Fehler zurückgegeben.

Sicherungsservice-APIs (XBSA)

Sicherungsservice-APIs (XBSA) wurden von der Open Group in Großbritannien als offene Anwendungsprogrammierschnittstelle zwischen Anwendungen und Funktionen definiert, die Datenspeicherverwaltung für Sicherungs- oder Archivierungszwecke benötigen. Dies wird in "Open Group Technical Standard System Management: Backup Services API (XBSA)", Dokumentnummer C425 (ISBN: 1-85912-056-3) dokumentiert.

Zur Unterstützung dieser API wurden zwei neue DB2-Registrierungsvariablen erstellt, die derzeit unter AIX, HP, Solaris und Windows NT unterstützt werden:

DB2_VENDOR_INI

Zeigt auf eine Datei mit allen herstellereigenen Umgebungseinstellungen. Der Wert wird ausgewählt, wenn der Datenbankmanager startet.

DB2_XBSA_LIBRARY

Zeigt auf die vom Hersteller bereitgestellte XBSA-Bibliothek. Unter AIX muss die Einstellung das gemeinsam benutzte Objekt umfassen, wenn dieses nicht `shr.o` heißt. HP, Solaris und Windows NT benötigen den Namen des gemeinsam benutzten Objekts nicht. Zur Verwendung von Legatos NetWorker Business Suite Module für DB2 z. B. muss die Registrierungsvariable wie folgt eingestellt werden:

```
db2set DB2_XBSA_LIBRARY="/usr/lib/libxdb2.a(bsashr10.o)"
```

Die XBSA-Schnittstelle kann über den Befehl `BACKUP DATABASE` oder `RESTORE DATABASE` aufgerufen werden. Beispiel:

```
db2 backup db sample use XBSA
db2 restore db sample use XBSA
```

OS/390-Agent

Inhalt des Dokuments: Installieren von OS/390 und seinen Funktionen

In diesem Dokument finden Sie Anweisungen zur Installation des OS/390-Agenten sowie Informationen zu seinen Merkmalen und Funktionen. Eine Kurzübersicht des Installationsprozesses finden Sie in *Installationsübersicht*, ausführliche Anweisungen sind in *Installationsdetails* zu finden. Informationen zu den Agentenfunktionen sind in den Abschnitten *Konfigurieren zusätzlicher Agentenfunktionen*, *Umsetzungsprogramme* und *Zugreifen auf Datenbanken außerhalb der DB2-Systemfamilie* enthalten.

Übersicht

Die DB2 Warehouse-Zentrale enthält jetzt einen OS/390-Agenten. Mit dem Agenten können Sie zwischen der DB2 Universal Database für OS/390-Datenbank und anderen Datenbanken, einschließlich DB2-Datenbanken auf anderen Plattformen und Fremddatenbanken, Daten übertragen. Der Agent kann mit unterstützten Datenquellen kommunizieren, die eine ODBC-Verbindung verwenden. Der Agent wird unter OS/390 UNIX Systems Services ausgeführt. Er benötigt OS/390 Version 2 Release 6 oder höher und ist kompatibel mit DB2 für OS/390 Version 5, 6 und 7.

Der OS/390-Agent unterstützt die folgenden Tasks:

- Kopieren von Daten aus einer DB2-Quellendatenbank in eine DB2-Zieldatenbank
- Ziehen von Stichprobeninhalten aus einer Tabelle oder Datei
- Ausführen benutzerdefinierter Programme
- Zugriff auf fremde Datenbanken über IBM DataJoiner
- Zugriff auf VSAM oder IMS-Daten über Cross Access Classic Connect
- Ausführen von Dienstprogrammen von DB2 Universal Database für OS/390
- Ausführen des Apply-Jobs für IBM Data Propagator

Installationsübersicht

In den folgenden Schritten ist der Installationsprozess zusammengefasst. Der Abschnitt *Installationsdetails* bietet ausführlichere Informationen zu diesen Schritten.

1. Installieren des OS/390-Agenten vom Band für DB2 Universal Database für OS/390
2. Aktualisieren der Umgebungsvariablen in Ihrer profile-Datei
3. Herstellen von Verbindungen:
 - Zwischen dem Kernel und dem Agentendämon
 - Zwischen dem Agenten und den Datenbanken, auf die er zugreift
4. Lokales Binden von CLI und Binden von CLI an alle fernen Datenbanken
5. Einrichten der ODBC-Initialisierungsdatei
6. Einrichten von Berechtigungen, so dass der Benutzer:
 - den Agentendämon ausführen kann
 - die Ausführungsberechtigung für den Zugriffsplan DSNAOCLI hat
 - Lese- und Schreibberechtigung für Protokoll- und ODBC-Trace-Verzeichnisse (falls erforderlich) hat
7. Starten des Agentendämons

Installationsdetails

Installieren des OS/390-Agenten

Der OS/390-Agent ist auf dem Band für DB2 Universal Database für OS/390 Version 7 vorhanden. Weitere Informationen zur Installation des OS/390-Agenten finden Sie im Programmverzeichnis des Bandes.

Sie müssen APAR PQ36585 oder PQ36586 auf Ihr DB2-Subsystem anwenden, bevor Sie den OS/390-Agenten installieren.

Aktualisieren der Umgebungsvariablen in Ihrer profile-Datei

Die Variablen verweisen den Agenten auf verschiedene DB2-Bibliotheken, -Ausgabeverzeichnis usw.

Unten sehen Sie den Inhalt einer .profile-Beispieldatei. Die .profile-Datei definiert die Umgebungsvariablen und muss sich im Ausgangsverzeichnis des Benutzers befinden, der den Agentendämon startet:

```
export VWS_LOGGING=/usr/lpp/DWC/logs
export VWP_LOG=/usr/lpp/DWC/vwp.log
export VWS_TEMPLATES=/usr/lpp/DWC/
export DSNAOINI=/usr/lpp/DWC/dsnaoini
export LIBPATH=/usr/lpp/DWC/:$LIBPATH
export PATH=/usr/lpp/DWC/:$PATH
export STEPLIB=DSN710.SDSNEXIT:DSN710.SDSNLOAD
```

Herstellen von Verbindungen

Fügen Sie der Datei /etc/services oder der Datei TCPIP.ETC.SERVICES folgende Zeilen hinzu, um die Kernel- und Dämonverbindungen herzustellen:

```
wkernel 11000/tcp vwd 11001/tcpvlogger 11002/tcp
```

Fügen Sie Ihrer OS/390-Kommunikationsdatenbank ferne Datenbanken hinzu, um Verbindungen zwischen dem OS/390-Agenten und Datenbanken herzustellen. Im Folgenden sehen Sie einige Beispieleinfügungen für die Kommunikationsdatenbank:

```
INSERT INTO SYSIBM.LOCATIONS
  (LOCATION, LINKNAME, PORT)
VALUES
  ('NTDB', 'VWNT704', '60002');

INSERT INTO SYSIBM.IPNAMES
  (LINKNAME, SECURITY_OUT, USERNAMES, IPADDR)
VALUES
  ('VWNT704', 'P', '0', 'VWNT704.STL.IBM.COM');

INSERT INTO SYSIBM.USERNAMES
  (TYPE, AUTHID, LINKNAME, NEWAUTHID, PASSWORD)
VALUES
  ('0', 'MVSUID', 'VWNT704', 'NTUID', 'NTPW');
```

Weitere Informationen zum Herstellen von Verbindungen und Aktualisieren Ihrer Kommunikationsdatenbank finden Sie im Kapitel "Connecting Distributed Database Systems" im Handbuch *DB2 UDB for OS/390 Installation Guide* (GC26-9008-00).

Binden von CLI

Da der OS/390-Agent CLI zur Kommunikation mit DB2 verwendet, müssen Sie Ihren CLI-Zugriffsplan an alle ferne Datenbanken binden, auf die Ihr Agent zugreifen wird. Einige Beispiele für BIND PACKAGE-Anweisungen für eine lokale DB2-Datenbank für OS/390:

```
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIMS)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (DWC6CLI) MEMBER(DSNCLIF4)
```

Einige Beispiele für BIND PACKAGE-Anweisungen für eine DB2-Datenbank unter Windows NT:

```
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLICS) ISO(CS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLINC) ISO(NC)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRR) ISO(RR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIRS) ISO(RS)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIUR) ISO(UR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIC2)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIQR)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIF4)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV1)
BIND PACKAGE (NTDB.DWC6CLI) MEMBER(DSNCLIV2)
```

Eine BIND-Beispielanweisung zum Binden der CLI-Pakete in einem Zugriffsplan:

```
BIND PLAN(DWC6CLI) PKLIST(*.DWC6CLI.* )
```

Definieren der ODBC-Initialisierungsdatei

Im Verzeichnis `usr/lpp/DWC/` befindet sich eine Beispiel-ODBC-Initialisierungsdatei (`inisamp`). Sie können diese Datei an die Anforderungen Ihres Systems anpassen oder eine eigene Datei erstellen. Prüfen Sie, ob die Datei richtig konfiguriert ist, um sicherzustellen, dass sie richtig funktioniert:

- Die Umgebungsvariable `DSNAOINI` muss auf die Initialisierungsdatei zeigen.
- Der Dateiname sollte die Namenskonvention `dsnaoini.positionsname` verwenden.
- Die Datei muss die Parameter `CONNECTTYPE=2` und `MVSATTACHTYPE=CAF` enthalten.

Weitere Informationen zum Binden von CLI und zur `DSNAOINI`-Datei finden Sie im Handbuch *DB2 UDB for OS/390 ODBC Guide and Reference* (SC26-9005).

Einrichten von Berechtigungen

Der OS/390-Agent ist ein Dämonprozess. Sie können den Agentendämon mit normaler UNIX-Sicherheit oder mit OS/390-UNIX-Sicherheit ausführen.

Weil der Agent Dämonenberechtigung erfordert, definieren Sie die folgenden ausführbaren Agentenelemente in der RACF-Programmsteuerung:

- libt1s4d.dll
- iwhcomnt.dll
- vwd

Zum Definieren der ausführbaren Programme in der RACF-Programmsteuerung wechseln Sie in das Verzeichnis, in dem die ausführbaren Dateien der Data Warehouse-Zentrale gespeichert sind, und führen Sie folgende Befehle aus:

```
extattr +p libt1s4d.dll
extattr +p iwhcomnt.dll
extattr +p vwd
```

Damit Sie den Befehl `extattr` mit dem Parameter `+p` verwenden können, müssen Sie mindestens Lesezugriff für die Klasse `BPX.FILEATTR.PROGCTL FACILITY` haben. Das folgende Beispiel zeigt den RACF-Befehl, mit dem der Benutzer-ID `SMORG` diese Berechtigung erteilt wird:

```
RDEFINE FACILITY BPX.FILEATTR.PROGCTL UACC(NONE)
PERMIT BPX.FILEATTR.PROGCTL CLASS(FACILITY) ID(SMORG) ACCESS(READ)
SETROPTS RACLIST(FACILITY) REFRESH
```

Weitere Informationen zu Berechtigungen finden Sie im Handbuch *OS/390 UNIX System Services Planning*, IBM Form SC28-1890.

