

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter Anhang B, „Bemerkungen“, auf Seite 175 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Connect Version 9.5 DB2 Connect User's Guide,
IBM Form SC23-5841-01,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1993, 2008
© Copyright IBM Deutschland GmbH 2008

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

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

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
März 2008

Inhaltsverzeichnis

Zu diesem Handbuch v

Teil 1. DB2 Connect - Konzepte 1

Kapitel 1. DB2 Connect 3

DB2 Connect - Produktangebot 3
Funktionen in Version 9 und vorhergehenden
Releases 3
Hostdatenbanken 6
DB2 Connect und SQL-Anweisungen 6
DB2 Connect-Verwaltungsdienstprogramme 7
WebSphere Federation Server und DB2 Connect 8

Kapitel 2. Distributed Relational Data- base Architecture (DRDA) 9

DRDA und Datenzugriff 9
DB2 Connect und DRDA 10
RUOW (Remote Unit of Work) 11
Verteilte Anforderungen 12

Kapitel 3. DB2 Connect-Szenarios 15

Direktzugriff auf Hostdatenbanken 15
Zugreifen auf Host- oder System i-DB2-Daten mit
DB2 Connect Personal Edition 17
DB2 Connect-Serverprodukte als Konnektivitätsser-
ver 18
DB2 Connect und Webanwendungen 19
DB2 Connect und IBM WebSphere 21
DB2 Connect als Java-Anwendungsserver 22
DB2 Connect auf dem Web-Server 23
DB2 Connect und Anwendungsserver 23
DB2 Connect und Transaktionsverarbeitungsmoni-
tore 27

Teil 2. DB2 Connect - Referenz- informationen 31

Kapitel 4. Aktualisieren von Datenbank- verzeichnissen 33

Werte des Systemdatenbankverzeichnisses 33
Werte des Knotenverzeichnisses 34
Werte des DCS-Verzeichnisses 35
Arbeitsblatt zur Verzeichnisanpassung 40
Definieren von mehreren Einträgen für dieselbe
Datenbank 41
Handhabung von BIDI-Daten 41

Kapitel 5. DB2 Connect - Sicherheit 45

Gesicherte Verbindungen über DB2 Connect 45
Erstellen und Beenden einer gesicherten Verbin-
dung über CLI 46

Wechseln von Benutzern über eine gesicherte
Verbindung mit CLI 48
Überlegungen zur DB2-Authentifizierung 50
Kerberos-Unterstützung 51
Hinweise und Tipps zur Sicherheit unter OS/390
und z/OS 52
Mit DB2 Connect unterstützte Sicherheitsein-
stufungen 53

Kapitel 6. Binden von Anwendungen und Dienstprogrammen (DB2 Connect) . 55

Kapitel 7. Aktualisierungen auf mehre- ren Systemen 59

Aktivieren von Aktualisierungen auf mehreren Sys-
temen über die Steuerzentrale 60
Testen der Aktualisierung auf mehreren Systemen
über die Steuerzentrale 60
Aktualisierungen auf mehreren Systemen und
Synchronisationspunktmanager 61
Konfiguration von DB2 Connect mit einem XA-kom-
patiblen Transaktionsmanager 62
DB2 Connect-Unterstützung für lose gekoppelte
Transaktionen 63

Kapitel 8. Versetzen von Daten mit DB2 Connect 65

Kapitel 9. SQLCODE-Zuordnung 69

Inaktivieren der SQLCODE-Zuordnung 69
Anpassung der SQLCODE-Zuordnung 69

Kapitel 10. Datenbanksystem- überwachung und DB2 Connect 75

Überwachung von Verbindungen für ferne Clients 75
Überwachung der Leistung mit Windows Perfor-
mance Monitor 75
Verwendung der GET SNAPSHOT-Befehle 76
DCS-Anwendungsstatus 78
Diagnosemonitor und Alerts 83
DB2 für z/OS-Diagnosemonitor - Übersicht 83
Starten, Stoppen und Aktualisieren des DB2 für
z/OS-Diagnosemonitors 84
Anzeigen, Übergeben und Speichern empfohlener
Aktionen 85
Anzeigen von Diagnosealertzusammenfassungen 88
Anzeigen von Diagnosealertobjekten 89

Teil 3. Hohe Verfügbarkeit und DB2 Connect 91

Kapitel 11. Hohe Verfügbarkeit und Lastausgleich für die Konnektivität von Hostdatenbanken 93

Kapitel 12. Automatische Clientweiterleitung - Beschreibung und Einrichtung (DB2 Connect) 95

Kapitel 13. Konfigurieren der automatischen Clientweiterleitung für die Distributions-technologie für die Clientverbindung 97

Teil 4. Optimierung und DB2 Connect 99

**Kapitel 14. DB2 Connect - Leistungs-
informationen 101**

**Kapitel 15. Optimierung des ODBC-
Zugriffs 105**

Kapitel 16. Anwendungsentwurf . . . 107

Kapitel 17. Verbindungsverwaltung 111
Verbindungspooling 111
Verbindungskonzentrator 113
Verbindungspooling und Verbindungskonzentrator 118
Mit WebSphere MQ Transaction Manager und DB2
für OS/390 erforderlicher Verbindungskonzentrator 119

**Kapitel 18. DB2 Connect - SYSPLEX-
Unterstützung 121**
Überlegungen zur SYSPLEX-Ausnutzung auf
OS/390- und zSeries-Systemen 121
DB2-SYSPLEX-Ausnutzung. 122
Konfigurationsvoraussetzungen für SYSPLEX-
Systeme 123

Kapitel 19. DB2 Connect-Optimierung 125
Optimierung der Hostdatenbank 127
Überlegungen zur Netzwerkoptimierung 127
Konflikte bei Systemressourcen 129
DB2 Connect-Fehlerbehebung 129
Optimierung von DB2 für OS/390 und z/OS. . . 130
Erhöhung der Datenübertragungsgeschwindigkeit
unter DB2 Connect 130
Zusätzlicher Abfrageblock 130
Fensterskalierung von RFC-1323 132
Hostdatenkonvertierung. 133
Datentypen für Zeichendaten 133
Netzwerkhardware 133

**Kapitel 20. Optimierung der CLI/OD-
BC-Anwendungsleistung 135**

Teil 5. Fehlerbehebung 137

Kapitel 21. Fehlerbehebung 139
Zusammenstellen relevanter Informationen . . . 139
Nicht erfolgreiche einleitende Verbindung. . . 139
Probleme nach dem Herstellen einer einleitenden
Verbindung 140
Diagnosetools 142

**Kapitel 22. DB2-Traces in DB2 Con-
nect 143**
Abrufen eines DB2-Trace mit 'db2trc' 143
Erstellen eines Speicherauszugs einer DB2-Traceda-
tei 144
Formatieren einer DB2-Tracedatei. 145

Kapitel 23. DRDA-Tracedateien 147
Dienstprogramm für Trace 147
Traceausgabe 148
Analyse der Traceausgabedatei 149
Beispiele für die Traceausgabedatei 150
Informationen zu nachfolgenden Puffern für
DRDA-Traces 155

Teil 6. Nachrichten 157

**Kapitel 24. DB2 Connect - Häufige
Probleme 159**

Teil 7. Anhänge und Schlussteil 163

**Anhang A. Übersicht über die techni-
schen Informationen zu DB2. 165**
Bibliothek mit technischen Informationen zu DB2
im Hardcopy- oder PDF-Format 166
Bestellen gedruckter DB2-Bücher 168
Aufrufen der Hilfe für den SQL-Status über den
Befehlszeilenprozessor 169
Zugriff auf verschiedene Versionen der DB2-
Informationszentrale 169
Anzeigen von Themen in der gewünschten Sprache
in der DB2-Informationszentrale 169
Aktualisieren der auf Ihrem Computer oder Intra-
net-Server installierten DB2-Informationszentrale . 170
DB2-Lernprogramme 172
Informationen zur Fehlerbehebung in DB2 . . . 173
Bedingungen 173

Anhang B. Bemerkungen 175

Index 179

Zu diesem Handbuch

Das *DB2 Connect-Benutzerhandbuch* enthält alle Informationen, die Sie benötigen, um DB2 Connect kennenzulernen und zu verwenden. Die DB2 Connect-Konzepte werden mit den typischen Szenarios dargestellt, die die Beziehungen zwischen DB2 Connect und den anderen Komponenten der Netzumgebung verdeutlichen. Darüber hinaus werden Aspekte der Datenbankverzeichnisse, der Sicherheit zwischen Systemen, der Aktualisierungen auf mehreren Systemen, des Versetzens von Daten und der Überwachung von DB2 Connect erläutert. Außerdem wird die von DB2 Connect bereitgestellte Unterstützung der Hochverfügbarkeit in Ihrer Netzumgebung dargestellt. Sie erhalten eine Einführung in die Sicherstellung guter Leistungswerte sowohl von DB2 Connect als auch im gesamten Netz. Das Handbuch enthält darüber hinaus eine Reihe von Themen zur Fehlerbehebung bei möglichen Problemen.

Zielgruppe

Für Systemadministratoren, Datenbankadministratoren und Fachleute für die Datenübertragung sind alle oder ausgewählte Teile dieses Handbuchs von Interesse.

Teil 1. DB2 Connect - Konzepte

Kapitel 1. DB2 Connect

DB2 Connect bietet schnelle und zuverlässige Konnektivität zu Host- und System i-Datenbanken für e-business- und andere Anwendungen, die unter den Betriebssystemen Linux, UNIX und Windows ausgeführt werden.

DB2 Connect Personal Edition bietet direkte Konnektivität zu Host- und System i-DB2-Servern, während DB2 Connect-Serverprodukte indirekte Konnektivität zur Verfügung stellen, die Clients den Zugriff auf Host- und System i-DB2-Server über das DB2 Connect-Gateway ermöglicht. Sie können aus einer Reihe von DB2 Connect-Serverprodukten mit unterschiedlichem Produktumfang und unterschiedlichen Lizenzierungslösungen das Produkt auswählen, das für Ihre Umgebung geeignet ist.

DB2 Connect - Produktangebot

DB2 Connect verfügt über mehrere Verbindungslösungen, einschließlich DB2 Connect Personal Edition und einer Reihe von DB2 Connect-Serverprodukten:

- DB2 Connect Enterprise Edition
- DB2 Connect Application Server Edition
- DB2 Connect Unlimited Edition für zSeries
- DB2 Connect Unlimited Edition für i5/OS

Ausführliche Informationen zum DB2 Connect-Produktangebot finden Sie unter <http://www.ibm.com/support/docview.wss?rs=73&uid=swg21219983>.

Funktionen in Version 9 und vorhergehenden Releases

Dieser Abschnitt bietet eine Zusammenfassung der Erweiterungen für die einzelnen Versionen und Releases.

Funktionen von DB2 Connect Version 9

DB2 Connect Version 9 umfasst die folgenden Erweiterungen:

- Clientunterstützung für gesicherte Verbindungen

Ein Client kann mittels ODBC, XA oder neuer Java-Methoden gesicherte Verbindungen zu Datenbankservern aufbauen, die gesicherte Kontexte unterstützen (zum Zeitpunkt der Erstellung dieses Handbuchs nur DB2 für z/OS). Dann kann der Client seinen Benutzernamen wechseln, ohne dass eine vollständige Authentifizierung des neuen Namens durchgeführt werden muss.

- Unterstützung für die Datentypen BINARY, VARBINARY und DECFLOAT

DB2 für z/OS unterstützt jetzt die Datentypen BINARY, VARBINARY und DECFLOAT. Die DB2-Befehlszeilenschnittstelle (CLI) und IBM Data Server Provider for .NET wurden um Unterstützung für diese Datentypen ergänzt. Wenn Sie Anwendungen einsetzen, die über DB2 Connect auf DB2 für z/OS zugreifen, können diese die DB2-CLI und IBM Data Server Provider for .NET verwenden, um die neuen Datentypen zu nutzen. Mit der neuen Verbindungseinstellung `SQL_ATTR_DECFLOAT_`

ROUNDING_MODE können Clients angeben, wie dezimale Gleitkommazahlen gegebenenfalls auf der Serverseite gerundet werden sollen.

- Die Kommunikationsprotokolle NetBIOS und SNA werden nicht mehr unterstützt

Kunden, die diese Protokolle verwenden, müssen ihre Knoten und Datenbanken mit einem unterstützten Protokoll (z. B. TCP/IP) neu katalogisieren.

- Neue Unterstützung für das Kommunikationsprotokoll IPv6

Internet Protocol Version 6 (IPv6) wird jetzt unterstützt. Damit sind jetzt Serververbindungen über IPv4- und IPv6-Adressen möglich.

- Beschränkung auf 64 KB für CLP-SQL-Anweisungen wurde aufgehoben

Der neue Befehlszeilenprozessor (CLP) hat eine Obergrenze von etwa 2 MB für SQL-Anweisungen und CLP-Befehle, die SQL-Anweisungen enthalten. Dies entspricht dem Wert der anderen DB2-Tools. Alle Anwendungen, die DB2 Connect verwenden, können diese neue Obergrenze nutzen.

- IBM Data Server Provider for .NET-Erweiterungen, einschließlich Unterstützung für .NET Framework 2.0

Diese Unterstützung und die Erweiterungen ermöglichen Ihnen die Entwicklung sehr leistungsfähiger .NET-Anwendungen für DB2 Connect. Die neue Funktionalität ermöglicht zum Beispiel Folgendes:

- Anwendungen können eine bestimmte Gruppe von Zeilen abrufen und müssen nicht durch die gesamte Ergebnismenge blättern.
- Anwendungen können größere Datenmengen mit einer Bulk-Operation kopieren.
- Anwendungen können festlegen, dass eine bestimmte Anzahl von SQL-Anweisungen gesammelt und dann als Batch an den DB2-Datenbankserver übergeben wird. Dadurch reduziert sich die Anzahl der Datenübertragungen zwischen der Clientanwendung und dem Datenbankserver.

- Zweiphasiges Commit für Datenquellen von unterschiedlichen Herstellern bei Verwendung von WebSphere Federation Server

DB2 Connect-Anwendungen können mit WebSphere Federation Server auf Datenquellen von IBM und vielen anderen Herstellern zugreifen.

- Unterstützung von Verbindungszeitlimits für Datenbankanwendungen

Sie können die Zeit, die Ihre DB2 Connect-Datenbankanwendungen auf eine Verbindung warten, beschränken. Diese Funktion ist besonders dann nützlich, wenn der Zieldatenbankserver nicht erreichbar ist.

- Vereinfachtes Upgrade von DB2 Connect Personal Edition

Sie können auf den Betriebssystemen Windows und Linux ein Upgrade von DB2 Connect Personal Edition durchführen, indem Sie die entsprechende elektronische Zertifikatsdatei verwenden. Für das Upgrade ist keine vollständige Installation mehr erforderlich.

- Änderungen am Lizenzsupport für DB2

Die Änderungen am DB2 Connect-Produktpaket sind Teil der Verbesserungen der Lizenzzentrale und des Lizenzverwaltungstools (Befehl db2licm).

Funktionen in DB2 Connect Version 8 Release 2

DB2 Connect Version 8.2 umfasste die folgenden funktionalen Erweiterungen:

- Automatische Clientweiterleitung
Wenn eine TCP/IP-Verbindung zu einem Server oder DB2 Connect-Server unterbrochen wird, versucht der Client automatisch, die Verbindung erneut herzustellen, falls ein Alternativserver vorhanden ist. Der Alternativserver wird in der Serverinstanz angegeben, und seine Adresse wird während der Verbindung an den Client gesendet.
- Datenverschlüsselung
Die Client/Server-Kommunikation bietet nun die Verschlüsselung von Benutzerdaten bei deren Übertragung im Netzwerk.

Funktionen in DB2 Connect Version 8 Release 1 (einschließlich aller FixPaks und Modifikationsstufen)

DB2 Connect Version 8.1 umfasste die folgenden funktionalen Erweiterungen:

- Unterstützung für längere SQL-Anweisungen (bis zu 2 MB)
SQL-Anweisungen mit einer Größe von maximal 2 MB können von CLI- und JDBC-Anwendungen verarbeitet werden. Der Grenzwert für die eingebettete Schnittstelle beträgt jedoch nach wie vor 64 KB.
- Diagnoseinformationen, die den Ursprung einer SQL-Anweisung identifizieren
Bietet die Möglichkeit, festzustellen, welches Anwendungsprogramm eine bestimmte Anweisung an den Cache für dynamische SQL-Anweisungen von DB2 für z/OS übergeben hat.
- Spaltenorientierter Eingabearray
Ermöglicht es Anwendungen, mehrere Parametersätze für eine einzelne SQL-Anweisung zur Verfügung zu stellen.
- Überwachen der Netzwerkzeit
Neue Monitorelemente werden dazu verwendet, die Datenbankaktivitäten und die Datenübertragung im Netzwerk auf Datenbank- oder Anwendungsebene besser zu dokumentieren.
- DB2 CLI-Unterstützung für dynamische verschiebbare Cursor
In DB2 CLI werden nun dynamische verschiebbare Cursor beim Zugriff auf Server unterstützt, auf denen DB2 Universal Database (UDB) für z/OS ab Version 8.1 ausgeführt wird.
- eWLM-Unterstützung
Bietet die Möglichkeit, End-to-End-Arbeitseinheiten über Middlewaregruppen zu überwachen, um Engpässe festzustellen.
- Erweiterung des DB2-Befehls ping
Der DB2-Befehl ping unterstützt nun die Angabe einer Anfrage- und Antwortpaketgröße.

Anmerkung: DB2 Connect unterstützt den PING-Befehl nicht, wenn dieser von einem Version 7-Client über ein Version 9-Gateway für den Host abgesetzt wird.

Funktionen in DB2 Connect Version 7 Release 2

DB2 Connect Version 7.2 umfasste die folgenden funktionalen Erweiterungen:

- Verbesserte Unterstützung für Microsoft Transaction Server (MTS) und COM+-Technologien
- DB2 Connect Web Starter Kit
- DB2 Connect für Linux unter S/390

Funktionen in DB2 Connect Version 7 Release 1

DB2 Connect Version 7.1 umfasste die folgenden funktionalen Erweiterungen:

- XA-Konzentrator
- Verbesserungen für die Aktualisierung auf mehreren Systemen

Hostdatenbanken

Der Begriff *Datenbank* bezeichnet in diesem Handbuch ein Verwaltungssystem für relationale Datenbanken (RDBMS). Bei anderen Systemen, mit denen DB2 Connect kommuniziert, wird mit dem Begriff 'Datenbank' möglicherweise ein leicht unterschiedliches Konzept bezeichnet. Der DB2 Connect-Terminus 'Datenbank' kann sich auch auf Folgendes beziehen:

OS/390 oder z/OS

DB2 Universal Database (UDB) für OS/390 und z/OS Version 7 oder DB2 UDB für z/OS Version 8. Ein durch seinen LOCATION NAME festgelegtes DB2 Universal Database für z/OS und OS/390-Subsystem. Der LOCATION NAME kann durch Anmeldung bei TSO und Ausgabe der folgenden SQL-Abfrage über eines der verfügbaren Abfrage-Tools ermittelt werden:

```
select current server from sysibm.sysdummy1
```

Der LOCATION NAME ist auch in der BSDS (Boot Strap Data Set) definiert und wird in der Nachricht DSNL004I (LOCATION=standort) angegeben, die beim Starten von DDF (Distributed Data Facility) erstellt wird. LOCATION NAME unterstützt bis zu 8 Standortaliasnamen und ermöglicht es Anwendungen damit, für den Zugriff auf einen z/OS Version 8-Server verschiedene Datenbankaliasnamen zu verwenden. Mit dem z/OS-Befehl `-display ddf` können Sie den Standortnamen, den Domänennamen, die IP-Adresse und den Port des DB2-Servers abrufen.

VSE DB2 for VSE, das in einer Datenbankpartition ausgeführt wird, die über ihren DBNAME (Datenbanknamen) angegeben wird.

VM DB2 for VM, das auf einer virtuellen CMS-Maschine läuft, die über ihren DBNAME angegeben wird.

OS/400

DB2 für i5/OS, einen integralen Bestandteil des Betriebssystems OS/400. Auf einem System i-Server kann nur eine einzige Datenbank vorhanden sein, es sei denn, das System ist für die Verwendung unabhängiger Zusatzspeicherpools konfiguriert.

DB2 Connect und SQL-Anweisungen

DB2 Connect leitet SQL-Anweisungen, die von Anwendungsprogrammen übergeben wurden, an Host- oder System i-Datenbankserver weiter.

DB2 Connect kann fast alle gültigen SQL-Anweisungen und die unterstützten DB2-Anwendungsprogrammierschnittstellen (APIs) weiterleiten:

- JDBC
- SQLJ
- ADO.NET
- OLE-DB
- ODBC

- Perl
- PHP
- DB2-CLI
- Eingebettetes SQL

Unterstützung für eingebettetes SQL

Es gibt zwei Arten der Verarbeitung für eingebettetes SQL: statisches SQL und dynamisches SQL. Statisches SQL verringert durch Verarbeitung im Voraus die zur Ausführung einer SQL-Anweisung erforderliche Zeit. Dynamisches SQL wird verarbeitet, wenn die SQL-Anweisung an den Host- oder System i-Datenbankserver übergeben wird. Dynamisches SQL ist flexibler, aber möglicherweise langsamer. Die Entscheidung, ob statisches oder dynamisches SQL verwendet werden soll, wird vom Anwendungsprogrammierer getroffen. Beide Arten werden von DB2 Connect unterstützt.

Verschiedene Host- oder System i-Datenbankserver implementieren SQL unterschiedlich. DB2 Connect bietet eine vollständige Unterstützung der allgemeinen IBM SQL-Implementierung sowie der SQL-Implementierungen von DB2 für OS/390 und z/OS, DB2 Server for VSE & VM (früher SQL/DS) sowie DB2 für System i. Für die Erhaltung der Datenbankunabhängigkeit wird die Verwendung von IBM SQL dringend empfohlen.

DB2 Connect-Verwaltungsdienstprogramme

Folgende Dienstprogramme sind zur Unterstützung von DB2 Connect-Administratoren verfügbar:

- Der Befehlszeilenprozessor (CLP) ermöglicht das Absetzen von SQL-Anweisungen für eine Datenbank des Host- oder System i-Datenbankservers. Er leitet die SQL-Anweisungen an die angegebene Datenbank weiter.
- Die DB2-Befehlszentrale bietet eine Grafikschnittstelle zum Befehlszeilenprozessor (CLP).
- Import- und Exportdienstprogramme ermöglichen das Laden, Importieren und Exportieren von Daten aus einer und in eine Datei auf einer Workstation sowie aus einer und in eine Datenbank des Host- oder System i-Datenbankservers. Diese Dateien können anschließend für das Importieren von Daten in Datenbankenprogramme, Tabellenkalkulationsprogramme und andere Anwendungen verwendet werden, die auf Ihrer Workstation ausgeführt werden.
- Wenn Sie ein DB2 Connect-Serverprodukt einsetzen, können Sie Event Viewer und Performance Monitor verwenden. Mit Event Viewer können Sie von DB2 Connect protokollierte Ausnahmeereignisse anzeigen. Mit Performance Monitor können Sie die Leistung von DB2 Connect-Servern entweder lokal oder fern überwachen und verwalten.
- Mit der DB2-Steuerzentrale können Sie alle Aspekte von DB2 Connect-Servern verwalten und überwachen. Außerdem können Administratoren mit Datenbankobjekten von DB2 für OS/390 oder z/OS arbeiten (beispielsweise Tabellen, Sichten, Pufferpools und Threads).
- Außerdem ermöglicht das Dienstprogramm des Datenbanksystemmonitors dem Systemadministrator das Überwachen von Systemverbindungen. Diese Funktion steht nur dann zur Verfügung, wenn DB2 Connect als Server fungiert. Darüber hinaus wird das Ermitteln von Fehlerursachen durch den Systemadministrator

vereinfacht. Der Systemadministrator kann Client-Anwendungen mit den entsprechenden Jobs korrelieren, die auf dem Host- oder System i-Datenbankserver ausgeführt werden.

Anmerkung: In Vorgängerreleases wurden die grafischen DB2-Verwaltungstools, wie z. B. die Steuerzentrale, auf allen Plattformen unterstützt. Bei Version 9 werden die grafischen DB2-Verwaltungstools nur unter Windows x86, Windows x64 (AMD64/EM64T), Linux (x86) und Linux (AMD64/EM64T) unterstützt. Sie können den DB2-Befehlszeilenprozessor (CLP) auf allen Plattformen für Verwaltungsaufgaben verwenden.

WebSphere Federation Server und DB2 Connect

WebSphere Federation Server ist ein separates Produktangebot für den Zugriff auf und die Integration von Daten aus Datenquellen verschiedener Hersteller. DB2 Connect ermöglicht Ihnen die Nutzung der großen Datenmengen, die sich auf bestehenden Host- und Mid-Range-Servern befinden.

WebSphere Federation Server unterstützt Sie bei der Integration von Informationen, indem dieses Produkt das Sichten und Bearbeiten einer Gruppe von Datenquellen wie eine einzige Quelle ermöglicht. Der Zugriff auf die Datenquelle ist für die aufrufende Anwendung vollständig transparent. WebSphere Federation Server kann mit DB2 Connect-Serverprodukten verwendet werden. WebSphere Federation Server stellt nativen Schreib- und Lesezugriff auf die Datenbanken der DB2-Produktfamilie sowie auf die Datenbanken von Informix, Oracle, Sybase, Teradata und Microsoft SQL Server zur Verfügung. Darüber hinaus bietet WebSphere Federation Server auch Lesezugriff auf nicht relationale und biowissenschaftliche Datenquellen wie BLAST, Documentum, Entrez, IBM Lotus Extended Search, Dateien mit Tabellenstruktur und XML. Sie können das Produkt zur Formulierung von Abfragen für Daten in einem föderierten System verwenden.

Kapitel 2. Distributed Relational Database Architecture (DRDA)

Distributed Relational Database Architecture (DRDA, Architektur einer verteilten relationalen Datenbank) ist eine Gruppe von Protokollen, die mehreren Datenbanksystemen (von IBM und anderen Herstellern) und Anwendungsprogrammen die Zusammenarbeit ermöglicht. Alle Kombinationen von Produkten zur Verwaltung relationaler Datenbanken, die DRDA verwenden, können zu einem verteilten Verwaltungssystem für relationale Datenbanken verbunden werden. DRDA koordiniert die Kommunikation zwischen Systemen, indem diese Architektur definiert, was ausgetauscht werden muss und wie dieser Austausch erfolgen soll.

UOW (Unit of Work, Arbeitseinheit)

Eine UOW (*Unit of Work, Arbeitseinheit*) stellt eine einzelne logische Transaktion dar. Sie besteht aus einer Folge von SQL-Anweisungen, bei der entweder alle Operationen erfolgreich ausgeführt werden oder die Folge als Ganzes als nicht erfolgreich angesehen wird.

DUOW (Distributed Unit of Work, verteilte Arbeitseinheit)

Eine DUOW (*Distributed Unit of Work, verteilte Arbeitseinheit*), auch als Aktualisierung auf mehreren Systemen bezeichnet, umfasst mehr als einen Datenbankserver innerhalb einer UOW. Eine DUOW weist folgende Merkmale auf:

- Pro UOW werden mehrere Datenbankverwaltungsserver aktualisiert.
- Die Anwendung steuert die Arbeitsverteilung und leitet das Commit ein.
- Pro UOW kann es Mehrfachanforderungen geben.
- Pro Anforderung gibt es einen Datenbankverwaltungsserver.
- Das Commit wird auf mehreren Datenbankservern koordiniert.

DRDA und Datenzugriff

DRDA definiert zwar Datenbankkommunikationsprotokolle, nicht jedoch die Programmierungsschnittstellen (APIs), die von den Anwendungsprogrammierern verwendet werden sollten. Im Allgemeinen kann DRDA von einem Anwendungsprogramm verwendet werden, um Anforderungen zu übergeben, die ein Ziel-DRDA-Server ausführen kann. Alle heutzutage verfügbaren DRDA-Server können SQL-Anforderungen ausführen, die von einem Anwendungsprogramm über DB2 Connect weitergeleitet wurden.

IBM bietet Anwendungsprogrammierern Tools zum Generieren von SQL-Anforderungen für Windows-, UNIX- und Linux-Betriebssysteme. Diese Tools sind Bestandteil des DB2-Clients. Der DB2-Datenbankmanager unterstützt mehrere Programmierschnittstellen: ADO.NET, JDBC, SQLJ, PHP, Perl DBI, eingebettetes SQL, DB2 Call Level Interface (DB2 Call Level Interface) und OLE DB. Diese APIs können von Programmierern zum Erstellen von Anwendungen in einer Vielzahl von Programmiersprachen verwendet werden.

DB2 Connect und DRDA

DB2 Connect implementiert die DRDA-Architektur, um den Systemaufwand und die Komplexität beim Zugriff auf Daten zu verringern, die auf Datenbankservern unter DB2 Universal Database (UDB) für iSeries, DB2 für System i, DB2 UDB für OS/390 und z/OS, DB2 für z/OS, DB2 Server for VSE & VM sowie auf anderen dem DRDA-Standard entsprechenden Datenbankservern gespeichert sind. Durch vollständiges Ausschöpfen der DRDA-Architektur bietet DB2 Connect eine leistungsfähige, preisgünstige Lösung mit den Systemverwaltungsfunktionen, wie sie Kunden verlangen.