Starten des Agentendämons

Wenn Sie das System konfiguriert haben, starten Sie den Agentendämon:

1. Stellen Sie eine Telnet-Verbindung zu UNIX Systems Services unter OS/390 über den OS/390-Host-Namen und USS-Anschluss her.
2. Starten Sie den Agentendämon:
 - Geben Sie `vwd` in der Befehlszeile ein, um den Dämon im Vordergrund zu starten.
 - Wenn Sie den Dämon im Hintergrund starten wollen, geben Sie Folgendes ein:

```
vwd>/usr/lpp/DWC/logs/vwd.log 2>&1 &
```

Geben Sie folgenden Befehl in einer UNIX-Shell-Befehlszeile ein, um zu prüfen, ob der OS/390-Agentendämon aktiv ist:

```
ps -e | grep vwd
```

Oder geben Sie `D 0MVS,a=all` an der OS/390-Konsole ein, und suchen Sie nach der Zeichenfolge `vwd`.

Konfigurieren zusätzlicher Agentenfunktionen

Das DB2 Warehouse Manager-Paket enthält die folgenden benutzerdefinierten Programme:

- vwpftp: Führt eine FTP-Befehlsdatei aus.
- vwpmvs: Übergibt einen JCL-Jobstrom.
- vwprcpy: Kopiert eine Datei mit FTP.
- XTClient: Client-Auslöserprogramm.
- etidlmvs: ETI-Dienstprogramm (Evolutionary Technologies International); löscht eine Datei unter MVS.
- etircmvs: ETI-Dienstprogramm ; führt FTP auf einem MVS-Host aus.
- etiexmvs: ETI-Dienstprogramm ; führt JCL auf MVS aus.

Darüber hinaus können Sie benutzerdefinierte Programme und gespeicherte Prozeduren in der Data Warehouse-Zentrale definieren. Der OS/390-Agent unterstützt alle ausführbaren Dateien, die unter UNIX Systems Services ausgeführt werden.

Ein benutzerdefiniertes Programm wird mindestens einem Schritt zugeordnet. Wenn Sie das benutzerdefinierte Programm ausführen, werden folgende Aktionen ausgeführt:

- Der Agent startet.
- Der Agent führt das benutzerdefinierte Programm aus.
- Das benutzerdefinierte Programm liefert einen Rückkehrcode und eine Rückmeldungsdatei an den Agenten.
- Der Agent liefert die Ergebnisse an den Kernel.

Damit ETI-Programme unter OS/390 ausgeführt werden können, müssen Sie zunächst FixPak 2 auf DB2 Universal Database Version 7.1 anwenden.

Verwenden Sie die Umgebungsvariable VWP_LOG, um ein Verzeichnis zu definieren, in das benutzerdefinierte Programme ihre Ausgabe schreiben können.

Wenn Sie einen Job mit FTP unter Verwendung eines benutzerdefinierten Programms übergeben, müssen Sie zunächst die JCL und die Daten, die Sie übergeben wollen, erstellen. Der Jobname in der JCL muss USERIDx lauten; hierbei ist x ein Buchstabe oder eine einstellige Zahl (z. B. MYUSERA). Die Ausgabeklasse für die Dateien MSGCLASS und SYSOUT in Ihrer JCL muss eine JES-gesteuerte Ausgabeklasse sein.

Einschränkung: Der Maximalwert für LRECL für den übergebenen Job ist 254 Zeichen. JES überprüft nur die ersten 72 Zeichen der JCL.

Ändern der Schablone der Data Warehouse-Zentrale für FTP-Unterstützung

Die Data Warehouse-Zentrale installiert eine JCL-Schablone für die Dateiübertragung mit Hilfe von FTP. Wenn der OS/390-Agent die FTP-Befehle GET oder PUT für die Dateiübertragung von einem OS/390-Host an einen anderen fernen Host verwenden soll, müssen Sie die Benutzerinformationen in der JCL-Schablone für Ihr OS/390-System ändern:

1. Melden Sie sich mit einer ID an, die eine Berechtigung zum Kopieren und Aktualisieren von Dateien im Verzeichnis /usr/lpp/DWC hat.
2. Suchen Sie ftp.jcl, und kopieren Sie die Datei mit dem neuen Dateinamen systemname.ftp.jcl. Hierbei ist *systemname* der Name des OS/390-Systems.
3. Erstellen Sie eine Kopie dieser Datei für jedes OS/390-System, auf dem Sie das Konvertierungsprogramm vwpmvms oder ETI extract ausführen wollen. Wenn Sie beispielsweise eines dieser Programme auf STLMVS1 ausführen wollen, erstellen Sie eine Kopie der Datei mit dem Namen STLMVS1.ftp.jcl.
4. Passen Sie die JCL mit Hilfe eines Texteditors den Anforderungen Ihres Standorts an. Passen Sie die Benutzerinformationen den Standardbenutzerinformationen für Ihr MVS-System an. Parameter in eckigen Klammern, z. B. [USERID] und [FTPFILE], dürfen Sie nicht ändern. (Die eckigen Klammern sind die Hexadezimalzeichen x'AD' bzw. x'BD'. Wenn Ihr TSO-Terminaltyp in SPF-Option 0 nicht auf 3278A gesetzt ist, werden diese Werte möglicherweise als Sonderzeichen und nicht als eckige Klammern angezeigt. Das ist kein Problem, solange Sie nicht x'AD' oder x'BD' oder die Daten zwischen diesen Zeichen ändern.)
5. Aktualisieren Sie die Umgebungsvariable VWS_TEMPLATES so, dass sie auf das Verzeichnis der kopierten Schablonendatei zeigt.

Die Data Warehouse-Zentrale wird mit der folgenden JCL-Schablone geliefert:

```

//[USERID]A JOB , 'PUT/GET',
// CLASS=A,
// USER=&SYSUID,
// NOTIFY=&SYSUID,
// TIME=(,30),
// MSGCLASS=H
//STEP1 EXEC PGM=FTP,PARM='( EXIT'
//INPUT DD DSN=[FTPFILE],DISP=SHR
//OUTPUT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*

```

Beispielinhalt einer Tabelle oder Datei

Mit dem OS/390-Agenten können Sie Stichproben des Inhalts von DB2-Tabellen und unstrukturierten Dateien, z. B. UNIX Systems Services-Dateien und nativer unstrukturierter OS/390-Dateien, entnehmen. Sie können auch Stichproben des Inhalts von IMS- oder VSAM-Dateien über Classic Connect mit dem OS/390-Agenten entnehmen. Bei unstrukturierten Dateien bestimmt der Agent das Dateiformat anhand der Parameter in den Merkmalen der Dateidefinition.

Planen von Warehouse-Schritten mit dem Auslöserprogramm (XTClient)

Mit Hilfe des Auslöserprogramms können Sie Warehouse-Schritte über die OS/390-Plattform planen. Sie oder ein OS/390-Job-Scheduler können/kann einen Job übergeben, der einen Schritt in der Data Warehouse-Zentrale auslöst. Ist der Schritt erfolgreich, gibt der Auslöserschritt in der JCL den Rückkehrcode 0 zurück.

Für die Verwendung des Auslöserprogramms müssen Sie Java Development Kit (JDK) 1.1.8 oder höher unter Ihren OS/390 UNIX Systems Services installiert haben. Starten Sie zunächst XTServer auf der Maschine, auf der Ihr Warehouse-Server aktiv ist, um den Auslöser zu starten. Dieser Prozess ist in Kapitel 5 des Handbuchs *Data Warehouse-Zentrale Verwaltung* im Abschnitt zum Starten eines Schritts außerhalb der Data Warehouse-Zentrale beschrieben. Starten Sie nach dem Start von XTServer den XTClient auf OS/390. Im Folgenden sehen Sie eine Beispiel-JCL zum Starten des Auslösers.

```
//DBA1A JOB 1,'XTCLIENT',CLASS=A,MSGCLASS=H,
//      MSGLEVEL=(1,1),REGION=4M,NOTIFY=&SYSUID
//*****
//* submit iwhetrig
//*****
//BRADS EXEC PGM=BXPBATCH,
//      PARM=('sh cd /usr/lpp/DWC/; java XTClient 9.317.171.133 1100x
//      9 drummond pw bvmvs2nt 1 1 100')
//STDOUT DD PATH='/tmp/xtclient.stdout',
//      PATHOPTS=(OWRONLY,OCREAT),
//      PATHMODE=SIRWXU
//STDERR DD PATH='/tmp/xtclient.stderr',
//      PATHOPTS=(OWRONLY,OCREAT),
//      PATHMODE=SIRWXU
//
```

Anmerkung: Der oben aufgeführte JCL-Beispielcode zeigt, wie die Parameter in einer neuen Zeile fortgesetzt werden. Hierfür geben Sie die Parameter bis Spalte 71 ein, geben ein 'X' in Spalte 72 ein und fahren in Spalte 16 auf der nächsten Zeile fort.

Der erste Abschnitt von PARM ist eine Anweisung (cd /usr/lpp/DWC/;), die in das Verzeichnis wechselt, in dem der OS/390-Agent installiert ist. Der zweite Abschnitt von PARM startet XTClient und übergibt die folgenden 8 Parameter:

- Host-Name oder Host-IP-Adresse des Servers der Data Warehouse-Zentrale
- Anschluss des Servers der Data Warehouse-Zentrale (normalerweise 11009)
- Benutzer-ID der Data Warehouse-Zentrale
- Kennwort der Data Warehouse-Zentrale
- Name des auszuführenden Schritts
- Server-Befehl der Data Warehouse-Zentrale. Dabei gilt Folgendes:
 - 1 = Schritt füllen
 - 2 = Schritt in den Testmodus hochstufen
 - 3 = Schritt in den Produktionsmodus hochstufen
 - 4 = Schritt in den Testmodus herabstufen
 - 5 = Schritt in den Entwicklungsmodus herabstufen
- Option zum Warten auf Abschluss des Schritts (1=ja und 0=nein)
- Die maximale Anzahl Zeilen (0 oder Leerzeichen verwenden, um alle Zeilen abzurufen)

Umsetzungsprogramme

Einführung

Die folgenden 12 Umsetzungsprogramme sind gespeicherte Java-Prozeduren, die einige grundlegende Datenumsetzungen bereitstellen. Damit Sie diese Umsetzungsprogramme ausführen können, müssen Sie zuerst die gespeicherten Java-Prozeduren auf Ihrem DB2-Subsystem einrichten. Weitere Informationen zu diesen Umsetzungsprogrammen finden Sie im Handbuch *IBM DB2 Universal Database Data Warehouse-Zentrale Verwaltung Version 7* (SC12-2885-00).

IWH.CLEAN
IWH.PERIODTABLE
IWH.KEYTABLE
IWH.CHISQUARE
IWH.CORRELATION
IWH.STATISTICS
IWH.INVERTDATA
IWH.PIVOTDATA
IWH.REGRESSION
IWH.ANOVA
IWH.SUBTOTAL
IWH.MOVINGAVERAGE

Definieren von gespeicherten Java-Prozeduren

Die folgenden Anweisungen sind eine Kurzform der vollständigen Anweisungen zum Definieren von gespeicherten Java-Prozeduren im Handbuch *Application Programming Guide and Reference for Java™* (SC26-9018).