In der DRDA-Terminologie ist ein *Anwendungsrequester (AR)* der Code, der die Anwendungsseite einer verteilten Verbindung verwaltet. Der AR ist die Anwendung, die Daten anfordert. DB2 Connect dient als Anwendungsrequester für Anwendungsprogramme, die sich lokal auf der DB2 Connect-Workstation oder auf einem separaten fernen Client von DB2 Connect befinden.

Ein *Anwendungsserver (AS)* ist der Code, der die Datenbankseite der Verbindung verwaltet.

DRDA unterstützt außerdem mehrschichtige Verbindungen zwischen einem Anwendungsrequester und einem Server. In dieser Topologie wird der Server, zu dem ein Anwendungsrequester eine Verbindung herstellt, als Anwendungsserver bezeichnet. Alle weiteren, nachgeschalteten Server werden dagegen als Datenbankserver (DS) bezeichnet, da sie nicht direkt mit dem Anwendungsrequester interagieren. Darüber hinaus wird jeder Anwendungsserver oder Datenbankserver, der sich zwischen einem Anwendungsrequester und dem letzten Datenbankserver befindet, auch als zwischengeschalteter Server (Intermediate) bezeichnet. Damit wird hervorgehoben, dass es sich bei diesem Server weder um das System handelt, das die Datenbankanforderung initiiert, noch um das System, das die der Anforderung entsprechende Funktion ausführt. Die Verwendung von Datenbankservern und zwischengeschalteten Servern wird von DB2 Connect unterstützt.

Abb. 1 zeigt den Datenfluss zwischen der DB2 Connect-Workstation und dem Host- bzw. System i-Server in einer Umgebung, in der nur lokale Clients existieren.

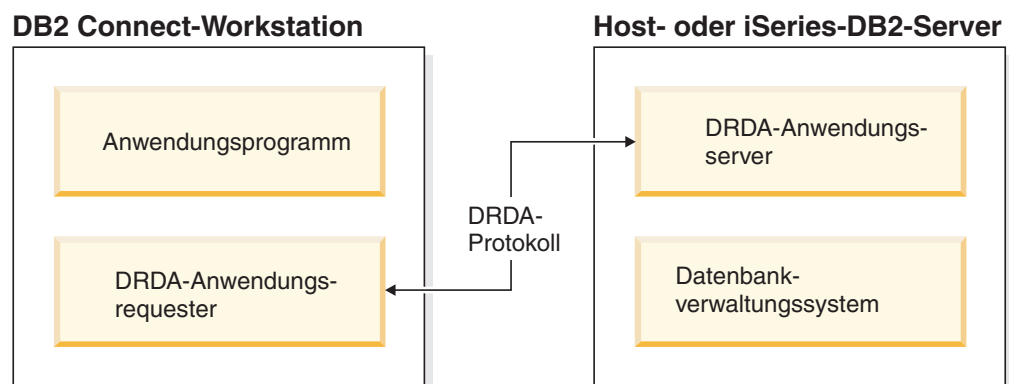


Abbildung 1. Datenfluss zwischen einem DB2 Connect-Server und einem Host- oder System i-Server

Zur Implementierung der Verbindungen zwischen Datenbankverwaltungssystemen auf DRDA-Servern und IBM Data Server-Clients verwendet DRDA folgende Architekturen:

- CDRA (Character Data Representation Architecture)
- DDMA (Distributed Data Management Architecture)
- FD:OCA (Formatted Data Object Content Architecture)
- TCP/IP (Transmission Control Protocol/Internet Protocol)

Diese Architekturen werden als Blöcke für die Erstellung verwendet. Die Datenströme, die über das Netzwerk fließen, werden von der DRDA-Architektur bestimmt, die ein Datenstromprotokoll angibt, das den Zugriff auf verteilte relationale Datenbanken unterstützt.

Eine Anforderung wird über Verzeichnisse, die verschiedene Typen von Kommunikationsinformationen und den Namen der DRDA-Serverdatenbank enthalten, auf die zugegriffen wird, an die richtige Zieladresse weitergeleitet.

RUOW (Remote Unit of Work)

Eine *RUOW* (Remote Unit of Work, ferne Arbeitseinheit) ermöglicht dem Benutzer oder Anwendungsprogramm das Lesen oder Aktualisieren von Daten an jeweils einer Speicherposition pro UOW. Sie unterstützt den Zugriff auf eine einzige Datenbank innerhalb einer UOW. Ein Anwendungsprogramm kann zwar mehrere ferne Datenbanken aktualisieren, aber innerhalb einer UOW nur auf eine einzige Datenbank zugreifen.

Die *RUOW* verfügt über folgende Eigenschaften:

- Es werden Mehrfachanforderungen (SQL-Anweisungen) pro UOW unterstützt.
- Es werden mehrere Cursor pro UOW unterstützt.
- In jeder UOW kann nur eine Datenbank aktualisiert werden.
- Das Anwendungsprogramm schreibt die UOW entweder fest (COMMIT) oder setzt sie zurück (ROLLBACK). Bei bestimmten Fehlerbedingungen kann der Datenbankserver oder DB2 Connect einen Rollback der UOW durchführen.

Beispiel: Abb. 2 auf Seite 12 zeigt einen Datenbank-Client, auf dem eine Anwendung für Geldtransaktionen ausgeführt wird, die auf eine Datenbank zugreift, die Tabellen mit Giro- und Sparkonten sowie eine Aufstellung der Transaktionsgebühren enthält. Die Anwendung muss:

- den zu überweisenden Betrag von der Benutzerschnittstelle akzeptieren;
- den Betrag vom Sparkonto subtrahieren und den neuen Guthabenbetrag ermitteln;
- die Aufstellung der Bankgebühren lesen, um die Transaktionsgebühr für ein Sparkonto mit dem betreffenden Guthaben zu ermitteln;
- die Transaktionsgebühr vom Guthaben auf dem Sparkonto subtrahieren;
- den Überweisungsbetrag dem Girokonto gutschreiben (addieren);
- die Transaktion (UOW) festschreiben.

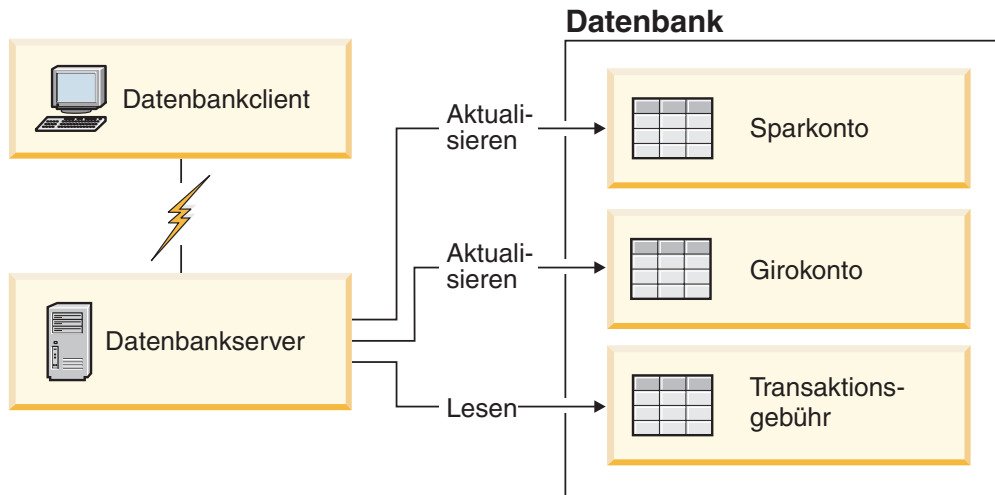


Abbildung 2. Verwendung einer einzelnen Datenbank in einer Transaktion

Um eine Anwendung einzurichten, müssen Sie folgende Tasks ausführen:

1. Erstellen der Tabellen für das Sparkonto, das Girokonto und die Aufstellung der Transaktionsgebühren in derselben Datenbank.
2. Falls physisch fern: Einstellen des Datenbankservers auf das entsprechende Kommunikationsprotokoll.
3. Falls physisch fern: Katalogisieren des Knotens und der Datenbank, um die Datenbank auf dem Datenbankserver zu identifizieren.
4. Vorkompilieren Ihres Anwendungsprogramms, um eine Typ-1-Verbindung anzugeben, d. h. Angeben von CONNECT(1) im Befehl PREP.

Verteilte Anforderungen

Eine *verteilte Anforderung* ist eine Funktion für verteilte Datenbanken, die es Anwendungen und Benutzern ermöglicht, SQL-Anweisungen zu übergeben, die sich auf zwei oder mehr Datenbankverwaltungssysteme oder Datenbanken in einer einzelnen Anweisung beziehen. Ein Beispiel ist ein Join zwischen Tabellen in zwei verschiedenen Subsystemen von DB2 für OS/390 und z/OS.

DB2 Connect bietet Unterstützung für verteilte Anforderungen zwischen Datenbanken und Datenbankverwaltungssystemen. Sie können beispielsweise die Operation UNION zwischen einer DB2-Tabelle und einer Oracle-Sicht ausführen. Zu den unterstützten Datenbankverwaltungssystemen gehören Produkte aus der DB2-Produktfamilie (wie beispielsweise DB2 Database für Linux, UNIX und Windows, DB2 für OS/390 und z/OS sowie DB2 für System i) und Oracle. Bei der Verwendung von DB2 Connect zusammen mit WebSphere Federation Server ist Unterstützung für mehrere Lieferanten verfügbar.

Die Funktion der verteilten Anforderung bietet *Transparenz bei der Speicherposition* von Datenbankobjekten. Wenn Informationen (in Tabellen und Sichten) verschoben werden, können Verweise auf diese Informationen (so genannte *Kurznamen*) aktualisiert werden, ohne Änderungen an den Anwendungen, die diese Informationen anfordern. Die Funktion der verteilten Anforderung bietet auch eine *Kompensation* für Datenbankverwaltungssysteme, die nicht alle DB2-SQL-Programmversionen oder nur bestimmte Optimierungsfunktionen unterstützen. Operationen, die unter einem solchen Datenbankverwaltungssystem nicht ausgeführt werden können (wie beispielsweise rekursives SQL) werden unter DB2 Connect ausgeführt.

Verteilte Anforderungen funktionieren auf *halbautonome* Weise. So können beispielsweise DB2-Abfragen, die Verweise auf Oracle-Objekte enthalten, übergeben werden, während Oracle-Anwendungen auf denselben Server zugreifen. Die verteilte Anforderung führt nicht dazu, dass der Zugriff (über die Integritäts- und Sperrbedingungen hinaus) auf Oracle oder andere Objekte von Datenbankverwaltungssystemen monopolisiert oder beschränkt wird.

Die Implementierung der Funktion der verteilten Anforderung besteht aus einer Instanz von DB2 Connect, einer Datenbank, die als föderierte Datenbank dient, und mindestens einer fernen Datenquelle. Die *föderierte Datenbank* umfasst Katalogeinträge zur Identifizierung von Datenquellen und deren Merkmalen. Eine *Datenquelle* besteht aus einem Datenbankverwaltungssystem und Daten. Anwendungen stellen Verbindungen zur föderierten Datenbank auf dieselbe Weise her, wie dies bei anderen DB2-Datenbanken der Fall ist. Die föderierte Datenbank von DB2 Connect verfügt über keine Lizenz für die Verwaltung von Benutzerdaten. Ihr einziger Zweck besteht darin, Informationen über Datenquellen zu speichern.

Nach der Einrichtung eines föderierten Systems kann auf die Informationen in den Datenquellen zugegriffen werden, als würden sie sich in einer großen Datenbank befinden. Benutzer und Anwendungen senden Abfragen an eine föderierte Datenbank, die daraufhin Daten nach Bedarf aus Systemen der DB2-Produktfamilie oder aus Oracle-Systemen abrufen. Benutzer und Anwendung geben in den Abfragen Kurznamen an. Diese Kurznamen stellen Verweise auf Tabellen und Sichten in den Datenquellen dar. Aus der Perspektive eines Endbenutzers entsprechen die Kurznamen in etwa den Aliasnamen.

Viele Faktoren können die Leistung verteilter Anforderungen beeinflussen. Der wesentlichste Faktor besteht darin sicherzustellen, dass genaue und aktuelle Informationen über die Datenquellen und ihre Objekte im Globalkatalog der föderierten Datenbank gespeichert sind. Diese Informationen werden vom DB2-Optimierungsprogramm verwendet und können einen Einfluss auf Entscheidungen haben, den Systembetrieb herunterzufahren, um eine Auswertung auf Ebene der Datenquellen durchzuführen.

Kapitel 3. DB2 Connect-Szenarios

DB2 Connect bietet eine Vielzahl von Lösungen für Ihre Anforderungen im Hinblick auf den Zugriff auf Host- oder System i-Datenbanken. Dieser Abschnitt beschreibt mehrere Szenarios, die unter Umständen auf Ihre spezifischen Anforderungen oder Ihre spezifische Umgebung zutreffen.

Direktzugriff auf Hostdatenbanken

Die Basisfunktion von DB2 Connect besteht darin, von Desktop-Anwendungen, die auf Windows-, Solaris- oder Linux-Workstations ausgeführt werden, eine Direktverbindung zu einer Hostdatenbank herzustellen. Am einfachsten lässt sich diese Lösung mit DB2 Connect Personal Edition implementieren.

Jede Workstation, auf der DB2 Connect Personal Edition installiert ist, kann eine direkte TCP/IP-Verbindung zu Servern mit DB2 Universal Database (UDB) für OS/390 und z/OS, DB2 für z/OS, DB2 UDB für iSeries, DB2 für i5/OS und DB2 Database für Linux, UNIX und Windows herstellen. Außerdem können Anwendungen innerhalb derselben Transaktion eine Verbindung zu mehreren Datenbanken aus der DB2-Produktfamilie herstellen und diese aktualisieren, wobei das Protokoll des zweiphasigen Commits die Datenintegrität vollständig gewährleistet.

In Abb. 3 auf Seite 16 wird eine Direktverbindung von einer Workstation, auf der DB2 Connect Personal Edition installiert ist, zu einem Host- oder System i-Datenbankserver hergestellt.