1. Wenden Sie die vorläufigen Programmkorrekturen UQ46170 und UQ46114 auf Ihr DB2-Subsystem an.
2. Installieren Sie Visual Age für Java 2.0 oder höher auf Ihrem OS/390-System.
3. Installieren Sie JDBC auf Ihrem DB2-System, und binden Sie die JDBC-Pakete in Ihrem DB2-Subsystem.
4. Definieren Sie gespeicherte RRS- und DB2 WLM-Prozeduren für Ihr DB2-Subsystem.
5. Definieren Sie gespeicherte Java-Prozeduren für Ihr DB2-System. Dies umfasst die Erstellung einer Java WLM-Startprozedur für den Adressraum der gespeicherten Java-Prozedur.
6. Unter WLM müssen Sie Ihre Java WLM-Startprozedur einem WLM-Umgebungsnamen zuordnen. Verwenden Sie die Anzeige **Create an Application Environment** der WLM-Anwendungsumgebung, um den Umgebungsnamen der JCL-Prozedur zuzuordnen.
7. Geben Sie den WLM-Anwendungsumgebungsnamen für die Option WLM_ENVIRONMENT in CREATE oder ALTER PROCEDURE an, um eine gespeicherte Prozedur oder benutzerdefinierte Funktion einer Anwendungsumgebung zuzuordnen.
8. Stellen Sie sicher, dass der Eigner Ihrer gestarteten DB2-Tasks Zugriff auf die Bibliotheken in der Java WLM-Startprozedur hat.

Schritte zum Einrichten von Warehouse-Umsetzungsprogrammen

Die folgenden Anweisungen sind eine Kurzform der vollständigen Anweisungen im Handbuch *IBM DB2 Universal Database Data Warehouse-Zentrale Verwaltung Version 7* (SC12-2885-00).

1. Wenden Sie entweder FixPak 3 auf DB2 Universal Database für NT Version 7 an, oder aktualisieren Sie die Warehouse-Steuerungsdatenbank auf TRANSREGISTERED = 1 und TRANSFENCED=1.

Zum Aktualisieren der Warehouse-Steuerungsdatenbank geben Sie die folgende SQL-Anweisung in einem Befehlszeilenprozessor von DB2 Universal Database ein:

```
CONNECT TO ihre_vw_steuerungsdatenbank
UPDATE IWH.INFORESOURCE SET TRANSREGISTERED = '1' WHERE SUBDBTYPE = 'DB2 MVS'
UPDATE IWH.INFORESOURCE SET TRANSFENCED = '1' WHERE SUBDBTYPE = 'DB2 MVS'
```

2. Definieren Sie die Umsetzungsprogramme für DB2.
 - Bei DB2 für OS/390 Version 7 verwenden Sie die SQL-Anweisungen in /usr/lpp/DWC/createXfSQLV7.
 - Bei DB2 für OS/390 Version 6 verwenden Sie die SQL-Anweisungen in /usr/lpp/DWC/createXfSQL.
 - Bei DB2 für OS/390 Version 5 verwenden Sie die kommentierten SQL-Anweisungen in /usr/lpp/DWC/createXfSQL. Setzen Sie alle Anweisungen CREATE PROCEDURE auf Kommentar. Entfernen Sie dann die Kommentarzeichen vor den Anweisungen INSERT INTO SYSIBM.SYSPROCEDURES, und verwenden Sie diese Anweisungen, um die Umsetzungsprogramme für DB2 für OS/390 Version 5 zu verwenden.

Wenn Sie eine gespeicherte Java-Prozedur definieren, verwenden Sie WLM zur Zuordnung der Java WLM-Startprozedur zu einem WLM-Umgebungsname. Der Umgebungsname wird in der Option WLM ENVIRONMENT der Anweisung CREATE PROCEDURE angegeben. DSNWLMJ ist der WLM-Umgebungsname, der in den oben beschriebenen Umsetzungsprogrammdefinitionen enthalten ist. Sie können entweder einen WLM-Zuordnungsname von DSNWLMJ hinzufügen oder die Option WLM ENVIRONMENT für jede Umsetzungsprogrammdefinition in einen Namen ändern, der bereits Ihrer Startprozedur zugeordnet ist.

3. Definieren Sie Verbindungen von UNIX Systems Services zu den Lademodulen des Umsetzungsprogramms in IWH710.SIWHLOAD.
 - Stellen Sie eine Telnet-Verbindung zu UNIX Systems Services auf Ihrem OS/390-Host-System her.
 - Wechseln Sie in das Verzeichnis, in dem der OS/390-Agent installiert ist. Das Standardinstallationsverzeichnis heißt /usr/lpp/DWC.
 - Wenn Sie DB2 Version 7 verwenden, überspringen Sie Schritt 4. Wenn Sie DB2 Version 5 oder 6 verwenden, editieren Sie den Datensatz trlinks im Installationsverzeichnis. Setzen Sie die folgende Zeile auf Kommentar, indem Sie ein Nummernzeichen (#) in Spalte 1 setzen:

```
ln -e IWHXFV7 xf.j11;
```

Entfernen Sie das Kommentarzeichen (#) in Spalte 1 der folgenden Zeile.

```
#ln -e IWHXF xf.j11;
```

Speichern Sie Ihre Änderungen.

- Geben Sie trlinks ein, und drücken Sie die Eingabetaste. Dadurch wird eine Verbindung xf.j11 im Verzeichnis erstellt, die den Agenten anweist, das Modul IWHXF oder IWHXFV7 zu laden.
4. Berechtigungen Sie IWH710.SIWHPDSE mit APF, und fügen Sie es der STEPLIB-Verkettung in der Startprozedur Ihrer gespeicherten DB2 Java-Prozedur hinzu.

5. Fügen Sie das Verzeichnis, in dem sich Ihre Verbindung `xf.j11` befindet (Standard: `/usr/lpp/DWC`), den Umgebungsvariablen `CLASSPATH` und `LIBPATH` im Datensatz Ihrer WLM-Umgebung hinzu.
 - Wenn Sie sich nicht sicher sind, wo sich Ihr WLM-Umgebungsdatensatz befindet, schauen Sie in der Startprozedur Ihrer gespeicherten DB2 Java-Prozedur nach. Ihr WLM-Umgebungsdatensatz ist das Element, auf das Ihre `JAVAENV DD`-Karte zeigt.
6. Starten Sie die gespeicherten Prozeduren. Erstellen Sie dann Ihre Warehouse-Schritte, und führen Sie sie aus.

Einschränkung für gespeicherte Java-Prozeduren

Java-Objekte in der Kennung einer gespeicherten Prozedur werden nur in DB2 für OS/390 Version 7 unterstützt. Aus diesem Grund unterstützen die Umsetzungsprogramme keine Nullwerte in ihren Parametern in DB2 für OS/390 Version 5 oder 6. Wenn Sie in diesen Versionen einen Nullparameter übergeben, verhält er sich wie eine Null. Die Umsetzungsprogramme der Version 5 und 6 behandeln Nullparameter wie Nullzeichenfolgen.

DB2 unterstützt die SQL-Anweisung `COMMIT` in gespeicherten Prozeduren nur in DB2 für OS/390 Version 7. Die gespeicherte Prozedur `INVERTDATA` löscht eine Tabelle in der gespeicherten Prozedur und erstellt sie erneut. Daher benötigt sie eine Anweisung `COMMIT`. Aus diesem Grund wird `IWH.INVERTDATA` nicht in DB2 für OS/390 Version 5 oder 6 unterstützt.

DB2 für OS/390 unterstützt keine benutzerdefinierten Java-Funktionen. `IWH.FORMATDATE` wird deshalb auf der OS/390-Plattform nicht unterstützt.

Beispielstartprozedur für gespeicherte Java-Prozeduren

(beschrieben im Handbuch *Application Programming Guide and Reference for Java*):

```
//DSNWLMJ PROC DB2SSN=DSN,NUMTCB=5,APPLENV=DSNWLMJ  <-- WLM ENVIRONMENT value
                                                    in CREATE PROC
//*****
//*          THIS PROC IS USED TO START THE WLM-ESTABLISHED SPAS          *
//*          ADDRESS SPACE FOR THE DSNWLMJ APPLICATION ENVIRONMENT        *
//*          V WLM,APPLENV=DSNWLMJ,RESUME                                *
//*****
//DSNWLMJ EXEC PGM=DSNX9WLM,TIME=1440,REGION=0M,
//          PARM='&DB2SSN, &NUMTCB, &APPLENV'
//STEPLIB DD DSN=DSN.TESTLIB,DISP=SHR
//          DD DSN=IWH710.SIWHPDSE,DISP=SHR          <-- This has the transformers in it
//          DD DSN=DSN.HPJSP.PDSE.JDBC,DISP=SHR      <-- HPJ DLLs from HPJ setup
//          DD DSN=SYS1.PP.PDSELINK,DISP=SHR         <-- HPJ runtime libraries
//          DD DSN=DSN710.SDSNEXIT,DISP=SHR
//          DD DSN=DSN710.SDSNLOAD,DISP=SHR
//          DD DSN=SYS1.SCEERUN,DISP=SHR
//          DD DSN=DSN.PDSE,DISP=SHR                <-- HPJ setup info
//JAVAENV  DD DSN=DSN.WLMENVJ.JSPENV,DISP=SHR       <-- Environment variables, see below
//CEEDUMP  DD SYSOUT=A
//DSSPRINT DD SYSOUT=A
//JSPDEBUG DD SYSOUT=A
//SYSABEND DD SYSOUT=A
//SYSPRINT DD SYSOUT=A
```

Beispieldatensatz für Umgebungsvariablen

(beschrieben im Handbuch *Application Programming Guide and Reference for Java*):

```
ENVAR("TZ=PST07",  
"DB2SQLJPROPERTIES=/usr/lpp/db2/jdbc/db2710/classes/db2sqljdbc.properties",  
"LIBPATH=/usr/lpp/DWC",  
"VWSPATH=/usr/lpp/DWC",  
"CLASSPATH=/usr/lpp/db2/jdbc/db2710/classes:/usr/lpp/DWC:/usr/lpp/hpj/lib"),  
MSGFILE(JSPDEBUG)
```

Unterstützung in der Landessprache für die Umsetzungsprogramme

Die meisten Nachrichten, die vom OS/390-Agenten erzeugt werden, werden zur Interpretation an die NT-Plattform gesendet. In den meisten Fällen hängt deshalb die Nachrichtensprache davon ab, wie UDB für NT installiert wurde. Die Umsetzungsprogramme bilden hiervon eine Ausnahme. Der OS/390-Agent wird mit den folgenden Nachrichtendateien für die Umsetzungsprogramme ausgeliefert:

Dateiname:	Für die Sprache:
Xf.properties_Fi_FI	Finnisch in Finnland
Xf.properties_No_NO	Norwegisch in Norwegen
Xf.properties_Ru_RU	Russisch in Russland
Xf.properties_Zh_CN	Chinesisch in China (Volksrepublik China)
Xf.properties_Zh_TW	Chinesisch in Taiwan
Xf.properties_Da_DK	Dänisch in Dänemark
Xf.properties_De_DE	Deutsch in Deutschland
Xf.properties_En_US	Englisch in USA
Xf.properties_Es_ES	Spanisch in Spanien
Xf.properties_Fr_FR	Französisch in Frankreich
Xf.properties_It_IT	Italienisch in Italien
Xf.properties_Ja_JP	Japanisch in Japan
Xf.properties_Ko_KR	Koreanisch in Korea
Xf.properties_Pt_BR	Portugiesisch in Brasilien
Xf.properties_Sv_SE	Schwedisch in Schweden

Wenn die Nachrichten Ihrer Umsetzungsprogramme in einer anderen Sprache als Englisch angezeigt werden sollen, wählen Sie eine der Dateien oben aus, und kopieren Sie den Inhalt in Xf.properties.