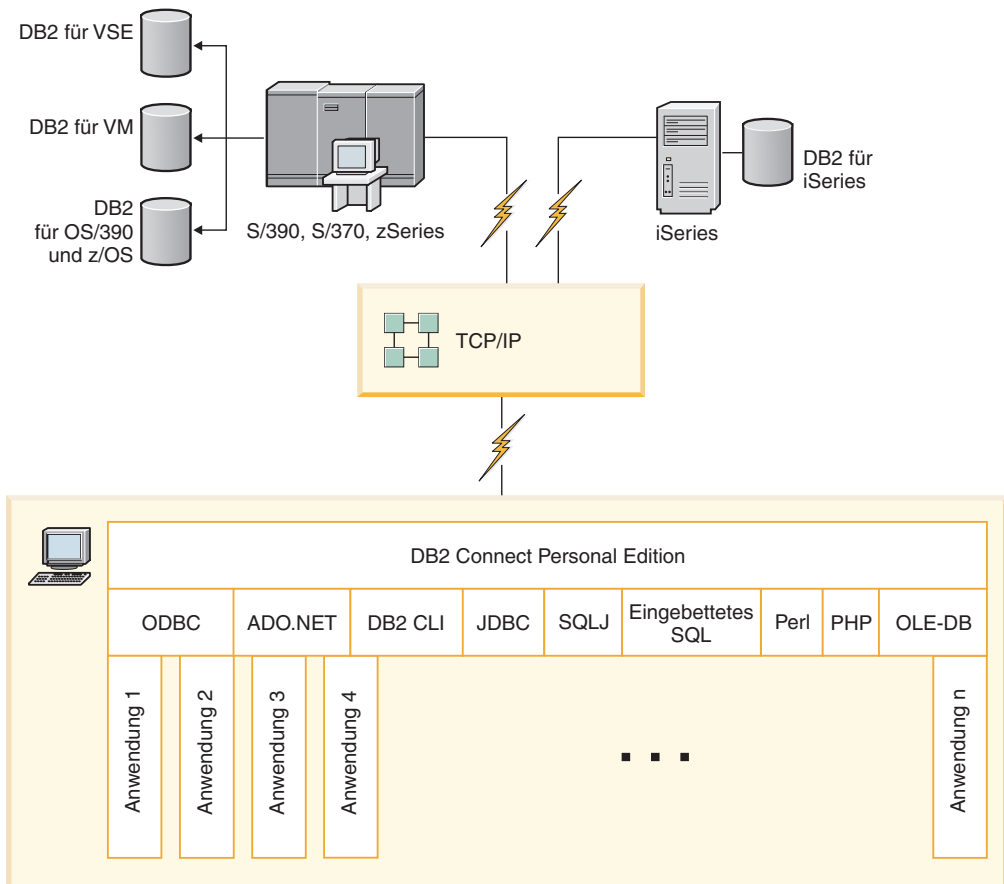


Abbildung 3. Direktverbindung zwischen DB2 Connect und einem Host- oder System i-Datenbankserver

Anmerkung:

1. DB2 muss nicht auf der DB2 Connect-Workstation installiert sein. Wenn Sie jedoch ein vollständiges Verwaltungssystem für relationale Datenbanken auf der DB2 Connect-Workstation benötigen, bestellen Sie DB2.
2. Der IBM Data Server Client ist jetzt Teil des DB2 Connect-Pakets und kann installiert werden, wenn ein Kunde das Produkt zur Anwendungsentwicklung verwenden möchte. Außerdem umfasst DB2 Connect jetzt Stored Procedure Builder, der verwendet werden kann, um gespeicherte Prozeduren für DB2 für OS/390 und z/OS zu erstellen, zu testen und zu implementieren.
3. C-Programmierer, die Windows-Anwendungen entwickeln, in denen Microsoft ODBC, OLE DB oder ActiveX Data Objects (ADO) verwendet wird, sollten *Microsoft Open Database Connectivity Software Development Kit* verwenden. Programmierer, die Anwendungen mithilfe der Programmiersprache Java entwickeln wollen, können eine beliebige Java-Entwicklungsumgebung verwenden.
4. Wenn eine Verbindung zu einem DB2 für z/OS-Datenbankserver, für den die Sysplex-Nutzung aktiviert ist, unterbrochen wird, versucht der Client automatisch, die Verbindung erneut herzustellen.

Zugreifen auf Host- oder System i-DB2-Daten mit DB2 Connect Personal Edition

Eine Direktverbindung ohne zwischengeschaltete Server ist eine sehr praktische und wünschenswerte Konfiguration. Das trifft vor allem dann zu, wenn der Host- oder der System i-Datenbankserver TCP/IP-Konnektivität unterstützt. In solchen Fällen stellt jede DB2 Connect-Workstation eine Direktverbindung zu dem Host- oder System i-Datenbankserver her.

Die TCP/IP-Konnektivität setzt voraus, dass die Host- oder System i-Datenbank TCP/IP unterstützt. Die folgenden Versionen unterstützen systemeigene TCP/IP-Verbindungen:

- DB2 Universal Database (UDB) für OS/390 und z/OS Version 7.1 oder höher
- DB2 UDB für iSeries Version 5 Release 1 oder höher
- DB2 Server für VSE & VM Version 7 oder höher

Für den Verbindungsaufbau zu einem IBM Host- oder einem System i-Datenbankserver benötigen Sie ein lizenziertes DB2 Connect-Produkt. Sie können keine direkte Verbindung zu einem IBM Host- oder System i-Datenbankserver mithilfe eines IBM Data Server-Clients herstellen.

In Abb. 4 auf Seite 18 wird eine Workstation mit installiertem DB2 Connect Personal Edition dargestellt, die direkt mit einem Host- oder System i-Datenbankserver verbunden ist.

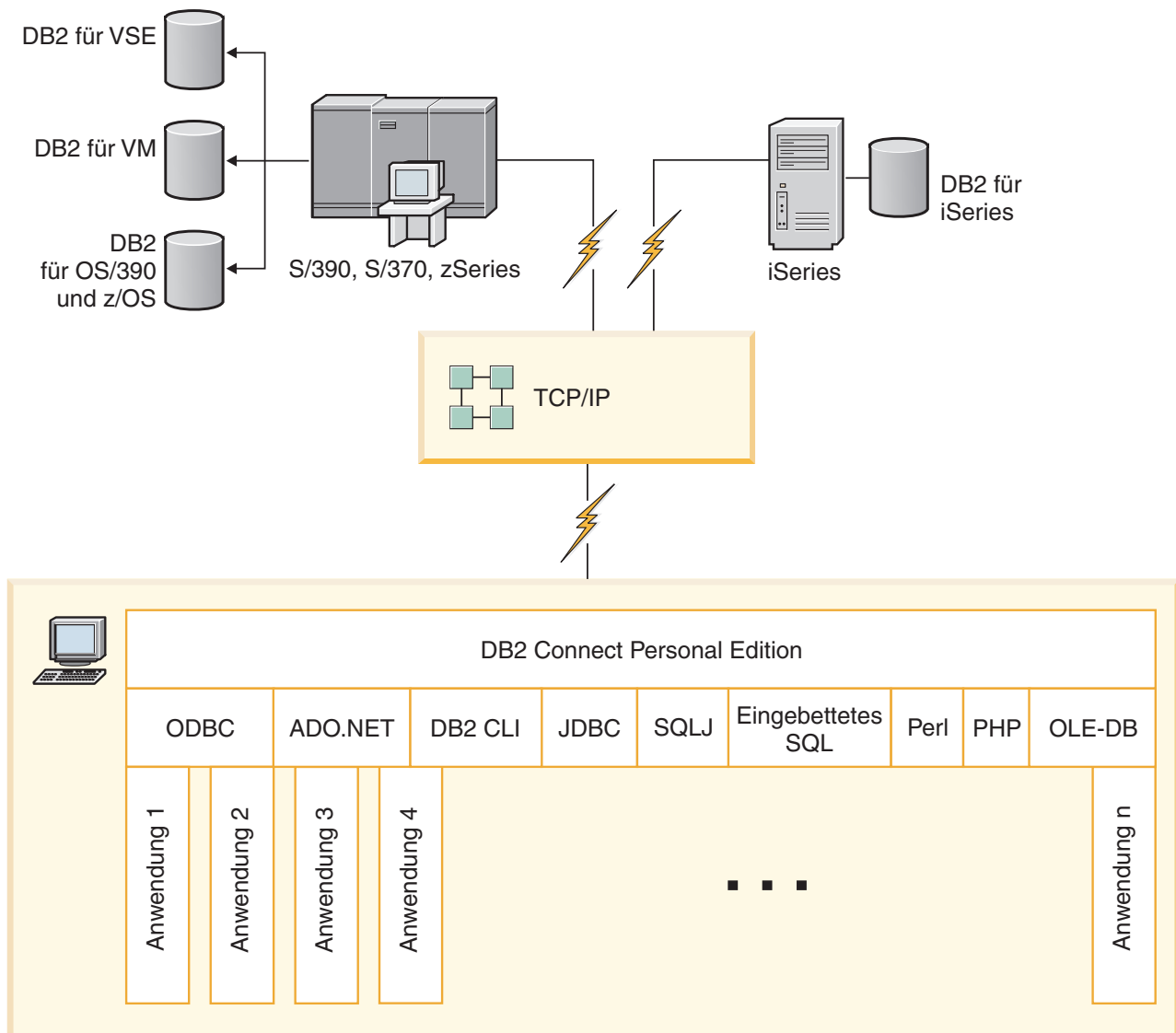


Abbildung 4. Direktverbindung zwischen DB2 Connect und einem Host- oder System i-Datenbankserver

DB2 Connect-Serverprodukte als Konnektivitätsserver

Ein DB2 Connect-Server ermöglicht es mehreren Clients, eine Verbindung zu Host- oder System i-Daten herzustellen, und kann den erforderlichen Aufwand zur Herstellung und Aufrechterhaltung des Zugriffs auf Unternehmensdaten beträchtlich senken. Abb. 5 auf Seite 19 veranschaulicht die IBM Lösung für Umgebungen, in denen ein DB2-Client eine indirekte Verbindung zu einem Host- oder System i-Datenbankserver über ein DB2 Connect-Serverprodukt, wie beispielsweise DB2 Connect Enterprise Server Edition, herstellen soll.

Anmerkung: Indirekte Verbindungen werden nur für DB2-Clients oder JCC-Clients unter Linux, UNIX oder Windows unterstützt. Wenn versucht wird, eine Verbindung zu einem Host- oder System i-Datenbankserver über ein DB2 Connect-Serverprodukt herzustellen und dabei einen anderen Client zu verwenden, tritt der Fehler SQL1334 auf.

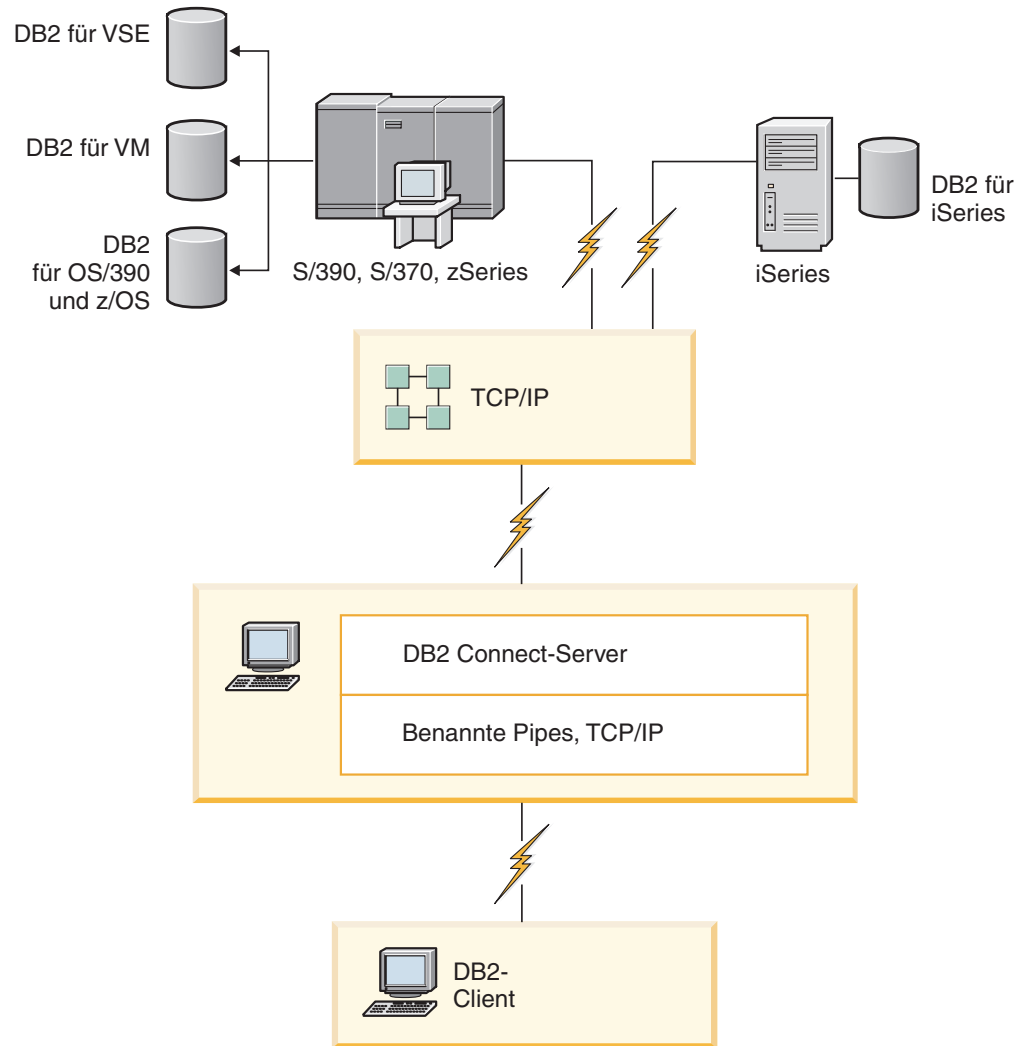


Abbildung 5. DB2 Connect Enterprise Server Edition

Wenn eine TCP/IP-Verbindung zum DB2 Connect-Server unterbrochen wird, versucht der Client automatisch, diese erneut herzustellen. Der Client versucht zunächst, die Verbindung zum ursprünglichen Server erneut herzustellen. Wenn die Verbindung nicht erneut hergestellt werden kann, weicht der Client auf einen alternativen DB2 Connect-Server aus. (Der Alternativserver wird in der Serverinstanz angegeben, und seine Adresse wird während der Verbindung an den Client zurückgegeben.) Kann die Verbindung zum Alternativserver nicht hergestellt werden, versucht der Client, die Verbindung zum ursprünglichen Server erneut herzustellen. Der Client versucht weiterhin, die Verbindung erneut herzustellen, und wechselt dabei zwischen dem ursprünglichen Server und dem Alternativserver ab, bis die Verbindung hergestellt ist oder der Grenzwert für die maximal mögliche Anzahl von Versuchen erreicht ist.

DB2 Connect und Webanwendungen

Der Web-Browser entwickelt sich schnell zu einer Standardschnittstelle für alles Mögliche, angefangen bei Onlinekatalogen bis hin zu Intranet-Anwendungen. Für einfache Web-Anwendungen reicht möglicherweise ein Web-Server aus.