Zugreifen auf Datenbanken außerhalb der DB2-Systemfamilie

Für den Zugriff auf andere Datenbanken als DB2 Universal Database verwendet der OS/390-Agent DataJoiner. Mit DataJoiner kann der Agent normalen DRDA-Fluss zu ihm verwenden, als wäre es eine UDB-Datenbank. Wenn eine ODBC-Anfrage an eine andere Datenbank übertragen wird, ruft DataJoiner eine zusätzliche Code-Ebene auf, um auf fremde Datenbanken zuzugreifen.

DataJoiner kann auf Oracle-, Sybase-, Informix-, Microsoft SQL Server-, Teradata-Datenbanken und allen anderen Datenbanken mit einem ODBC-Treiber zugreifen, der unter Windows NT, AIX oder der Sun Solaris-Betriebsumgebung ausgeführt wird. Der OS/390-Agent kann auf DataJoiner als Quelle, nicht jedoch als Ziel zugreifen. DataJoiner unterstützt keine zweiphasige Festschreibung. DataJoiner unterstützt TCP/IP zwar als Anwendungs-Requester in den Versionen 2.1 und 2.1.1, verfügt jedoch über keinen Anwendungs-Server. Da der OS/390-Agent für die Verwendung von TCP/IP einen Anwendungs-Server benötigt, müssen Sie stattdessen für den Zugriff auf DataJoiner von OS/390 aus eine SNA-Verbindung verwenden.

Zugriff auf IMS und VSAM unter OS/390

Classic Connect ist separat vom Warehouse-Agenten zu erwerben und zu installieren. Der OS/390-Agent kann über den ODBC-Treiber von Classic Connect auf IMS und VSAM zugreifen. Mit Classic Connect können Sie eine DB2-ähnliche Definition von IMS- und VSAM-Datensätzen erstellen und dann über ODBC auf sie zugreifen.

Der OS/390-Agent lädt den korrekten ODBC-Treiber, je nachdem, ob eine Anfrage an Classic Connect oder DB2 übertragen wird. Wenn Sie auf eine DB2-Quelle zugreifen, lädt der Agent den ODBC-Treiber von DB2. Wenn Sie auf eine VSAM- oder IMS-Quelle zugreifen, lädt der Agent den ODBC-Treiber von Classic Connect. Die Agentenanfrage wird dann verarbeitet.

Definieren des Classic Connect ODBC-Treibers und des Warehouse-Zugriffs

Classic Connect ist separat vom OS/390-Agenten zu erwerben und zu installieren. Classic Connect kann eine einzelne Datei oder einen Teil einer Datei als eine oder mehrere relationale Tabellen anzeigen. Sie müssen die IMS- und VSAM-Daten für Classic Connect zuordnen, um darauf zugreifen zu können. Sie können die Daten manuell oder mit dem nichtrelationalen Classic Connect Data Mapper von Microsoft Windows zuordnen.

1. Installieren Sie Classic Connect Data Server unter OS/390.
2. Optional: Installieren Sie Classic Connect Data Mapper unter NT.
3. Definieren Sie die logischen Tabellendefinitionen von Classic Connect, so dass Classic Connect relational auf Daten zugreifen kann. Sie können Data Mapper verwenden, um die Definitionen für IMS- und VSAM-Strukturen zu erstellen, oder Definitionen manuell erstellen.
4. Sobald Sie Classic Connect eingerichtet haben, können Sie den Zugriff auf Ihr Warehouse einrichten.
 - a. Erstellen Sie eine .ini-Datei für Classic Connect.

Die Beispielanwendungskonfigurationsdatei für Classic Connect, `cca.ini`, befindet sich im Verzeichnis `/usr/lpp/DWC/`, und ist hier für Sie abgebildet:

```
* national language for messages
NL = US English
* resource master file
NL CAT = usr/lpp/DWC/v4r1m00/msg/engcat
FETCH BUFFER SIZE = 32000
DEFLOC = CXASAMP
USERID = uid
USERPASSWORD = pwd
DATASOURCE = DJX4DWC tcp/9.112.46.200/1035
MESSAGE POOL SIZE = 1000000
```

- b. Aktualisieren Sie die Zeile `DATASOURCE` in der .ini-Datei. Diese Zeile enthält einen Datenquellennamen und eine Protokolladresse. Der Datenquellennamen muss einem Query Processor-Namen entsprechen, der auf dem Datenserver von Classic Connect definiert ist, der wiederum in `QUERY PROCESSOR SERVICE INFO ENTRY` in der Konfigurationsdatei des Datenservers definiert ist. Die Protokolladresse ist in derselben Datei im Eintrag `TCP/IP SERVICE INFO` zu finden. Die Benutzer-ID (`USERID`) und das Benutzerkennwort (`USERPASSWORD`) in dieser Datei werden beim Definieren einer Warehouse-Datenquelle verwendet.

- c. Exportieren Sie die Umgebungsvariable CXA_CONFIG in Ihre Classic Connect-Programmdateien, die sich in der Regel im selben Verzeichnis wie Ihre .ini-Datei befinden.
- d. Aktualisieren Sie Ihre Umgebungsvariable LIBPATH, um den Pfad zu Ihren Programmdateien von Classic Connect, die sich in der Regel in demselben Verzeichnis wie die .ini-Datei befinden, aufzunehmen.
- e. Optional: Prüfen Sie die Installation mit dem Testprogramm cxasamp: Geben Sie cxasamp von dem Verzeichnis aus ein, das Ihre .ini-Datei enthält. Position/Benutzer-ID/Kennwort ist Name/Benutzer-ID/Benutzerkennwort der Datenquelle, wie in Ihrer .ini-Datei definiert.
- f. Definieren Sie wie gewohnt eine Datenquelle für das Warehouse.

Sie müssen Ihre Datei dsnaoini nicht aktualisieren, da DB2 für OS/390 keinen Treibermanager hat. Der Treibermanager für Classic Connect ist in den OS/390-Agenten integriert.

Ausführen von Dienstprogrammen von DB2 für OS/390

Sie müssen APAR PQ44904 auf den OS/390-Agenten anwenden, um mit dem Agenten Dienstprogramme ausführen zu können.

DSNUTILS ist eine gespeicherte DB2 für OS/390-Prozedur, die in einer WLM- und RRS-Umgebung ausgeführt wird. Sie können sie verwenden, um beliebige installierte DB2-Dienstprogramme mit der Schnittstelle für benutzerdefinierte gespeicherte Prozeduren auszuführen. Die Dienstprogramme LOAD, REORG und RUNSTATS von DB2 für OS/390 haben Eigenschaftsseiten, über die Sie die Ausführungsart des Dienstprogramms ändern können. Wenn Sie die Merkmale der Dienstprogramme ändern wollen, klicken Sie mit Maustaste 2 das Dienstprogramm im Fenster **Prozesserstellungsprogramm** an, und klicken Sie anschließend **Merkmale** an.

DB2 Warehouse Manager bietet außerdem eine Schnittstelle zu DSNUTILS, um die Aufnahme von DB2-Dienstprogrammen in DB2 Warehouse Manager-Schritte zu ermöglichen.

Gehen Sie wie folgt vor, um eine gespeicherte DSNUTILS-Prozedur einzurichten:

1. Führen Sie den Job DSNTIJSJG beim Installieren von DB2 aus, um die gespeicherte DSNUTILS-Prozedur einzustellen und zu binden. Stellen Sie sicher, dass die Definition den Parameterstil GENERAL hat.
2. Aktivieren Sie die von WLM verwalteten gespeicherten Prozeduren.
3. Richten Sie Ihre RRS- und WLM-Umgebungen ein.
4. Führen Sie die von DB2 bereitgestellten DSNUTILS-Beispielstapelprogramme aus. (Dieser Schritt ist nicht erforderlich, wird aber empfohlen.)
5. Binden Sie den DSNUTILS-Zugriffsplan mit Ihrem DSNCLI-Zugriffsplan, so dass CLI die gespeicherte Prozedur aufrufen kann:

```

BIND PLAN(DSNAOCLI) PKLIST(*.DSNAOCLI.*, *.DSNUTILS.*)

```
6. Definieren Sie einen Schritt mit Hilfe von DB2 Warehouse Manager, und führen Sie ihn aus. Der Fülltyp sollte APPEND lauten. *Andernfalls löscht DB2 Warehouse Manager vor der Ausführung des Dienstprogramms alle Daten in der Tabelle.*

Kopieren von Daten zwischen Tabellen von DB2 für OS/390 mit dem Dienstprogramm LOAD

Angenommen, Sie wollen eine Tabelle kopieren, indem Sie sie in eine unstrukturierte Datei entladen und dann die unstrukturierte Datei in eine andere Tabelle laden. Dazu müssen Sie normalerweise die Daten entladen, die Ladesteueranweisungen für das Entladen editieren und dann die Daten laden. Mit dem Warehouse können Sie angeben, dass Sie die Daten in eine andere Tabelle laden wollen, ohne zwischen den Schritten anhalten und die Steueranweisungen manuell editieren zu müssen. Gehen Sie dazu wie folgt vor: Verwenden Sie die Schnittstelle Reorg/Generic, um einen Schritt zu erstellen, der eine Datei mit dem Dienstprogramm UNLOAD oder dem Dienstprogramm REORG TABLESPACE entlädt. Beide Dienstprogramme erzeugen zwei Ausgabedatensätze: einen mit den Tabellendaten und einen mit der Dienstprogrammsteueranweisung, die in LOAD eingegeben werden kann. In der Steueranweisung, die das Dienstprogramm generiert, ist der Tabellename INTO TABLE der Name der aus der Tabelle geladenen Tabelle. Im Folgenden sehen Sie ein Beispiel der DSNUTILS-Parameter, die Sie für den Schritt Reorg Unload verwenden könnten:

Tabelle 33. Schrittmerkmale - Reorg Unload

UTILITY_ID	REORGULX
RESTART	NO
UTSTMT	REORG TABLESPACE DBVW.USAINENT UNLOAD EXTERNAL
UTILITY_NAME	REORG TABLESPACE
RECDSN	DBVW.DSNURELD.RECDSN
RECDEVT	SYSDA
RECSpace	50
PNCHDSN	DBVW.DSNURELD.PNCHDSN
PNCHDEVT	SYSDA
PNCHSpace	3

Verwenden Sie die DSNUTILS-Schnittstelle Reorg/Generic, um einen Ladeschritt zu erstellen. Normalerweise gibt der Anweisungsparameter des Dienstprogramms DSNUTILS eine Dienstprogrammsteueranweisung an. Die Warehouse-Dienstprogrammsschnittstelle lässt auch einen Dateinamen im Anweisungsfeld des Dienstprogramms zu. Sie können die Datei, die die gültige Steueranweisung enthält, mit dem Schlüsselwort :FILE: und den Namen der zu ladenden Tabelle mit dem Schlüsselwort :TABLE: angeben. Wenn Sie das Dienstprogramm LOAD zum Arbeiten mit der Ausgabe aus dem vorherigen Beispiel verwenden wollen, wenden Sie die folgenden Parameterwerte in den LOAD-Merkmalen an:

Anmerkung: Geben Sie in das Feld UTSTMT entweder eine Ladeanweisung oder den Namen der Datei ein, die vom Dienstprogramm REORG mit der Option UNLOAD EXTERNAL ausgegeben wurde.