Für Anwendungen mit hohem Volumen, die unter Umständen Zugriff auf Datenbanken und Transaktionsverarbeitung erfordern, bietet IBM Lösungen, die mithilfe von DB2 Connect eine sehr große Anzahl an gleichzeitig ablaufenden Transaktionen über das Web verwalten.

Vorteile und Einschränkungen der herkömmlichen CGI-Programmierung

e-business-Anwendungen im World Wide Web verwenden normalerweise Common Gateway Interface (CGI, allgemeine Gateway-Schnittstelle), um es Benutzern zu ermöglichen, Back-End-Datenbanken abzufragen. Viele Unternehmen verwenden Webanwendungen auch intern, und diese Anwendungen haben in der Regel ebenfalls eine Datenbank im Hintergrund.

Benutzer füllen Formulare auf einer Webseite aus, und diese Formulare werden über CGI an Anwendungen oder Scripts auf dem Web-Server übergeben. Das Script wiederum verwendet eine verfügbare Datenbank-API, um SQL-Abfragen an eine Host-Datenbank zu übergeben. Dasselbe Script kann dann eine Webseite (HTML) mit den Ergebnissen der Abfrage erstellen und an den Web-Browser des Benutzers zurücksenden, wo sie angezeigt wird. Ein Beispiel hierfür ist ein Online-katalog, in dem der Benutzer die Verfügbarkeit und den aktuellen Preis bestimmter Waren oder Dienstleistungen abfragen kann.

CGI-Anwendungen können leicht aufgebaut und verwaltet werden. Da der CGI-Standard sowohl von Betriebssystemen als auch Sprachen unabhängig ist, steht er auf fast allen Datenverarbeitungsplattformen zur Verfügung. CGI-Programme können in der Programmiersprache C++ oder in einer Scripting-Sprache wie Perl oder PHP geschrieben werden.

Doch obwohl CGI als eine ideale Lösung für webbasierte Anwendungen erscheinen mag, weist diese Schnittstelle doch deutliche Unzulänglichkeiten auf. Die Programmierungsumgebung für CGI ist nicht so leistungstark wie die anderer APIs. Darüber hinaus kann es zu Problemen mit der Skalierbarkeit bei großen E-Commerce-Operationen kommen. Jedes Mal, wenn eine CGI-Anwendung aufgerufen wird, wird auf dem Web-Server ein neuer Prozess erstellt. Jeder Prozess muss eine eigene Verbindung zur Datenbank herstellen und eine eigene Abfrage übergeben. In Transaktionsumgebungen mit hohem Volumen kann diese Einschränkung zu erheblichen Leistungsproblemen führen.

Sie können DB2 Connect mit einem Web-Server verwenden, um zuverlässige E-Commerce-Anwendungen mit hohem Volumen zu erstellen. DB2 Connect bietet mehrere Lösungen zur Verbesserung der Leistung von webbasierten Anwendungen. Gespeicherte Prozeduren ermöglichen es Benutzern von DB2 Connect, die Anzahl der an die Datenbank gesendeten Abfragen zu reduzieren.

Das Verbindungspooling reduziert die Häufigkeit, mit der Verbindungen zu einer Datenbank hergestellt oder unterbrochen werden.

Verwenden von PHP als Web-Server-Modul oder -Plug-in

Obwohl PHP für die CGI-Programmierung verwendet werden kann, wird es normalerweise als Web-Server-Modul oder -Plug-in verwendet. Bei einem Mehrprozess-Web-Server wie Apache kann der IBM DB2-Treiber für PHP verwendet werden, um die Skalierbarkeitsprobleme zu reduzieren.

Bei einem Mehrprozess-Web-Server werden Prozesse eines Prozesspools wiederverwendet, um Web-Server-Anforderungen zu verarbeiten. Damit nicht für jede Webanforderung eine Datenbankverbindung hergestellt werden muss, kann eine persistente Verbindung erstellt werden. In dieser Umgebung kann eine persistente Verbindung über ein einzelnes PHP-Script hinaus bestehen bleiben. Die Verbindung wird wiederverwendet, wenn eine nachfolgende Webanforderung eine identische Verbindung benötigt.

DB2 Connect und IBM WebSphere

IBM WebSphere stellt eine vollständigere e-business-Lösung zur Verfügung als es mit herkömmlichen Scripting-Tools wie PHP möglich ist. WebSphere-Anwendungsserver führen nicht nur die Scriptfunktionen von PHP aus, sondern ermöglichen Ihnen darüber hinaus, mithilfe von Servlets, Active Server Pages und Enterprise-JavaBeans komplexe und hochleistungsfähige Services über das Web zur Verfügung zu stellen und Unterstützung für Technologien auf Webbasis wie beispielsweise Java, TCP/IP, HTTP, HTTPS, HTML, DHTML, XML, MIME, SMTP, IIOP, X.509 und andere zu integrieren. Mit WebSphere haben Sie folgende Möglichkeiten:

- Nutzung von Industriestandards zur Beschleunigung der Entwicklung und Maximierung der Interoperabilität
- Anschließen von Tools-Technologien und Anwendungsgerüsten anderer Hersteller
- Analysieren der Leistung und Verwendung von Website-Inhalten
- Einfaches Skalieren Ihrer Site zur Integration von mehr Benutzern und zum Aufrechterhalten des Durchsatzes
- Implementierung auf vielen wichtigen Betriebssystemen (AIX, HP-UX, Linux, Novell NetWare, OS/390, z/OS, OS/400, Solaris, Microsoft Windows)
- Verwenden Ihres bestehenden Web-Servers, einschließlich derjenigen von Apache, IBM, Netscape und Microsoft.

WebSphere ist kein einzelnes Produkt, sondern eine Familie aus drei Produkten, die drei verschiedene Zielmärkte ansprechen. Das Kernstück der WebSphere-Lösung ist der WebSphere-Anwendungsserver.

WebSphere Application Server stellt eine Umgebung für drei Objekttypen bereit. Beim ersten Typ handelt es sich um die Java-Serverseiten, die Active Server Pages entsprechen. Die zweite Komponente besteht aus Java-Servlets, und bei der dritten handelt es sich um Enterprise JavaBeans. Enterprise JavaBeans sind ein Standard für die Implementierung sehr umfangreicher, zuverlässiger Unternehmensanwendungen, der zunehmend an Bedeutung gewinnt.

WebSphere-Anwendungen können auf derselben Plattform implementiert werden wie der Web-Server und DB2. Im Falle von DB2 Universal Database (UDB) für OS/390 und z/OS, DB2 für z/OS, DB2 für VM, DB2 für VSE, DB2 UDB für iSeries sowie DB2 für i5/OS wird WebSphere auf derselben Plattform implementiert wie das DB2 Connect-Serverprodukt.

Es stehen mehrere WebSphere-Lösungen sowie Rational Application Developer (RAD) zur Verfügung. Weitere Informationen finden Sie unter <http://www.ibm.com/software/webservers/appserv/was/>

DB2 Connect als Java-Anwendungsserver

Durch die Verwendung von Java lassen sich viele Nachteile scriptbasierter Sprachen umgehen. IBM stellt Applets und Anwendungen zur Verfügung, die es Ihnen ermöglichen, Java in allen Phasen einer Webtransaktion einzusetzen. Die Lösungen von IBM ermöglichen die Verwendung einer Mischung aus verschiedenen Technologien. Dadurch haben Sie die Möglichkeit, scriptbasierte Lösungen wie beispielsweise Perl DBI oder Microsoft Active Server Pages mit DB2 zu verwenden oder sich auf eine noch zuverlässigere Implementierung zu verlegen, wie sie durch einen Java-Anwendungsserver wie beispielsweise WebSphere zur Verfügung gestellt wird.

Für Java-Programmierer gibt es zwei Anwendungsprogrammierschnittstellen (APIs). Die erste, JDBC, unterstützt die Verwendung von Java zur Entwicklung von datensensitiven Java-Applets und Java-Anwendungen sowie Java-Servlets, Java-Serverseiten (JSP) und Enterprise JavaBeans (EJB). JDBC ist eine API auf Aufrufebene, d. h. zum Aufrufen von Methoden. Die andere Java-API heißt SQLJ. SQLJ ermöglicht die Angabe von SQL innerhalb eines Java-Programms. DB2 kann beide APIs verwenden, und zwar entweder auf der Client- oder der Serverseite einer Webtransaktion.

Auf der Clientseite werden Applets, datensensitive Applets und Anwendungen unterstützt. Auf der Datenbankseite bezieht sich die Aktivierung von Java auf Datenbankobjekte wie beispielsweise benutzerdefinierte Funktionen und gespeicherte Prozeduren.

Bei DB2 für OS/390 und z/OS, DB2 für VSE und VM, DB2 Universal Database (UDB) für iSeries und DB2 für i5/OS gibt es zwei unterschiedliche Methoden zur Implementierung einer Java-Anwendung. Sie können die direkte Konnektivität verwenden, die von DB2 Connect Personal Edition mit TCP/IP zur Verfügung gestellt wird, oder Sie verwenden ein DB2 Connect-Serverprodukt, das die Konnektivität zum Host- oder zum System i-Datenserver bereitstellt.

In beiden Fällen benötigt der Benutzer im Web keine besondere Software für den Zugriff auf die Datenbank, sondern lediglich einen Standard-Web-Browser. Außer einem DB2 Connect-Serverprodukt und einem Web-Server gemäß Industriestandard braucht nichts weiter installiert zu sein. Wenn der Web-Server und DB2 Connect sich nicht auf derselben physischen Maschine befinden, muss ein IBM Data Server-Client auf dem Web-Server installiert werden.

Die Schlüsselkomponente für DB2 für OS/390 und z/OS ist ein DB2 Connect-Serverprodukt auf einem Mittlerserver. Diese Komponente stellt die JDBC-Serveraktivierung bereit und stellt darüber hinaus die Verbindung zum Server unter DB2 für OS/390 und z/OS, DB2 für VSE und VM, DB2 Universal Database (UDB) für iSeries sowie DB2 für i5/OS her. Wiederum ist keine besondere Software für den Web-Browser des Clients erforderlich.

IBM stellt umfangreiche Unterstützung und Tools für die Entwicklung von Java-Anwendungen und -Applets zur Verfügung. Für die Entwicklung von Datenbank-anwendungen stellt DB2 Database Enterprise Developer Edition folgende Produkte zur Verfügung: Rational Web Developer, IBM Data Studio, DB2 Embedded Application Server, Cloudscape Version 10.2 sowie DB2 und DB2 Connect für Testzwecke. Tools anderer Hersteller wie beispielsweise NetBeans, Borland JBuilder oder Symantec Visual Cafe arbeiten ebenfalls mit den Datenbanklösungen von IBM zusammen.

DB2 Connect auf dem Web-Server

IBM stellt HTTP-Server (Web-Server) mit allen DB2 Connect-Produkten zur Verfügung. DB2 Connect-Serverprodukte wie DB2 Connect Enterprise Server Edition bieten sofort einsatzfähige Unterstützung für Apache- und Lotus Domino Go-Web-Server und können auch mit allen anderen Web-Servern wie beispielsweise Microsoft Internet Information Server oder Netscape Enterprise Server zusammenarbeiten.

Wenn Sie mit Datenbanken aus der DB2-Produktfamilie arbeiten, die auf zSeries-, System i-, VM- und VSE-Systemen ausgeführt werden, ist ein DB2 Connect-Serverprodukt auf dem Web-Server erforderlich. DB2 Connect-Serverprodukte stellen die erforderlichen Bibliotheken und Kommunikationsschnittstellen zur Verfügung, damit Web-Server auf diese Host- und System i-Plattformen zugreifen können. Für die Kommunikation zwischen dem Web-Server und einer Datenbank unter zSeries, System i, VM oder VSE kann TCP/IP verwendet werden.

Anmerkung: IBM Weblösungen bieten die Möglichkeit, innerhalb desselben CGI-Scripts (beispielsweise PHP) oder innerhalb derselben Transaktion in einem CGI-Script mit mehreren Datenbanken zu arbeiten.

Gespeicherte Prozeduren

Eine wichtige Überlegung im Hinblick auf Web-Anwendungen wie auch Client/Server-Umgebungen besteht darin, den Datenverkehr zwischen dem HTTP-Server und den Back-End-Datenbanken zu minimieren. Diese Überlegung gewinnt insbesondere bei der Transaktionsverarbeitung mit hohem Volumen an Bedeutung, die in den meisten e-business-Anwendungen eine zentrale Rolle spielt.

Der empfohlene Ansatz sieht eine Kombination aus der CGI-Anwendungsprogrammierung sowie der in gespeicherten Prozeduren integrierten Programmierung und Geschäftslogik vor. DB2 Database für Linux, UNIX und Windows und DB2 Universal Database (UDB) unter OS/390 und z/OS, DB2 für z/OS, DB2 UDB für iSeries, DB2 für i5/OS und DB2 für VSE verwenden alle dieselben Parameterkonventionen zum Aufrufen gespeicherter Prozeduren.

Wie auch die herkömmlichen Webschnittstellenscripts übergibt der Web-Browser das Formular an den Web-Server, auf dem das Webschnittstellenscript ausgeführt wird. Es wird jedoch nicht jede einzelne SQL-Anweisung an die DB2-Datenbank gesendet, sondern stattdessen eine Anforderung zur Ausführung einer gespeicherten Prozedur. Diese gespeicherte Prozedur umfasst eine Reihe von SQL-Anweisungen, die andernfalls einzeln hätten ausgeführt werden müssen. Gespeicherte Prozeduren reduzieren die Anzahl der Nachrichten, die zwischen dem Webschnittstellenscript und der Back-End-Datenbank hin- und hergesendet werden.

Der wichtigste Vorteil der gespeicherten Prozeduren liegt in der Reduzierung des Datenaustauschs auf dem Netz zwischen dem HTTP-Server und der DB2-Datenbank im Hintergrund (Back-End).

DB2 Connect und Anwendungsserver

Die Entstehung von Client/Server-Anwendungen ermöglichte es Anwendungsentwicklern, die Benutzerfreundlichkeit zu verbessern und die Schulungskosten zu reduzieren, indem Anwendungen mit grafischen Benutzerschnittstellen auf Plattformen wie beispielsweise Windows zur Verfügung gestellt wurden.

Gleichzeitig ermöglichten diese Anwendungen die Flexibilität, Datenbankverwaltungsfunktionen an zuverlässige Datenbankserver unter einer Vielzahl von Betriebssystemen und auf verschiedenen Hardwareplattformen zu delegieren.

Das Client/Server-Modell, bei dem die Anwendungslogik an Client-Workstations verteilt wird, wird im Allgemeinen als *zweischichtiges Client/Server-Modell* bezeichnet. In dem zweischichtigen Modell wird die Anwendung in der Clientschicht implementiert und der Datenbankserver implementiert den Server oder die Back-End-Schicht. DB2 Connect bietet vollständige Unterstützung für zweischichtige Client/Server-Anwendungen für folgende Datenbankserver: DB2 Universal Database (UDB) für OS/390 und z/OS, DB2 für z/OS, DB2 UDB für iSeries, DB2 für i5/OS und DB2 für VM und VSE.