Tabelle 34. Schrittmerkmale - LOAD

UTILITY_ID	LOADREORG
RESTART	NO
UTSTMT	:FILE:DBVW.DSNURELD.PNCHDSN:TABLE:[DBVW].INVENTORY
UTILITY_NAME	LOAD
RECDSN	DBVW.DSNURELD.RECDSN
RECDEVT	SYSDA

Dies funktioniert für alle Quellen- und Zieltabellen von DB2 für OS/390 im selben oder in unterschiedlichen DB2-Subsystemen. Die unstrukturierte Datei mit den Steueranweisungen kann entweder eine HFS- oder eine native MVS-Datei sein.

Ausführlichere Anweisungen zu DSNUTILS und den für OS/390 verfügbaren DB2-Dienstprogrammen finden Sie im Handbuch *DB2 for OS/390 Utility Guide and Reference*.

Replikation

Sie können den OS/390-Agenten verwenden, um Ihre Data Propagator-Schritte der Apply-Komponente bei der Datenreplikation zu automatisieren. Die Replikation erfordert eine Quelldatenbank, eine Steuerungsdatenbank und eine Zieldatenbank. Hierbei kann es sich um verschiedene oder dieselben Datenbanken handeln. Ein Capture-Job liest das DB2-Protokoll, um festzustellen, welche der Zeilen in der Quelldatenbank hinzugefügt, aktualisiert oder geändert wurden. Der Job schreibt die Änderungen dann in eine CD-Tabelle. Dann wird ein Apply-Job ausgeführt, um die Änderungen auf eine Zieldatenbank anzuwenden. DB2 Warehouse Manager kann den Apply-Job automatisieren, indem ein Replikationsschritt erstellt wird. Mit DB2 Warehouse Manager können Sie den auszuführenden Typ des Apply-Jobs sowie die Ausführungszeit definieren. Sie müssen Ihre SASNLINK-Bibliothek in die Umgebungsvariable `steplib` exportieren.

Hinzufügen von Replikationsunterstützung zur Schablone der Data Warehouse-Zentrale

Die Data Warehouse-Zentrale umfasst eine JCL-Schablone für die Replikationsunterstützung. Wenn Sie den OS/390-Agenten für die Ausführung des Apply-Programms verwenden wollen, müssen Sie die Benutzer- und Dateinformationen in dieser Schablone für Ihr OS/390-System ändern. Gehen Sie wie folgt vor, um die Schablone zu ändern:

1. Melden Sie sich mit einer ID an, die eine Berechtigung zum Kopieren und Aktualisieren von Dateien im Verzeichnis `/usr/lpp/DWC/` hat.
2. Suchen Sie `apply.jcl`, und kopieren Sie diese Datei als `systemname,apply.jcl`. Hierbei ist `systemname` der Name des MVS-Systems. Erstellen Sie beispielsweise auf `STLMVS1` eine Kopie der Datei mit dem Namen `STLMVS1.apply.jcl`.
3. Passen Sie die JCL mit Hilfe eines Texteditors den Anforderungen Ihres Standorts an. Passen Sie die Benutzerinformationen den Standardbenutzerinformationen an, und ändern Sie die Datei für `STEPLIB DD` und `MSGSD DD` für Ihr MVS-System.

4. Falls erforderlich, ändern Sie den Programmnamen auf der EXEC-Karte. Einzelheiten zur Änderung von Programmnamen finden Sie im *Replikation Referenzhandbuch*. Parameter in eckigen Klammern, z. B [USERID] und [FTPFILE], dürfen Sie nicht ändern. (Die eckigen Klammern sind die Hexadezimalzeichen x'AD' bzw. x'BD'. Wenn Ihr TSO-Terminaltyp in SPF-Option 0 nicht auf 3278A gesetzt ist, werden diese Werte möglicherweise als Sonderzeichen und nicht als eckige Klammern angezeigt. Das ist kein Problem, solange Sie nicht x'AD' oder x'BD' oder die Daten zwischen den Zeichen ändern.)
5. Aktualisieren Sie die Umgebungsvariable VWS_TEMPLATES so, dass sie auf das Verzeichnis der kopierten Schablonendatei zeigt.

Das folgende Beispiel zeigt die JCL-Schablone, die in der Data Warehouse-Zentrale enthalten ist:

Anwenden der JCL-Schablone:

```
//[USERID]A JOB ,MSGCLASS=H,MSGLEVEL=(1,1),
// REGION=2M,TIME=1440,NOTIFY=&SYSUID
//* DON'T CHANGE THE FIRST LINE OF THIS TEMPLATE.
//* THE REMAINING JCL SHOULD BE MODIFIED FOR YOUR SITE.
//*****
//* RUN APPLY/MVS ON OS/390 DB2 6.1 *
//*****
//ASNARUN EXEC PGM=ASNAPV66,REGION=10M,
// [APPLY_PARMS]
//STEPLIB DD DISP=SHR,DSN=DPROPR.V6R1M0.SASNLINK
// DD DISP=SHR,DSN=DSN610.SDSNLOAD
//MSGS DD DSN=DPROPR.V2R1M0A.MSGS,DISP=SHR
//ASNASPL DD DSN=&&ASNASPL,DISP=(NEW,DELETE,DELETE),
// UNIT=SYSDA,SPACE=(CYL,(10,1)),
// DCB=(RECFM=VB,BLKSIZE=6404)
//SYSTEM DD SYSOUT=*
//SYSTSPRT DD SYSOUT=*
//SYSPRINT DD SYSOUT=*
//
```

Agentenprotokollierung

Viele DB2 Warehouse Manager-Komponenten, z. B. Server, Protokollfunktion, Agenten und einige Programme der Data Warehouse-Zentrale, schreiben Protokolle in das Protokollverzeichnis, das in der Umgebungsvariable VWS_LOGGING angegeben ist. Diese Protokolldateien sind Textdateien.

Sie können den Agenten starten, indem Sie sich an der Data Warehouse-Zentrale anmelden. Klicken Sie im linken Teilfenster **Warehouse** mit Maustaste 2 an, und klicken Sie anschließend **Merkmale** an. Ändern Sie auf der Indexzunge **Trace-Stufe** die Einstellungen so, dass Sie die gewünschte Trace-Stufe erhalten. Der Agenten-Trace unterstützt die Ebenen 0-4:

- Ebene 1 - Ein-/Ausgangs-Trace
- Ebene 2 - Ebene 1 plus Debug-Trace
- Ebene 3 - Ebene 2 plus Daten-Trace
- Ebene 4 - interner Puffer-Trace

Wird der Trace höher als Ebene 1 gesetzt, ist die Leistung geringer. Aktivieren Sie den Trace nur für das Debugging. Die Trace-Informationen werden in der Datei AGNTxxx.LOG gespeichert. Umgebungsinformationen werden in der Datei AGNTxxx.SET gespeichert.

Client-seitiges Caching in Windows NT

Versucht ein Benutzer, auf eine Datenbank-Datei zuzugreifen, für die Lesezugriff erteilt wurde und die sich auf einer Windows NT-Server-Maschine befindet, auf der DB2 Datalinks installiert ist, und verwendet er ein gemeinsames Laufwerk mit einem gültigen Token, wird die Datei wie erwartet geöffnet. Danach erreichen nachfolgende Anforderungen zum Öffnen der Datei mit demselben Token jedoch nicht den Server, sondern werden vom Cache auf dem Client bedient. Auch wenn das Token verfällt, bleibt der Inhalt der Datei weiterhin für den Benutzer sichtbar, weil sich der Eintrag immer noch im Cache befindet. Dieses Problem tritt jedoch nicht auf, wenn sich die Datei auf einer Windows NT-Workstation befindet.

Eine Lösung besteht darin, den Registrierungseintrag `\\HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Lanmanserver\Parameters\EnableOpLocks` auf dem Windows NT-Server auf Null zu setzen. Mit dieser Einstellung in der Registrierdatenbank erreicht eine Zugriffsanforderung über ein gemeinsames Laufwerk von einer Client-Workstation für eine Datei, die sich auf dem Server befindet, immer den Server und wird nicht vom Client-Cache bedient. Daher wird das Token für alle Anforderungen erneut ausgewertet.

Der Nachteil dieser Lösung besteht darin, dass die Gesamtleistung für alle Dateizugriffe vom Server über gemeinsame Laufwerke beeinträchtigt wird. Erfolgt der Zugriff auf die Datei über eine Zuordnung für gemeinsame Laufwerke auf dem Server selbst und nicht von einer anderen Client-Maschine, scheint es selbst mit dieser Einstellung so, als würde die Anforderung immer noch vom Cache bedient. Daher wird der Token-Verfall nicht wirksam.

Anmerkung: Erfolgt der Dateizugriff lokal und nicht über ein gemeinsames Laufwerk, treten in allen Fällen Gültigkeitsprüfung und nachfolgender Verfall des Tokens wie erwartet ein.

Testprodukte auf den CD-ROMs von Enterprise Edition für UNIX

Die CD-ROMs für DB2 Universal Database (UDB) Enterprise Edition (EE) für UNIX-Plattformen Version 6 und Version 7 enthalten eine 90 Tage gültige Testversion von DB2 Connect Enterprise Edition (CEE). Da die Funktionalität von DB2 Connect in DB2 UDB EE integriert ist, müssen Sie DB2 CEE auf Systemen, auf denen DB2 UDB EE installiert ist, nicht installieren, um die DB2 Connect-Funktionalität nutzen zu können. Wenn Sie die Testversion von DB2 CEE installieren und auf eine lizenzierte Version aufrüsten wollen, müssen Sie DB2 CEE erwerben und die DB2 CEE-Lizenzberechtigung installieren. Das Produkt muss nicht erneut installiert werden. Die Anweisungen zur Installation der Lizenzberechtigung finden Sie in den Handbüchern *Einstieg* (Quick Beginnings) von DB2 EE oder DB2 CEE für UNIX.

Wenn Sie das CEE-Testprodukt zusammen mit Ihrer EE-Installation installiert haben und CEE nicht dauerhaft nutzen wollen, können Sie die CEE-Testversion mit den folgenden Schritten entfernen. Wenn Sie die Testversion von Connect EE entfernen, ist dennoch DB2 Connect-Funktionalität in Verbindung mit DB2 EE verfügbar.

Zum Entfernen von DB2 Connect Version 7 deinstallieren Sie die folgenden Dateigruppen von den entsprechenden Plattformen:

- AIX: Deinstallieren Sie die Dateigruppe `db2_07_01.clic`.
- NUMA-Q und Solaris-Betriebsumgebungen: Deinstallieren Sie das Paket `db2clic71`.
- Linux: Deinstallieren Sie den RPM `db2clic71-7.1.0-x`.
- HP-UX: Deinstallieren Sie die Dateigruppe `DB2V7CONN.clic`.

Zum Entfernen von DB2 Connect Version 6 deinstallieren Sie die folgenden Dateigruppen von den entsprechenden Plattformen:

- AIX: Deinstallieren Sie die Dateigruppe `db2_06_01.clic`.
- NUMA-Q und Solaris-Betriebsumgebungen: Deinstallieren Sie das Paket `db2cplic61`.
- Linux: Deinstallieren Sie den RPM `db2cplic61-6.1.0-x`.
- HP-UX: Deinstallieren Sie die Dateigruppe `DB2V6CONN.clic`.