Mit der zunehmenden Größe der Client/Server-Anwendungen wurde deutlich, dass das zweischichtige Client/Server-Modell beträchtliche Einschränkungen aufweist. Das Verteilen von umfangreicher Geschäftslogik an Hunderte oder sogar Tausende von Client-Workstations machte die Änderungsverwaltung zu einer komplexen und kostspieligen Aufgabe. Bei jeglichen Änderungen an den Geschäftsregeln wurde es erforderlich, den Clientteil der Anwendung zu ersetzen. Häufig mussten diese Verteilungen von Anwendungen auf allen Client-Workstations des Unternehmens gleichzeitig erfolgen, um zu gewährleisten, dass die Geschäftsregeln konsequent angewandt werden.

Eine weitere Unzulänglichkeit des zweischichtigen Client/Server-Modells, die mit zunehmender Größe deutlich wurde, besteht in der Menge der Ressourcen, die von solchen Anwendungen verbraucht werden. Das Implementieren Hunderter oder Tausender *Fat Clients*, wie zweischichtige Clients häufig genannt werden, erhöhte auf allen Client-Workstations den Bedarf an Verarbeitungsleistung und -kapazität. Des Weiteren werden auch die Anforderungen an den Datenbankserver stark erhöht, da jeder Client eine dedizierte Datenbankverbindung erfordert sowie die entsprechenden Ressourcen für die Aufrechterhaltung einer solchen Verbindung. Obwohl die Abhängigkeit des zweischichtigen Client/Server-Modells von der Verteilung der Geschäftslogik in gewissem Maße durch die weit reichende Verwendung gespeicherter Prozeduren reduziert werden kann, lassen sich die anderen Unzulänglichkeiten nicht so einfach beheben, ohne Änderungen an dem Modell vorzunehmen.

Eine Lösung mit Anwendungsservern

Im Zuge der ausufernden Kosten und Komplexität der zweischichtigen Client/Server-Anwendungen ging die Entwicklung der meisten der größten Anwendungen in Richtung mehrschichtiger Client/Server-Modelle. Im Rahmen des mehrschichtigen Modells bleibt die Rolle der Datenschicht unverändert. Die Clientschicht wird jedoch durch eine oder mehrere Mittelschicht(en) ergänzt. Meistens handelt es sich jedoch um eine Schicht; daher auch die Bezeichnung *dreischichtig*.

Im dreischichtigen Modell dient der Client lediglich noch zur Verarbeitung von Benutzerinteraktionen und enthält keine Geschäftslogik. Die Mittelschicht besteht aus einem oder mehreren Anwendungsserver(n). Die Aufgabe des Anwendungsservers besteht darin, eine zuverlässige, kostengünstige Implementierung der den Geschäftsprozessen und Geschäftsregeln zugrundeliegenden Logik zur Verfügung zu stellen. Wie bereits beim zweischichtigen Modell wird die Implementierung der Geschäftsregeln häufig durch die Verwendung von gespeicherten Prozeduren ergänzt, um die Leistung zu verbessern.

Da die Client-Workstations nicht mehr die umfangreiche Anwendungslogik implementieren, sondern lediglich noch die Benutzerinteraktionen verarbeiten, ist der Ressourcenbedarf der Clientschicht weitaus geringer. Daraus folgt, dass die Clientschicht im dreischichtigen Modell häufig als *Thin Client* bezeichnet wird. Da ein zentraler Anwendungsserver Anforderungen von allen Clients verarbeitet, kann er darüber hinaus Ressourcen wie beispielsweise Datenbankverbindungen mit allen Clients gemeinsam benutzen. Daher muss der Datenbankserver nicht mehr für jeden Anwendungsbenutzer eine dedizierte Verbindung verwalten.

In der Industrie gibt es heute viele Beispiele für dreischichtige Anwendungsserver. Fast alle ERP-Lieferanten (Enterprise Resource Planning; Unternehmensressourcenplanung) implementieren ihre Anwendungen mithilfe des dreischichtigen Modells wie beispielsweise SAP R/3- und PeopleSoft V7-Anwendungen. Weitere Beispiele umfassen ERM-Lieferanten (Enterprise Relationship Management, Unternehmensbeziehungsverwaltung) wie beispielsweise Siebel und Vantive.

Anwendungsserver und DB2 Connect

Die DB2 Connect-Serverprodukte bieten umfangreiche Unterstützung für die Implementierung mehrschichtiger Anwendungen. Die Unterstützung durch DB2 Connect umfasst eine Reihe unterschiedlicher APIs zur Entwicklung von Anwendungslogik (ODBC, ADO.NET, DB2 CLI, eingebettetes SQL, JDBC, SQLJ, Perl, PHP und OLE DB) sowie eine umfassende Kommunikationsinfrastruktur zur Interaktion mit Datenbankservern aus der DB2-Produktfamilie.

DB2 Connect unterstützt auch Implementierungen, in denen sich eine Datenbankschicht aus mehreren Datenbankservern der DB2-Produktfamilie zusammensetzt. Dadurch können Anwendungsserver Transaktionen implementieren, mit deren Hilfe sich die Daten auf mehreren Datenbankservern durch eine einzige Transaktion aktualisieren lassen.

Die Integrität solcher verteilter Transaktionen wird durch das von DB2 Connect unterstützte Protokoll des zweiphasigen Commits gewährleistet. Beispielsweise kann eine Anwendung innerhalb derselben Transaktion Daten in einer Datenbank unter DB2 für OS/390 und z/OS sowie unter DB2 Database für Linux, UNIX und Windows aktualisieren. Wenn Unterstützung für verteilte Anforderungen installiert und aktiviert wird, kann die Anwendung innerhalb derselben Transaktion eine Oracle-Datenbank lesen und eine Datenbank aus der DB2-Produktfamilie aktualisieren.

Im folgenden Diagramm werden sowohl die APIs als auch der Konnektivitätsmechanismus zwischen dem Anwendungsserver und den Back-End-Datenbankservern von einem DB2 Connect-Serverprodukt (z. B. DB2 Connect Enterprise Server Edition) zur Verfügung gestellt.

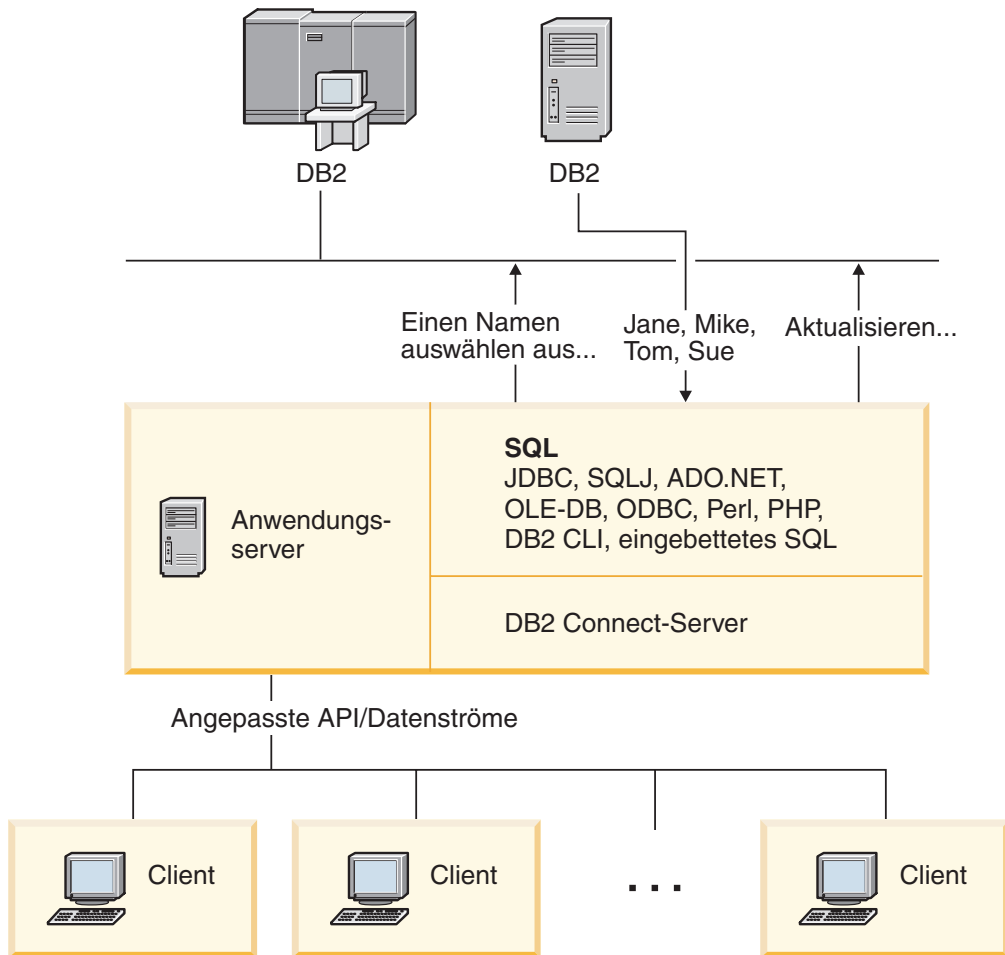


Abbildung 6. DB2 Connect-Unterstützung für Anwendungsserver

Die erweiterten Funktionen von DB2 Connect wie beispielsweise das Verbindungspooling sorgen für eine beträchtliche Reduzierung des Bedarfs an Anwendungsressourcen und eine vereinfachte Implementierung der Anwendungsserver.

DB2 Connect und Anwendungsserverkonfigurationen

Die Verwendung mit Anwendungsservern ist nur mit einem DB2 Connect-Serverprodukt möglich. DB2 Connect Personal Edition wird nicht unterstützt und ist für die Verwendung mit Anwendungsservern nicht lizenziert. Kunden, die Anwendungsserver implementieren, sollten darüber hinaus die mit ihrer DB2 Connect-Instanz gelieferten Vertragsbedingungen überprüfen, um zu ermitteln, wie viele Benutzerlizenzen erworben werden müssen.

Für die Implementierung von DB2 Connect in der Anwendungsserverumgebung stehen zwei Methoden zur Verfügung. Ein DB2 Connect-Serverprodukt kann auf einer der folgenden beiden Maschinen installiert werden:

- der Anwendungsservermaschine
- einer unabhängigen Kommunikationsservermaschine

In den meisten Fällen wird eine Kopie von DB2 Connect vorzugsweise auf demselben Server installiert wie der Anwendungsserver selbst. Die Installation von DB2 Connect auf dem Anwendungsserver ermöglicht die Teilnahme an Systemen für Funktionsübernahme und Lastausgleich, die der

Anwendungsserver möglicherweise implementiert. Diese Konfiguration bietet potenziell eine bessere Leistung, da ein zusätzlicher Zwischenschritt im Netz eliminiert wird, der erforderlich wird, wenn DB2 Connect auf einem anderen Server installiert ist. Darüber hinaus kann die Verwaltung vereinfacht werden, da kein zusätzlicher Server installiert und verwaltet werden muss.

Die Installation von DB2 Connect auf einem anderen Server ist dann eine gute Lösung, wenn für das Betriebssystem bzw. die Hardwareplattform, unter dem bzw. auf der der Anwendungsserver ausgeführt wird, kein DB2 Connect-Serverprodukt zur Verfügung steht.

DB2 Connect und Transaktionsverarbeitungsmonitore

Ein Anwendungsserver ermöglicht es einer großen Anzahl von Benutzern, Anwendungen mit einem Minimum an Systemressourcen auszuführen. Anwendungsserver lassen sich erweitern, damit koordinierte Transaktionen über Anwendungen aufgerufen werden können, die vom Anwendungsserver ausgeführt werden. Diese Transaktionskoordinierung wird normalerweise als TP-Monitor (Transaction Processing Monitor; Transaktionsverarbeitungsmonitor) bezeichnet. Ein TP-Monitor arbeitet mit einem Anwendungsserver zusammen.

Man kann sich eine *Transaktion* als Routineereignis (gewöhnlich eine Serviceanforderung) im alltäglichen Betrieb eines Unternehmens vorstellen. TP-Monitore sind für die ordnungsgemäße Verarbeitung von Transaktionen zuständig.

Transaktionsverarbeitung

Jedes Unternehmen hat Regeln und Prozeduren, die beschreiben, wie der Betrieb ablaufen soll. Die Benutzeranwendungen, die diese Regeln implementieren, können als *Geschäftslogik* bezeichnet werden. Die von diesen Geschäftsanwendungen ausgeführten Transaktionen werden häufig als Transaktionsverarbeitung (Transaction Processing) oder als Online-Transaktionsverarbeitung (Online Transaction Processing, OLTP) bezeichnet.

Die kommerzielle OLTP weist folgende Hauptmerkmale auf:

Viele Benutzer

Die Transaktionsverarbeitung wird im Allgemeinen von der Mehrzahl der Mitarbeiter eines Unternehmens verwendet, da der aktuelle Geschäftsstatus von so vielen Mitarbeitern beeinflusst wird.

Wiederholend

Bei den meisten Interaktionen mit dem Computer handelt es sich gewöhnlich um die gleichen, sich wiederholenden Prozesse, die immer wieder ausgeführt werden. Die Funktionen zur Eingabe von Bestellungen oder zur Verarbeitung von Zahlungen beispielsweise werden jeden Tag unzählige Male verwendet.

Kurze Interaktionen

Die meisten Interaktionen zwischen den Mitarbeitern des Unternehmens und dem Transaktionsverarbeitungssystem sind nur von kurzer Dauer.

Gemeinsame Daten

Da Daten den Status des Unternehmens wiedergeben, kann es nur ein einziges Exemplar der Daten geben.

Datenintegrität

Die Daten müssen den aktuellen Status des Unternehmens wiedergeben und in sich schlüssig sein. So muss beispielsweise jede Bestellung einem Kundeneintrag zugeordnet sein.

Geringe Kosten/Transaktionen

Da die Transaktionsverarbeitung einen direkten Kostenfaktor im Geschäftsbetrieb darstellt, müssen die Kosten des Systems minimiert werden. DB2 Connect ermöglicht es Anwendungen, die von einem Anwendungsserver unter Linux, UNIX oder Windows gesteuert werden, Transaktionen für ferne LAN-, Host- und System i-Datenbankserver auszuführen und diese Transaktionen von einem TP-Monitor koordinieren zu lassen.