Testprodukte auf den CD-ROMs von DB2 Connect Enterprise Edition für UNIX

Die CD-ROMs für DB2 Connect Enterprise Edition (EE) für UNIX-Plattformen Version 6 und Version 7 enthalten eine 90 Tage gültige Testversion von DB2 Universal Database (UDB) Enterprise Edition (EE). Diese Testversion von DB2 UDB EE wird zu Beurteilungszwecken zur Verfügung gestellt, ist jedoch nicht erforderlich für die Funktion von DB2 Connect.

Wenn Sie die Testversion von DB2 UDB EE installieren und auf eine lizenzierte Version aufrüsten wollen, müssen Sie DB2 UDB EE erwerben und die DB2 UDB EE-Lizenzberechtigung installieren. Das Produkt muss nicht erneut installiert werden. Die Anweisungen zur Installation der Lizenzberechtigung finden Sie in den Handbüchern *Einstieg* (Einstieg) von DB2 EE oder DB2 CEE für UNIX. Wenn Sie das UDB EE-Testprodukt zusammen mit Ihrer Connect EE-Installation installiert haben und UDB EE nicht dauerhaft nutzen wollen, können Sie die Testversion mit den folgenden Schritten entfernen. Wenn Sie die Testversion von DB2 UDB EE entfernen, wirkt sich dies nicht auf die Funktionalität von DB2 Connect EE aus.

Zum Entfernen von DB2 UDB EE Version 7 deinstallieren Sie die folgenden Dateigruppen von den entsprechenden Plattformen:

- AIX: Deinstallieren Sie die Dateigruppe `db2_07_01.elic`.
- NUMA-Q und Solaris-Betriebsumgebungen: Deinstallieren Sie das Paket `db2elic71`.
- Linux: Deinstallieren Sie den RPM `db2elic71-7.1.0-x`.
- HP-UX: Deinstallieren Sie die Dateigruppe `DB2V7ENTP.elic`.

Zum Entfernen von DB2 UDB EE Version 6 deinstallieren Sie die folgenden Dateigruppen von den entsprechenden Plattformen:

- AIX: Deinstallieren Sie die Dateigruppe `db2_06_01.elic`.
- NUMA-Q und Solaris-Betriebsumgebungen: Deinstallieren Sie das Paket `db2elic61`.
- Linux: Deinstallieren Sie den RPM `db2elic61-6.1.0-x`.
- HP-UX: Deinstallieren Sie die Dateigruppe `DB2V6ENTP.elic`.

Löschen von Data Links Manager

Sie können jetzt einen DB2 Data Links Manager für eine bestimmte Datenbank löschen. Die Verarbeitung einiger Data Links-bezogener SQL-Anforderungen sowie Dienstprogramme wie backup/restore erfordert die Kommunikation mit allen DLMs, die in einer Datenbank konfiguriert sind. Bisher hatte DB2 nicht die Möglichkeit, einen konfigurierten DLM zu löschen, selbst wenn er nicht mehr betriebsbereit war. Dies führte zu zusätzlichem Systemaufwand bei der SQL- und Dienstprogrammverarbeitung. Sobald ein DLM hinzugefügt wurde, kommunizierte die Steuerkomponente mit ihm bei der Verarbeitung von Anforderungen, was zu Fehlern einiger SQL-Anforderung führen konnte (z. B. drop table/tablespace/database).

Deinstallation von DLFM-Komponenten mit SMIT entfernt unter Umständen zusätzliche Dateigruppen

Vor der Deinstallation von DB2 (Version 5, 6 oder 7) von einer AIX-Maschine, auf der Data Links Manager installiert ist, führen Sie folgende Schritte aus:

1. Als Root erstellen Sie eine Kopie von /etc/vfs mit folgendem Befehl:

```
cp -p /etc/vfs /etc/vfs.bak
```
2. Deinstallieren Sie DB2.
3. Als Root ersetzen Sie /etc/vfs durch die in Schritt 1 erstellte Sicherungskopie:

```
cp -p /etc/vfs.bak /etc/vfs
```

Fehler SQL1035N bei der Verwendung des Befehlszeilenprozessors unter Windows 2000

Wenn DB2 in einem Verzeichnis installiert ist, auf das nur einige Benutzer (z. B. Administratoren) Schreibzugriff haben, kann ein normaler Benutzer den Fehler SQL1035N erhalten, wenn er versucht, den DB2-Befehlszeilenprozessor zu verwenden.

Zur Behebung dieses Problems sollte DB2 in einem Verzeichnis installiert werden, auf das alle Benutzer Schreibzugriff haben.

Verbesserung bei SQL Assist

Das Tool **SQL Assist** ermöglicht es jetzt dem Benutzer, einen anderen Verknüpfungsoperator als "=" für Tabellenverknüpfungen anzugeben. Der Dialog **Verknüpfungsart**, der durch Anklicken des Knopfs **Verknüpfungsart** von SQL Assist geöffnet wird, wurde erweitert, so dass er jetzt eine Dropdown-Liste mit Verknüpfungsoperatoren enthält.

Verfügbar sind die Operatoren "=", "<>", "<", ">", "<=" und ">=". SQL Assist ist ein Tool, das dem Benutzer bei der Erstellung einfacher SQL-Anweisungen hilft. Es ist verfügbar in der Befehlszentrale (Indexzunge **Interaktiv**), der Steuerzentrale (Dialog **Sicht erstellen** und **Auslöser erstellen**), dem Stored Procedure Builder (Assistent für das Einfügen von gespeicherten SQL-Prozeduren) und in der Data Warehouse-Zentrale (SQL-Verarbeitungsschritt).

DB2 für Linux - Integration der Gnome- und KDE-Desktops

Die neue DB2-Version enthält eine Reihe von Dienstprogrammen zur Erstellung von Ordnern und Symbolen für DB2-Desktops, mit deren Hilfe die gängigsten DB2-Tools auf den Gnome- und KDE-Desktops für unterstützte, Intel-basierte Linux-Versionen gestartet werden können. Diese Dienstprogramme, die von DB2 Version 7.2 standardmäßig installiert werden, können nach der Installation zum Erstellen und Entfernen von Desktop-Symbolen für einen oder mehrere ausgewählte Benutzer verwendet werden.

Verwenden Sie zum Hinzufügen von Desktop-Symbolen für einen oder mehrere Benutzer den folgenden Befehl:

```
db2icons <benutzer1> [<benutzer2> <benutzer3>...]
```

Anmerkung: Wenn bei der Symbolerstellung eine Desktop-Umgebung für Gnome oder KDE aktiv ist, muss der Benutzer möglicherweise eine manuelle Desktop-Aktualisierung vornehmen, damit die neuen Symbole angezeigt werden.

Verwenden Sie zum Entfernen von Desktop-Symbolen für einen oder mehrere Benutzer den folgenden Befehl:

```
db2rmicons <benutzer1> [<benutzer2> <benutzer3>...]
```

Anmerkung: Zum Erstellen oder Entfernen von Symbolen für andere Benutzer müssen die entsprechenden Berechtigungen vorliegen. Ein normaler Benutzer kann mit **db2icons** und **db2rmicons** in der Regel nur eigene Symbole erstellen und entfernen. Für einen anderen Benutzer muss eine Root-Berechtigung oder eine Schreibberechtigung für die entsprechenden Ausgangsverzeichnisse vorliegen.

Ausführen von DB2 unter einem Windows 2000-Terminalserver (im Verwaltungsmodus)

Bei DB2 UDB Version 7.1 ab FixPak 3 kann DB2 unter einem Windows 2000-Terminalserver (im Verwaltungsmodus) ausgeführt werden. Zuvor konnte DB2 im Rahmen einer Client-Sitzung eines Windows 2000-Terminalservers (im Verwaltungsmodus) nicht ausgeführt werden.

Onlinehilfefunktion für Befehle zum Sichern und Wiederherstellen

Bei Eingabe von `db2 ? backup` werden falsche Informationen angezeigt. Die richtige Ausgabe lautet:

```
BACKUP DATABASE aliasname-der-datenbank [USER benutzername [USING kennwort]]
[TABLESPACE (tabellenbereichsname [ {,tabellenbereichsname} ... ])] [ONLINE]
[INCREMENTAL [DELTA]] [USE TSM [OPEN anzahl-der-sitzungen SESSIONS]] |
TO dir/dev [ {,dir/dev} ... ] | LOAD bibliotheksname
[OPEN anzahl-der-sitzungen SESSIONS]]
[WITH anzahl-der-puffer BUFFERS] [BUFFER puffergröße] [PARALLELISM n]
[WITHOUT PROMPTING]
```

Bei Eingabe von `db2 ? restore` werden falsche Informationen angezeigt. Die richtige Ausgabe lautet:

```
RESTORE DATABASE aliasname-der-quellendatenbank { wiederherstellungsoptionen |
CONTINUE | ABORT }";
```

```
wiederherstellungsoptionen:";
[USER benutzername [USING kennwort]] [{TABLESPACE [ONLINE] |};
TABLESPACE (tabellenbereichsname [ {,tabellenbereichsname} ... ]) [ONLINE] |";
HISTORY FILE [ONLINE]]} [INCREMENTAL [ABORT]]";
[{USE TSM [OPEN anzahl-der-sitzungen SESSIONS] |};
FROM dir/dev [ {,dir/dev} ... ] | LOAD gemeinsame-bibliothek";
[OPEN anzahl-der-sitzungen SESSIONS]]} [TAKEN AT datum-uhrzeit]
[TO zielverzeichnis]";
[INTO aliasname-der-zielendatenbank] [NEWLOGPATH verzeichnis]";
[WITH anzahl-der-puffer BUFFERS] [BUFFER puffergröße]";
[DLREPORT dateiname] [REPLACE EXISTING] [REDIRECT] [PARALLELISM n]";
[WITHOUT ROLLING FORWARD] [WITHOUT DATALINK] [WITHOUT PROMPTING]";
```

"Warehouse Manager" muss "DB2 Warehouse Manager" heißen

Alle Vorkommen des Ausdrucks "Warehouse Manager" in Produktanzeigen und in der Produktdokumentation sollten "DB2 Warehouse Manager" lauten.

Teil 9. Zusätzliche Informationen

Zusätzliche Informationen

Online-Unterstützung für DB2 Universal Database und DB2 Connect

Eine vollständige und aktuelle DB2-Informationsquelle, die auch Informationen zu Fragen enthält, die erst nach der Veröffentlichung dieses Dokuments aufgetreten sind, finden Sie auf der Website für die Unterstützung von DB2 Universal Database und DB2 Connect unter

<http://www.ibm.com/software/data/db2/udb/winos2unix/support>.

DB2-Magazin

Um die jeweils aktuellen Informationen zur DB2-Produktfamilie zu erhalten, können Sie die Zeitschrift "DB2 magazine" (nur in englischer Sprache) kostenlos abonnieren. Die Online-Ausgabe dieser Zeitschrift ist verfügbar unter

<http://www.db2mag.com>.

Anweisungen zum Abonnieren finden Sie ebenfalls auf dieser Site.

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 Dokument sowohl alte als auch neue Termini gleichbedeutend verwendet werden. Dies ist der Fall, wenn auf ältere existierende Handbuchausschnitte und/oder Programmteile zurückgegriffen wird.

Teil 10. Anhänge und Schlussteil

Anhang. 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. An Stelle 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.