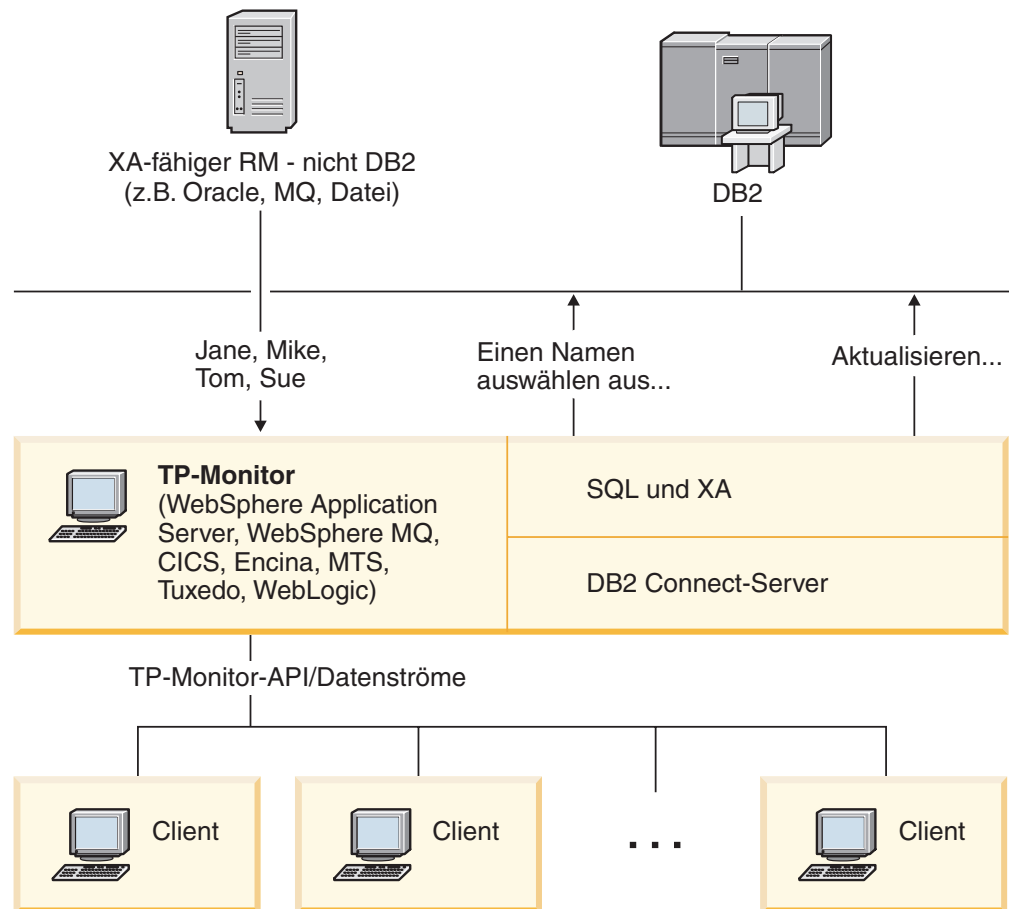


Abbildung 7. DB2 Connect-Unterstützung für TP-Monitore

In Abbildung Abb. 7 werden sowohl die APIs als auch der Konnektivitätsmechanismus zwischen dem Anwendungsserver und den Back-End-Datenbankservern von einem DB2 Connect-Serverprodukt (z. B. DB2 Connect Enterprise Server Edition) zur Verfügung gestellt.

Beispiele für Transaktionsverarbeitungsmonitore

Folgende TP-Monitore sind heute am häufigsten auf dem Markt erhältlich:

- IBM WebSphere Application Server
- IBM WebSphere MQ
- IBM TxSeries CICS
- IBM TxSeries Encina Monitor
- BEA Tuxedo
- BEA WebLogic
- Microsoft Transaction Server (MTS)

Ferne System i-, zSeries- und LAN-Datenbankserver können im Rahmen von Transaktionen verwendet werden, die von diesen TP-Monitoren koordiniert werden.

X/Open-Modell der verteilten Transaktionsverarbeitung (DTP)

Eine Anwendung, die Geschäftslogik ausführt, muss unter Umständen mehrere Ressourcen innerhalb einer einzigen Transaktion aktualisieren. Eine Bankanwendung beispielsweise, die die Überweisung eines Geldbetrages von einem Konto auf ein anderes durchführt, könnte die Belastung des Abbuchungskontos in einer Datenbank und die Gutschrift auf dem Empfängerkonto in einer anderen Datenbank erforderlich machen.

Es besteht auch die Möglichkeit, dass diese beiden Datenbanken von verschiedenen Lieferanten zur Verfügung gestellt werden. So kann es sich beispielsweise bei der einen Datenbank um eine Datenbank unter DB2 Universal Database für OS/390 und z/OS und bei der anderen um eine Oracle-Datenbank handeln. Anstatt nun für jeden TP-Monitor jeweils die eigene Transaktionsschnittstelle des entsprechenden Datenbanklieferanten zu implementieren, wurde eine gemeinsame Transaktionsschnittstelle definiert, und zwar zwischen einem TP-Monitor und allen Ressourcen, auf die eine Anwendung zugreift. Diese Schnittstelle wird als *XA-Schnittstelle* bezeichnet. Ein TP-Monitor, der die XA-Schnittstelle verwendet, wird als *XA-kompatibler Transaktionsmanager (TM)* bezeichnet. Eine aktualisierbare Ressource, die die XA-Schnittstelle implementiert, wird als *XA-kompatibler Ressourcenmanager (RM)* bezeichnet.

Bei den oben aufgeführten TP-Monitoren handelt es sich in allen Fällen um XA-kompatible Transaktionsmanager. Ferne Host-, System i- und LAN-gestützte DB2-Datenbanken stellen XA-kompatible Ressourcenmanager dar, wenn der Zugriff über DB2 Connect erfolgt. Daher können alle TP-Monitore, die über einen XA-kompatiblen Transaktionsmanager verfügen, Host-, System i- und LAN-gestützte DB2-Datenbanken innerhalb von Geschäftsanwendungen einsetzen, die Transaktionen ausführen.

Teil 2. DB2 Connect - Referenzinformationen

Kapitel 4. Aktualisieren von Datenbankverzeichnissen

DB2 Connect verwendet folgende Verzeichnisse zum Verwalten von Datenbankverbindungsinformationen:

- Das *Systemdatenbankverzeichnis*, das Informationen zu Namen, Knoten und Authentifizierungen für jede Datenbank enthält, auf die DB2 Connect zugreift.
- Das *Knotenverzeichnis*, das Informationen zu Netzwerkadressen und Kommunikationsprotokollen für jeden Host- oder System i-Datenbankserver enthält, auf den DB2 Connect zugreift.
- Das *DCS-Verzeichnis* (Database Connection Services Directory), das spezifische Informationen zu Datenbanken des Host- oder System i-Datenbankservers enthält.

Anmerkung:

1. Bevor Sie diese Verzeichnisse aktualisieren, sollten Sie die Kommunikation auf dem Host- oder System i-Datenbankserver und auf den Workstations konfigurieren.
2. Datenbankverzeichnisse können mithilfe des Konfigurationsassistenten aktualisiert werden.

Um Datenbankverzeichnisse zu aktualisieren, gehen Sie wie folgt vor:

1. Erfassen Sie die Datenbankverzeichnisinformationen mithilfe des Arbeitsblatts zur Verzeichnisanpassung.
2. Führen Sie die Schritte aus, die unter „Aktualisieren der Verzeichnisse mit Informationen zu fernen Datenbankservermaschinen“ in der Steuerzentrale beschrieben sind.

Werte des Systemdatenbankverzeichnisses

Im Systemdatenbankverzeichnis können folgende Informationen angegeben werden:

Datenbankname

Derselbe Wert, der in der Parametertabelle für das DCS-Verzeichnis angegeben wurde.

Aliasname der Datenbank

Ein Aliasname für den Host- oder System i-Datenbankserver. Dieser Name wird von jedem Anwendungsprogramm verwendet, das auf die Datenbank zugreift. Als Standardwert wird der für den Datenbanknamen angegebene Wert verwendet.

Format: 1 bis 8 alphanumerische Einzelbytezeichen, einschließlich des Nummernzeichens (#), des kommerziellen A (@), des Dollarzeichens (\$) und des Unterstreichungszeichens (_). Der Wert darf nicht mit einem Unterstreichungszeichen oder einer Zahl beginnen.

Knotenname

Derselbe Wert, der in der Parametertabelle für das Knotenverzeichnis angegeben wurde.

Authentifizierung (Authentication)

Gibt an, wo die Gültigkeitsprüfung für den Namen und das Kennwort des

Benutzers bei den vom DB2 Connect-Server ausgehenden Verbindungen erfolgen soll. Gültige Optionen sind: SERVER, SERVER_ENCRYPT, CLIENT, KERBEROS und DATA_ENCRYPT. Für den Authentifizierungstyp GSSPLUGIN ist im Systemdatenbankverzeichnis keine Unterstützung verfügbar.

Werte des Knotenverzeichnisses

Im Knotenverzeichnis können folgende Informationen angegeben werden:

Knotenname

Ein Kurzname für das System des Host- oder System i-Datenbankservers, auf dem sich die ferne Datenbank befindet. Dieser Name ist benutzerdefiniert. Derselbe Knotenname muss sowohl in der Parametertabelle für das Knotenverzeichnis als auch in der Parametertabelle für das Systemdatenbankverzeichnis angegeben werden.

Format: 1 bis 8 alphanumerische Einzelbytezeichen, einschließlich des Nummernzeichens (#), des kommerziellen A (@), des Dollarzeichens (\$) und des Unterstreichungszeichens (_). Der Wert darf nicht mit einem Unterstreichungszeichen oder einer Zahl beginnen.

Protokoll (Protocol)

Muss TCP/IP sein.

Sicherheitseinstufung (Security Type)

Die vorzunehmende Art von Sicherheitsprüfung. Für TCP/IP-Knoten ist SECURITY SOCKS eine Option, die angibt, dass der Knoten für SOCKS aktiviert wird. In diesem Fall sind die Umgebungsvariablen SOCKS_NS und SOCKS_SERVER verbindlich und müssen zur Aktivierung von SOCKS festgelegt sein.

Ferner TCP/IP-Hostname oder ferne IP-Adresse

Beim Definieren eines TCP/IP-Knotens ist dies entweder der ferne TCP/IP-Hostname oder die ferne TCP/IP-Adresse. Wenn ein Hostname angegeben ist, muss er auf der DB2 Connect-Workstation aufgelöst werden. Dies erfolgt entweder durch die DNS-Suchfunktion (Domain Name System Lookup Function) oder durch einen Eintrag in der lokalen TCP/IP-Datei 'hosts'.

Bei fernen Hosts unter DB2 für OS/390 und z/OS wird der Hostname in der Nachricht DSNL004I (DOMAIN=hostname) angezeigt, wenn DDF (Distributed Data Facility) gestartet wird. Der Befehl -DISplay DDF kann ebenfalls verwendet werden.

Beim Zugriff auf eine z/OS-Gruppe mit gemeinsamer Datenbenutzung muss der Domänenname der dynamischen VIPA-Adresse der DB2-Gruppe entsprechen. Diese Adresse bewirkt eine Weiterleitung an das DB2-Member mit der geringsten Auslastung. Um auf ein bestimmtes Member zuzugreifen, verwenden Sie die dynamische VIPA-Adresse des entsprechenden DB2-Members und inaktivieren Sie die Sysplex-Routing-Funktion. Für jedes Member zeigt die Nachricht DSNL004I den memberspezifischen Domänennamen an.

TCP/IP-Servicename oder -Portnummer (TCP/IP Service Name or Port number)

Beim Definieren eines TCP/IP-Knotens ist dies entweder der ferne TCP/IP-Servicename oder die ferne TCP/IP-Portnummer. Dieser Wert muss für TCP/IP auf dem fernen Host definiert werden. Portnummer 446 wurde als Standardportnummer für DRDA eingetragen.

Bei fernen Hosts unter DB2 für OS/390 und z/OS ist die Portnummer in der BSDS (Boot Strap Data Set) als PORT definiert und wird auch in der Nachricht DSNL004I (TCPPORT=portnummer) angegeben, wenn DDF (Distributed Data Facility) gestartet wird. Der Befehl -DISplay DDF kann ebenfalls verwendet werden.

Beim Zugriff auf eine z/OS-Gruppe mit gemeinsamer Datenbenutzung muss der Domänenname der dynamischen VIPA-Adresse der DB2-Gruppe entsprechen. Diese Adresse bewirkt eine Weiterleitung an das DB2-Member mit der geringsten Auslastung. Um auf ein bestimmtes Member zuzugreifen, verwenden Sie die dynamische VIPA-Adresse des entsprechenden DB2-Members und inaktivieren Sie die Sysplex-Routing-Funktion. Für jedes Member zeigt die Nachricht DSNL004I den memberspezifischen Domänennamen an.

Anmerkung: Ein zweiter Port für Resynchronisationsoperationen mit zweiphasigem Commit über TCP/IP-Verbindungen kann vom Server zugeordnet werden. Zum Beispiel ordnet das BSDS von DB2 Universal Database für z/OS und OS/390 eine Portnummer (RESPORT) für die Resynchronisation von eingehenden Verbindungen nur DB2 Universal Database für z/OS und OS/390 zu. Dafür muss kein Servicename definiert werden.

Werte des DCS-Verzeichnisses

Im DCS-Verzeichnis können folgende Informationen angegeben werden:

Datenbankname

Ein benutzerdefinierter Kurzname für den Host- oder System i-Datenbankserver. Derselbe Datenbankname muss sowohl in der Parametertabelle für das DCS-Verzeichnis als auch in der Parametertabelle für das Systemdatenbankverzeichnis angegeben werden.

Format: 1 bis 8 alphanumerische Einzelbytezeichen, einschließlich des Nummernzeichens (#), des kommerziellen A (@), des Dollarzeichens (\$) und des Unterstreichungszeichens (_). Der Wert darf nicht mit einem Unterstreichungszeichen oder einer Zahl beginnen.

Zioldatenbankname (Target Database Name)

Die Datenbank auf dem Host- oder System i-Datenbankserver; folgende Angaben sind möglich:

OS/390 und z/OS

Ein DB2 Universal Database für z/OS und OS/390-Subsystem, das durch seinen Standortnamen (LOCATION NAME) oder durch einen der auf dem z/OS-Server definierten Standortaliasnamen identifiziert wird.

Der LOCATION NAME kann durch Anmeldung bei TSO und Ausgabe der folgenden SQL-Abfrage über eines der verfügbaren Abfrage-Tools ermittelt werden:

```
select current server from sysibm.sysdummy1
```

Mehrere Standortnamen (LOCATION NAME) sind auch in der BSDS (Boot Strap Data Set) definiert und werden in der Nachricht DSNL004I (LOCATION=standort) angegeben, die beim Starten von DDF (Distributed Data Facility) erstellt wird. Der Befehl -DISplay DDF kann ebenfalls verwendet werden.