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

- Abfrageergebnisse, anzeigen 215
- Address Windowing Extensions (AWE) 114
- Adobe Acrobat Reader
 - Zugriff mit anderen Ländereinstellungen als Englisch 13
- ADT transforms 366
- AIX 4.3.3
 - AutoLoader blockiert nach Aufspaltung 494
- Alert-Zentrale 73
- ALTER SEQUENCE statement
 - detailed description 418
- ambiguous cursor 440
- Anweisungsebene, Isolation 105
- app_ctl_heap_sz 119
- Archive Log 149
- Archivieren und abrufen
 - Benutzer-Exit-Programm 101
- Archivierung von Protokollen nach Bedarf 89
- AS/400
 - Capture und Apply für 176
- assigning strings to columns, rules for 473
- assignments
 - strings, basic rules for 473
- Assistent: Datenbank erstellen
 - Speicherbereichsgröße einstellen in 219
- Assistent "MQSeries Assist" 219
- Assistent "OLE DB Assist" 219
- Assistenten
 - Datenbank erstellen 219
 - MQSeries Assist 219
 - OLE DB Assist 219
- Ausdrücke
 - NEXTVAL 76
 - PREVVAL 76
- Authentifizierung
 - nach Domäne mit DB2DOMAINLIST 494
- authorization
 - granting create on sequence 443
 - public create on sequence 443
- AutoLoader
 - blockiert während Aufspaltung 494

B

- Baltic Rim languages
 - code page support 71
- Bedarfsorientierte Protokollarchivierung 89
- Befehlsprachendatei, Data Warehouse-Zentrale, ICM, ERwin-Datei, ER1-Datei 233

- Behälter
 - einem SMS-Tabellenbereich hinzufügen 78
- Beispielinhalte, anzeigen 215
- Benannte Pipes
 - sichern auf 87
- Benutzer-IDs
 - Zeichenbegrenzungen 66
- Bereitstellungsfunktionen
 - bei paralleler DMS-Behältererstellung und -erweiterung 73
- Bidirektionale Sprachunterstützung
 - mit der Steuerzentrale unter Windows NT 209
- binding database utilities using Run-Time Client 365

C

- Capture und Apply
 - aus einer Anwendung starten 180
 - bei AS/400 176
 - Fehlernachrichten 179
- catalog views
 - SEQUENCES 461
- character conversion
 - rules for operations combining strings 475
 - rules when comparing strings 475
- character string
 - assignment, overview 473
- CLI
 - writing a Unicode application 476
- CLI functions
 - SQLBindFileToParam 369
 - SQLNextResult 370
- CLI-Schnittstelle
 - Codepage-Einstellungen 71
- CLI stored procedures
 - autobinding 369
 - limitations 369
- Clients
 - herunterladen 17
 - three-tier environment 367
- Codepages
 - Baltic Rim languages 71
 - in Anwendungen mit CLI 71
 - Zuordnung zu Ländereinstellungen 67
- column
 - string assignment, basic rules for 473
- Commit.tag template
 - example values 282
 - tokens 282
- Common Warehouse Metamodel
 - XML-Unterstützung 278
- Compound-Anweisungen
 - dynamisch 106
- compound SQL, using 368
- Compound Statement (Dynamic)
 - variables 425

- concatenation
 - operator 475
- conversion rules
 - for operations combining strings 475
 - for string comparisons 475
- CREATE PROCEDURE statement
 - dynamic compound statement 425
 - FOR statement 452
 - RETURN statement 456
 - SIGNAL statement 458
 - SQL procedure statement 450
- CREATE SEQUENCE statement
 - detailed description 434
- cursors
 - attributes, default 367
 - concurrency 367
 - keyset-driven
 - updating 367
 - scrollable 367
 - keyset-driven 367
 - OS/390 367
 - server-side 367
 - static 367
 - troubleshooting applications 368
 - sensitivity 367
 - type 367
 - updatable 367
 - values concurrency 367
- CWM
 - XML-Unterstützung 278

D

- data types
 - promotion in a Unicode database 471
- Data Warehouse-Zentrale
 - gespeicherte Prozeduren verwenden 499
 - iwh2exp2, Befehlssyntax 500
 - Unterstützung von OLE DB 255
 - XML-Unterstützung für CWM 278
- DataJoiner
 - Einschränkungen 496
- DATALINK
 - Probleme bei der Wiederherstellung 495
 - Replikation 173
- Dateizugriff verweigert, Fehler 210
- Datenbereinigung, Namens- und Adressbereinigung 241
- Datenverschlüsselung 80
- DAYOFWEEK_ISO scalar function 372
- db2_all 102
- DB2_AWE, Registrierungsvariable 115
- DB2 Connect
 - Vorbedingungen bei Sun Cluster 2.2 64
- DB2 control server
 - minimum requirements 129
 - response file installation 129

- DB2 control server (*Forts.*)
 - Version 6, upgrading 132
- DB2 Data Links Manager
 - Interaktionen bei Wiederherstellung 95
- DB2 Enterprise Edition
 - setting up as satellite 129
 - prerequisites 129
 - Version 6, using as DB2 control server 132
- DB2_INDEX_2BYTEVARLEN 77
- DB2 Personal Edition
 - setting up as satellite 129
 - prerequisites 129
- DB2 Relational Connect
 - Installation unter AIX, Solaris und Linux 41
 - Installation unter Windows NT 40
- DB2 Server für VM und VSE
 - Datenbanken von der Steuerzentrale aus verwalten 209
- db2ArchiveLog 327
- DB2CODEPAGE, Registrierungsvariable in Anwendungen mit CLI 71
- db2expln
 - ausführen 127
- db2inidb, Tool 82
- db2iupdt, Befehl, nach DB2-Upgrade ausführen 18
- DB2MSCS, Dienstprogramm, Setzen von PATH beim Warmstart 64
- db2start
 - on Windows 95, 98 and ME 183
- deadlocks 186
- DECLARE CURSOR 440
- DECRYPT function
 - detailed description 386
 - values and arguments 386
- decrypting information
 - DECRYPT function 386
- Deinstallieren
 - DB2 DFS Client Enabler 494
- DFS Client Enabler
 - deinstallieren 494
- dl_expint, Einstellungsempfehlung 117
- dlfm client_conf
 - Fehlerursachen 493
- DLFS
 - inkompatibel mit Norton Utilities 66
- DMS-Behälter
 - parallel erstellen und erweitern 73
- DMS-Tabellenbereiche
 - parallele Erstellung und Größenänderung 80
- Doppelte Protokollierung 90
- dynamic compound statement 425
- Dynamische Compound-Anweisungen 106
- dynexpln
 - ausführen 127

E

- enable_MQFunctions 204, 206
- ENCRYPT function
 - detailed description 387
 - values and arguments 387

- encrypting information
 - ENCRYPT function 387
 - GETHINT function 389
- error messages
 - executing triggers 440
- ERwin, Metadaten, extrahieren, importieren 231
- expression
 - concatenation operator 475
 - string 475
- expressions
 - sequences 382

F

- Fehlernachrichten
 - bei der Migration 6
 - beim Hinzufügen von Knoten zu partitionierten Datenbanken 4
 - Bindedateien nicht gefunden 492
 - Dateizugriff verweigert 210
 - SQL10012N 214
 - Verknüpfungsfehler 210
- Ferner Client, Zugriff
 - Kommunikations-Server für NT konfigurieren 495
- FOR statement 452
- ForeignKey.tag template
 - example values 284
 - tokens 282
- ForeignKeyAdditional.tag template
 - example values 286
 - tokens 284
- functions
 - in a Unicode database 475
 - procedures 414
 - scalar
 - DECRYPT 386
 - ENCRYPT 387
 - GETHINT 389
 - IDENTITY_VAL_LOCAL 390
 - MQPUBLISH 394
 - MQREAD 396
 - MQRECEIVE 397
 - MQSEND 398
 - MQSUBSCRIBE 400
 - MQUNSUBSCRIBE 401
 - REC2XML 403
 - table
 - MQREADALL 410
 - MQRECEIVEALL 412
- Funktionen
 - DECRYPT 81
 - ENCRYPT 80
 - GETHINT 81

G

- gateways
 - three-tier environment 367
- Gespeicherte Prozeduren
 - in der Data Warehouse-Zentrale 499
- Geteilter Spiegel
 - als Bereitschaftsdatenbank 83
- Geteilter Spiegel, Handhabung
 - online 82

- GETHINT function
 - detailed description 389
 - values and arguments 389
- Governor-Konfigurationsdatei 116
- GRANT
 - CREATE ON SEQUENCE 443
- GRANT (Sequence Privileges) statement
 - detailed description 443
- GUI-Tools
 - Bindefehler 492

H

- High Availability unter Solaris 64
- Hohe Verfügbarkeit 82
- HP und Sun Solaris
 - Unterstützung von Sichern und Wiederherstellen 91
- HP-UX
 - keine Unterstützung für Query Enabler 499
- HP-UX 11
 - Speicherfenster 492

I

- IDENTITY-Spalte
 - ändern 78
- IDENTITY-Spalten 77
- IDENTITY_VAL_LOCAL function
 - detailed description 390
 - values and arguments 390
- Import, Befehlsprachendatei, ICM, Information Catalog Manager 235
- Indexschlüssel
 - größere 107
- Indizes
 - größere Schlüssel 107
 - Mehrfachzugriff 113
- Information Catalog Manager
 - Dienstprogramm 305
- Information Catalog-Manager
 - in Hebrew 496
- Inkrementelle COMMIT-Operation mit Ersetzen 257
- inoperative triggers
 - detailed description 439
- Installation
 - Blockierung bei Wechsellaufwerken 7
 - DB2-Prozesse stoppen während 18
 - Servicebenutzereintrag erforderlich unter Windows 498
- Installationsvorbereitungen
 - für UDB EE und CEE unter Linux für OS/390 19
- Internet Explorer
 - Probleme bei der Steuerzentrale 7
- IPX/SPX-Protokollunterstützung, unter Windows 2000 17
- isolation levels
 - in DELETE statement 441
 - in INSERT statement 444
 - in SELECT INTO statement 445
 - in SELECT statement 416
 - in UPDATE statement 450

Isolationsstufen
Anweisungsebene 105
iwh2exp2, Befehl
neue Option 500

J

Japanese
locale setting on Linux 7
Java methods
in CREATE PROCEDURE or CREATE
FUNCTION 363
JDBC
scrollable cursors, accessing 367
JDBC/ODBC/CLI static profiling, limita-
tions 365
JDK 1.1
Installationspfad 213

K

Klondatenbanken
erstellen 82
Komponenten-Trace-
Datendateinamen 257
Konfigurationsparameter
app_ctl_heap_sz 119
maxappls 119
maxlocks 117
MIN_DEC_DIV_3 118
mon_heap_sz 119
softmax 119
Kopierdämon 494
Kurzüberblick 491

L

Ländereinstellungen
in Anwendungen mit CLI 71
Kompatibilität zwischen Verwaltungs-
server und Exemplar 70, 498
Landessprachenversionen 67
Landessprachenunterstützung
Unicode 469
Linux
locale settings for Japanese and Simp-
lified Chinese 7
Umgebung für Steuerzentrale definie-
ren 18
locale
Japanese and Simplified Chinese on
Linux 7
locales
Simplified Chinese on Red Flag
Linux 6
lock waits 186
locking, optimistic 367

M

maxappls 119
Microsoft SQL Server-Datenquellen
Codepages 56
DSN-Name 54
Leistung verbessern 53

Microsoft SQL Server-Datenquellen
(Forts.)
MERANT-Bibliothek 51
MERANT-Oberflächen 53
mit DB2 verknüpfen 52
ODBC-Traces 56
ODBC-Treiber 51
Umgebungsvariablen 52
Migration
"db2iupdt" zur Aktualisierung von
Exemplaren ausführen 18
Fehler 6
Probleme mit Sichten 17
unter Windows 2000 17
mon_heap_sz 119
MQ-Funktionen 204, 206
MQPUBLISH function
detailed description 394
values and arguments 394
MQREAD function
detailed description 396
values and arguments 396
MQREADALL function
detailed description 410
values and arguments 410
MQRECEIVE function
detailed description 397
values and arguments 397
MQRECEIVEALL function
detailed description 412
values and arguments 412
MQSEND function
detailed description 398
values and arguments 398
MQSUBSCRIBE function
detailed description 400
values and arguments 400
MQUNSUBSCRIBE function
detailed description 401
values and arguments 401

N

Nachrichten
für Capture und Apply 179
Nachrichtenübertragung, MQSeries 251,
252
Namens- und Adressbereinigung, Trilli-
um 241
Namenskonventionen 66
Zeichenbegrenzungen 66
Netscape
Fehlernachrichten 12, 217
für Zugriff auf Onlineinformatio-
nen 11
next key locking 186
NEXTVAL 76
nextval-expression 382
Nicht-DB2-Quelle, Beispieleinträge für
Sybase 257
Nicht-DB2-Warehouse-Quellen, AIX 257
Nicht-DB2-Warehouse-Quellen, Solaris-
Betriebsumgebung 257
Norton Utilities
inkompatibel mit DLFS 66

NUMA-Q
keine Unterstützung für Query Enab-
ler oder Tracker 499

O

ODBC
fetch orientations 367
scrollable cursors, accessing 367
ODBC Unicode applications 481
Open Client, Nicht-DB2-Warehouse-
Quellen und 257
operand
string 475
Oracle-Datenquellen
Linux-Oberfläche 42
Solaris-Oberfläche 42
Umgebungsvariablen 42
OS/2
JDK level required for messages 345
Steuerzentrale installieren 210

P

Parallele Wiederherstellung 86
Partitionierte Datenbanken
Fehler beim Hinzufügen von Kno-
ten 4
Partitionierungsschlüssel
aktualisieren 63
Performance Monitor 73
PREVVAL 76
prevval-expression 382
PrimaryKey.tag template
example values 287
tokens 286
PrimaryKeyAdditional.tag template
example values 289
tokens 288
Programmkorrekturstufen
bei Solaris Version 2.6 4
Protokoll
spiegeln 90
Protokolle
Archivierung nach Bedarf 89
Prozesse, vor Installation stoppen 18
Prozesserstellungsprogramm
Data Warehouse-Zentrale 278
Pufferpools
AWE 114

Q

Query Enabler
nicht unterstützt unter HP-UX und
NUMA-Q 499
Query Patroller
Verlust von Benutzerzugriffsrechten
nach der Migration 499
Query Patroller Tracker
nicht unterstützt unter NUMA-Q 499

R

- REC2XML function
 - detailed description 403
 - values and arguments 403
- Red Flag Linux
 - Simplified Chinese locale 6
- Registrierungsvariable
 - DB2_AWE 115
 - neue und geänderte 122
- Registrierungsvariablen
 - DB2DOMAINLIST 494
 - zur Unterstützung von Sicherungsservice-APIs 500
- Replication Analyzer 174
- Replikation
 - bei Nicht-IBM Servern 171
 - DATALINK 173
 - Einschränkungen bei Tabellen- und Spaltennamen 173
 - Fehlerbestimmung bei 174
 - planen 173
 - Szenarios für 171
 - Vorbedingung für beliebige Tabellenreplikation 174
- Replizierte Übersichtstabellen 63, 111
- response file installation
 - DB2 control server keywords 129
 - satellite-specific keywords 130
- return identity column value
 - IDENTITY_VAL_LOCAL function 390
- RETURN statement 456
- routines
 - procedures 414

S

- satellite environment
 - administering with Version 6
 - Satellitenverwaltungszentrale 139
 - DB2 Personal Edition satellite 129
 - DB2 Workgroup Edition satellite 129
 - installation considerations 130
 - interactive installation of satellite 131
 - prerequisites 129
 - response file installation of satellite 130
- Satellitenverwaltungszentrale
 - Version 6, using to administer satellite environment 139
- satellites
 - DB2 Personal Edition 129
 - DB2 Workgroup Edition 129
 - installation considerations 130
 - interactive installation 131
 - response file installation 130
- scalar functions
 - DAYOFWEEK_ISO 372
 - WEEK_ISO 372
- Schemamodellierungsprogramm
 - Data Warehouse-Zentrale 279
- Schemennamen
 - Zeichenbegrenzungen 66
- SEARCH, Discovery-Modus 492
- sequence invocation 382
- sequences
 - invoking 382
 - nextval-expression 382
 - prevval-expression 382
- sequences, description 355
- Sequenzen 77
 - ändern 79
 - erstellen 75
 - löschen 79
 - Zugriffsrechte 80
- SET CONSTRAINTS, Anweisung
 - ersetzt durch SET INTEGRITY 67
- SET ENCRYPTION PASSWORD 81
- SET ENCRYPTION PASSWORD statement
 - detailed description 446
- SET INTEGRITY, Anweisung
 - ersetzt SET CONSTRAINTS 67
- SET Variable statement
 - detailed description 447
- Sicherung
 - auf benannte Pipes 87
 - offline
 - von geteiltem Spiegelimage 87
 - Teil- 84
- Sicherungsservice-APIs
 - DB2-Unterstützung 500
- Sichten
 - nicht verfügbar nach Migration 17
- SIGNAL statement 458
- SIGTTIN 102
- Simplified Chinese
 - locale setting on Linux 7
- SMS-Tabellenbereiche
 - Behälter hinzufügen 78
- SNA Server
 - von Host- oder AS/400-Anwendungen auf UDB zugreifen 497
- SNA SPM
 - Start nach dem Neustart prüfen 497
- softmax 119
- Softwarevoraussetzungen, Befehls-sprachendatei von ERwin, Data Warehouse-Zentrale, Information Catalog Manager, ICM 232
- Solaris
 - erforderliche Programmkorrekturstufe bei Version 2.6 4
 - fehlende Unterstützung für Versionen vor UltraSparc 4
- Solaris-Betriebssystem
 - 64-Bit
 - Einschränkungen 187
 - Konfiguration 187
- Solaris-Betriebsumgebung
 - Onlineinformationen suchen 11
- Sortierspeichergroße 116
- Sortierspeicherschwellexwert 116
- Speicherbereichsgröße
 - in "Assistent: Datenbank erstellen" einstellen 219
- Speicherfenster
 - für HP-UX 11 492
- Sperrliste
 - maxlocks, Konfigurationsparameter 117

- Spiegeln
 - Protokolle 90
- SQL, compound 368
- SQL procedure
 - DECLARE statement 425
 - dynamic compound statement 425
 - FOR statement 452
 - RETURN statement 456
 - SIGNAL statement 458
 - variables 425
- SQL statements
 - ALTER SEQUENCE 418
 - CREATE SEQUENCE 434
 - GRANT (Sequence Privileges) 443
 - SET ENCRYPTION PASSWORD 446
 - SET Variable 447
- SQL variables 425
- SQLBindFileToParam CLI function, correction 369
- SQLNextResult function 370
- starting DB2
 - using db2start on Windows 95, 98 and ME 183
- static profiling, in JDBC/ODBC/CLI applications 365
- stdin 102
- Steuerzentrale
 - Anzeigeproblem der Kurzhilfe bei OS/2 213
 - auf der OS/390-Plattform 210
 - bidirektionale Unterstützung bei Windows NT 209
 - Einschränkungen bei gleichzeitig ablaufenden Applets 214
 - erforderlicher Fix für OS/390 212
 - Fehlerbehebung 213
 - Fehlerbehebung bei UNIX-Plattformen 213
 - Probleme bei Internet Explorer 7
 - Probleme der Prozedurzentrale bei Windows 95 215
 - Tool - Einstellungen, Notizbuch 214
 - übersetzte Versionen ausführen 69
 - Version 6, using to administer satellite environment 139
 - zur Verwaltung von VM- und VSE-Datenbanken 209
- string
 - expression 475
 - operand 475
- Subelementstatistik haben: 108
- Suchen von Onlineinformationen in der Solaris-Betriebsumgebung 11
- Sun Cluster 2.2
 - Vorbedingungen für DB2 Connect 64
- Sun Solaris und HP
 - Unterstützung von Sichern und Wiederherstellen 91
- SWITCH ONLINE, Klausel 80
- Sybase
 - Inkompatibilität zu ICM unter Windows 8
- Sybase-Datenquellen 44
 - Codepages 50
 - Leistung verbessern 47
 - mit DB2 verknüpfen 46
 - Umgebungsvariablen 44

Sybase Open Client 44
Systeme zusammenschlossener Datenbanken
 Einschränkungen 39
 Kurznamen 39
 Oberflächen 39
Systeme zusammenschlossener Datenbanken, Einschränkungen 496
Systemkatalogstatistik
 Verteilungsstatistik sammeln 107

T

Tabellenbereiche
 auf Status ONLINE einstellen 80
Tabellenstrukturen 179
Teilsicherung und -wiederherstellung 84
template
 Commit.tag
 example values 282
 token 282
 ForeignKey.tag
 example values 284
 tokens 282
 ForeignKeyAdditional.tag
 example values 286
 tokens 284
 PrimaryKey.tag
 example values 287
 tokens 286
 PrimaryKeyAdditional.tag
 example values 289
 tokens 288
Testverbindung für Aktualisierung auf mehreren Systemen
 Inkompatibilität der Exemplarversionen 210
triggered SQL statements
 SET Variable statement 447
triggers
 error messages 440
 inoperative 439

U

Unicode
 Datenbank und Anwendungen 469
 Dokumentationsaktualisierungen 471
 functions in 475
 in ODBC applications 481
Unicode-Clients
 Einschränkungen beim Versetzen von Daten 482
Unterstützung nationaler Zeichensätze 71
Unterstützung von OLE DB 255
USAGE, Zugriffsrecht 80

V

Veritas
 High Availability unter Solaris 64
Verknüpfungsfehler 210
Verschlüsselung
 Daten 80

Versetzen von Daten
 Beschränkungen von Unicode-Clients 482
Verteilte Datenbanken
 für die Aktualisierung einrichten 63
VI
 zur Verwendung mit DB2 einrichten 102

W

Web-Browser
 Empfehlungen für Windows 2000 11
WEEK_ISO scalar function 372
Wiederherstellen
 Probleme mit DATALINK-Spalten 495
Wiederherstellung
 Interaktion mit DB2 Data Links Manager 95
 parallel 86
 Teil- 84
Wiederherstellung nach einem Systemabsturz
 mit dem Tool "db2inidb" 82
Windows
 Inkompatibilität zwischen Sybase und ICM 8
Windows 2000
 empfohlene Web-Browser 11
 IPX/SPX-Protokollunterstützung 17
 migrieren unter 17

X

XML-Unterstützung
 für Common Warehouse Metamodel (CWM) 278

Z

Zugriff auf mehrere Indizes 113
Zugriffsmöglichkeiten 3, 498
Zugriffsrechte
 USAGE 80
Zurückgestellte Ein-/Ausgabe
 Unterstützung ständiger Verfügbarkeit 82

IBM