Beim Zugriff auf eine z/OS-Gruppe mit gemeinsamer Datenbenutzung muss der Domänenname der dynamischen VIPA-Adresse der DB2-Gruppe entsprechen. Diese Adresse bewirkt eine Weiterleitung an das DB2-Member mit der geringsten Auslastung. Um auf ein bestimmtes Member zuzugreifen, verwenden Sie die dynamische VIPA-Adresse des entsprechenden DB2-Members und inaktivieren Sie die Sysplex-Routing-Funktion. Für jedes Member zeigt die Nachricht DSNL004I den memberspezifischen Domänennamen an.

VSE oder VM

Der Datenbankname (DBNAME)

OS/400 und z/OS

Der Name der relationalen Datenbank (RDBNAME)

Andere Systeme

Bei den Betriebssystemen Windows, Linux und UNIX ist dies der Aliasname der Datenbank aus dem Datenbankverzeichnis.

Parameterzeichenfolge (Parameter String)

Wenn die Standardwerte geändert werden sollen, müssen beliebige oder alle der folgenden Parameter in der nachstehenden Reihenfolge angegeben werden.

zuordnungsdatei

Der Name einer SQLCODE-Zuordnungsdatei, die die Standard-SQLCODE-Zuordnung überschreibt. Wenn die SQLCODE-Zuordnung ausgeschaltet werden soll, geben Sie **NOMAP** an.

Anmerkung: Bei der Verarbeitung einer Abfrageanforderung gibt der DRDA-Server Daten als eine Gruppe von Zeilen zurück, die die Ergebnismenge darstellen. Mit jeder Zeile wird auch ein SQL-Kommunikationsbereich zurückgegeben, der normalerweise einen SQLCODE-Wert von null oder einen positiven SQLCODE-Wert (wie beispielsweise +12 oder +802) enthält. Wenn Sie auf einem DB2 Connect-Server eine angepasste Zuordnungsdatei verwenden, werden solche positiven SQLCODE-Werte nicht zugeordnet, wenn sie in der angepassten Zuordnungsdatei enthalten sind und über angepasste Zuordnungen verfügen (also beispielsweise einem anderen SQLCODE-Wert zugeordnet werden oder angepasste Token-Zuordnungen haben).

Folgendes ist unbedingt zu beachten:

1. Positive SQLCODE-Werte stellen Warnungen dar, im Gegensatz zu negativen SQLCODE-Werten, die auf Fehlerbedingungen hinweisen. Alle negativen SQLCODE-Werte werden in jedem Fall zugeordnet, unabhängig von der verwendeten Zuordnungsdatei. Alle positiven SQLCODE-Werte, die in der angepassten Zuordnungsdatei enthalten sind und untereinander ohne Änderungen zugeordnet werden, werden ebenfalls in jedem Fall zugeordnet. Auch diejenigen positiven SQLCODE-Werte, die nicht in der angepassten

Zuordnungsdatei auf dem DB2 Connect-Server enthalten sind, werden in jedem Fall zugeordnet.

2. Wenn Sie eine Standardzuordnungsdatei verwenden oder eine direkte Verbindung zur Hostdatenbank herstellen, erfolgt die Zuordnung der SQLCODE-Werte stets für alle SQLCODE-Werte.

,D Dies ist der zweite positionsgebundene Parameter. Wird er angegeben, trennt die Anwendung die Verbindung zur Datenbank des Host- oder System i-Datenbankservers, wenn einer der folgenden SQLCODE-Werte zurückgegeben wird:

SQL3000N
SQL30040N
SQL30050N
SQL30051N
SQL30053N
SQL30060N
SQL30070N
SQL30071N
SQL30072N
SQL30073N
SQL30074N
SQL30090N

Wenn der Trennungsparameter **,D** nicht angegeben ist, wird die Verbindung nur getrennt, wenn die folgenden SQLCODE-Werte zurückgegeben werden:

SQL30020N
SQL30021N
SQL30041N
SQL30061N
SQL30081N

Das Handbuch *Fehlernachrichten* enthält Erläuterungen dieser Codes.

Anmerkung: Wenn DB2 Connect aufgrund eines Fehlers getrennt wird, wird automatisch ein Rollback durchgeführt.

„INTERRUPT_ENABLED

Dies ist der dritte positionsgebundene Parameter. INTERRUPT_ENABLED gilt nur dann, wenn der Endserver keine Unterbrechungen unterstützt. Wenn ein Server den DRDA-Unterbrechungsfluss unterstützt, übergibt DB2 Connect die Unterbrechungsanforderung einfach an den Server. Wenn INTERRUPT_ENABLED im DCS-Verzeichnis auf der DB2 Connect-Workstation konfiguriert ist und eine Clientanwendung eine Unterbrechung absetzt, während eine Verbindung zum Host- oder System i-Datenbankserver besteht, führt DB2 Connect die Unterbrechung aus. Dabei wird die Verbindung freigegeben und die UOW mit einem Rollback rückgängig gemacht. Dieses Unterbrechungsverhalten wird von AIX und Windows unterstützt.

Die Anwendung empfängt den SQLCODE -30081, der darauf hinweist, dass die Verbindung zum Server getrennt wurde. Die Anwendung muss anschließend eine neue Verbindung zum Host- oder System i-Datenbankserver herstellen, damit weitere Datenbankanforderungen verarbeitet

werden können. Auf anderen Plattformen als AIX ab Version 5.2 und Windows unterstützt DB2 Connect die Option der automatischen Verbindungsunterbrechung nicht, wenn eine Anwendung, die DB2 Connect verwendet, eine Unterbrechungsanforderung empfängt.

Anmerkung: Diese Unterstützung funktioniert für TCP/IP-Verbindungen auf allen Plattformen. Der Client kann u. U. den Socket mit KILL abbrechen, aber je nach Serverimplementierung können noch Daten zum Empfang bereitstehen oder nicht. DB2 Universal Database für z/OS und OS/390 verwendet asynchrone Socket-Aufrufe und kann daher den Verlust der Verbindung erkennen und für alle lang andauernden SQL-Anweisungen, die gerade ablaufen, einen Rollback durchführen.

,,,,,SYSPLEX

Mit diesem Parameter, dem sechsten positionsgebundenen Parameter, können Sie explizit SYSPLEX-Unterstützung unter DB2 Connect für eine bestimmte Datenbank aktivieren.

,,,,,LOCALDATE=<wert>

Mit diesem Parameter, dem siebten positionsgebundenen Parameter, können Sie Unterstützung für Datumsformatierung unter DB2 Connect aktivieren. Sie wird wie folgt mithilfe einer Datumsmaske für <wert> implementiert:

Angenommen, Sie setzen die folgenden Anweisungen vom Befehlszeilenprozessor (CLP) ab:

```
catalog TCP/IP node nynode remote myhost server myport
catalog dcs database nydb1 as new_york
catalog database nydb1 as newyork1 at node nynode
authentication server
```

Mit dem Aliasnamen für die Datenbank *newyork1* soll ohne Datumsumsetzung auf eine Host-Datenbank zugegriffen werden, weil keine Datumsmaske angegeben wurde.

Mit der neuen Unterstützung für Datumsformatierung können Sie jedoch die folgenden Befehle des Befehlszeilenprozessors verwenden. Da in diesem Fall der Befehlszeilenprozessor verwendet und die Parameterzeichenfolge in doppelten Anführungszeichen angegeben ist, muss der Wert für LOCALDATE innerhalb von zwei Paaren doppelter Anführungszeichen angegeben werden. Durch die Verwendung des Escape-Zeichens für das Betriebssystem "\" (umgekehrter Schrägstrich) können Sie sicherstellen, dass die doppelten Anführungszeichen nicht aus der Angabe für LOCALDATE entfernt werden.

```
catalog dcs database nydb2 as new_york
parms "\",,,,,,LOCALDATE=\"\"JJJJMMTT\"\"\\"
catalog database nydb2 as newyork2 at node nynode
authentication server
```

Mit dem Aliasnamen *newyork2* für die Datenbank können Sie auf dieselbe Hostdatenbank zugreifen. Für diesen Aliasnamen wurde jedoch zusätzlich eine Datumsformatmaske

angegeben. Dieses Beispiel verdeutlicht, dass die Datumsformatmaske mit dem Schlüsselwort LOCALDATE angegeben wird und dass es sich hierbei um den siebten positionsgebundenen Parameter im Feld PARMS eines DCS-Verzeichniseintrags handelt.

Die Datumsmaske ist nur gültig, wenn ALLE folgenden Aussagen zutreffen (d. h. wahr sind):

1. Es kann nur jeweils eine Folge von Angaben aus J, M und T geben, wobei J eine Jahresziffer, M eine Monatsziffer und T eine Tagesziffer ist.
2. Die maximale Anzahl für J in einer Folge ist 4.
3. Die maximale Anzahl für M in einer Folge ist 2.
4. Die maximale Anzahl für T in einer Folge ist 2.

Die folgenden Angaben sind alle gültige Datumsmasken:

"JJjMmTt" - Die Groß-/Kleinschreibung wird bei J, M und T nicht berücksichtigt
"MM+TT+JJJJ" - Maske mit mehr als 10 Byte.
Ferner sind andere Zeichen als J, M und T in der Maske zulässig
"abcJJ+MM" - Folge ohne T zulässig

Die folgenden Angaben sind alle ungültige Datumsmasken:

"JJJJJMMTT" - ungültig, weil 5 J in einer Folge
"JJJJMTT" - ungültig, weil 2 Folgen von M

Wenn eine Datumsformatmaske ungültig ist, wird kein Fehler abgesetzt. Sie wird lediglich ignoriert. Selbst wenn eine Datumsmaske gültig ist, bedeutet dies nicht automatisch, dass sie verwendet wird. Datumsformatumsetzung basierend auf einer gültigen Datumsmaske wird nur ausgeführt, wenn ALLE folgenden Aussagen wahr sind:

1. Es gibt keine SQL-Fehler.
2. Die Ausgabe ist ein Datumswert in ISO-ähnlichem (ISO und JIS) Format.
3. Der Ausgabedatenbereich ist mindestens 10 Byte lang. Dies ist die minimale Größe eines Ausgabedatenbereichs für die Speicherung eines Datenwerts, selbst wenn KEINE Datumsformatumsetzung ausgeführt wird. Diese Anforderung gilt selbst dann, wenn die Datumsformatmaske kürzer als 10 Byte ist.
4. Im DCS-Verzeichniseintrag ist eine gültige Datumsformatmaske angegeben, und diese Maske passt in den Ausgabedatenbereich.

,,,,,,BIDI=<ccsid>

Mit diesem Parameter, dem neunten positionsgebundenen Parameter, können Sie die bidirektionale ID für codierten Zeichensatz angeben, durch die der Standardwert für die bidirektionale (BIDI) ID für codierten Zeichensatz (CCSID) der Serverdatenbank überschrieben werden soll. Beispiel:

" , , , , , , , BIDI=xyz"

Hierbei gilt: xyz ist die Überschreibung für die ID für codierten Zeichensatz.

Arbeitsblatt zur Verzeichnisanpassung

Das Arbeitsblatt zur Verzeichnisanpassung zeigt die Informationen, die erfasst werden müssen. Es ist sinnvoll, das Arbeitsblatt zu kopieren und die eigenen Systemwerte einzugeben.

Parameter für das Knotenverzeichnis

Table 1. Parameter für das Knotenverzeichnis

Parameter	Beispiel	Ihr Wert
Knotenname (Node Name)	DB2NODE	
Ferner Hostname (TCP/IP-Knoten) (Remote Hostname (TCP/IP Node))	ZOSHOST	
Server (TCP/IP-Servicename oder -Portnummer) (Server (TCP/IP service name or port number))	db2inst1c (oder 446)	

Anmerkung:

1. Die TCP/IP-Portnummer für DRDA ist standardmäßig 446.
2. Geben Sie für TCP/IP-Knoten nur dann SECURITY an, wenn Sie sicher sind, dass der Host- oder System i-Datenbankserver SECURITY SOCKS unterstützt.

DCS-Verzeichnisparameter

Table 2. DCS-Verzeichnisparameter

Parameter	Beispiel	Ihr Wert
Datenbankname (Database Name)	DB2DB	
Zieldatenbankname (Target Database Name)	NEW_YORK3	
Anwendungsrequester (Application Requester)		
Parameterzeichenfolge (Parameter String)	" ,,,,,LOCALDATE= \" \"JJMMTT \" \" \" \"	

Parameter für das Systemdatenbankverzeichnis

Table 3. Parameter für das Systemdatenbankverzeichnis

Parameter	Beispiel	Ihr Wert
Datenbankname (Database Name)	DB2DB	
Aliasname für die Datenbank (Database Alias)	NYC3	
Knotenname (Node Name)	DB2NODE	
Authentifizierung (Authentication)	SERVER	

Definieren von mehreren Einträgen für dieselbe Datenbank

Für jede Datenbank muss mindestens ein Eintrag in jedem der drei Verzeichnisse (Knotenverzeichnis, DCS-Verzeichnis und Systemdatenbankverzeichnis) definiert werden. In einigen Fällen ist es möglicherweise sinnvoll, mehr als einen Eintrag für die Datenbank zu definieren.

Zum Beispiel soll möglicherweise die SQLCODE-Zuordnung für Anwendungen, die vom Host- oder System i-Datenbankserver übertragen wurden, ausgeschaltet, aber die Standardzuordnung für Anwendungen, die für die Client-/Server-Umgebung entwickelt wurden, akzeptiert werden. Hierfür müssten Sie wie folgt vorgehen:

- Einen Eintrag im Knotenverzeichnis definieren
- Zwei Einträge im DCS-Verzeichnis mit verschiedenen Datenbanknamen definieren. Für einen Eintrag den Wert NOMAP in der Parameterzeichenfolge angeben.
- Zwei Einträge im Systemdatenbankverzeichnis definieren, mit verschiedenen Aliasnamen für die Datenbank und den zwei Datenbanknamen, die im DCS-Verzeichnis angegeben wurden.

Beide Aliasnamen greifen auf dieselbe Datenbank zu, einer mit und der andere ohne SQLCODE-Zuordnung.

Handhabung von BIDI-Daten

Der folgende Abschnitt bezieht sich nur auf OS/390- und z/OS-Server. Diese Funktion darf für einen DB2 für i5/OS-Server nicht implementiert werden, da die BIDI-Unterstützung hierfür bereits in vollem Umfang zur Verfügung steht.

Die folgenden BIDI-Attribute sind für die ordnungsgemäße Handhabung von BIDI-Daten auf verschiedenen Plattformen erforderlich:

- Zeichengestaltung (ARABIC bzw. HINDI)
- Ausrichtung (RIGHT-TO-LEFT bzw. LEFT-TO-RIGHT)
- Gestaltung (SHAPED bzw. UNSHAPED)
- Symmetrische Auslagerungsfunktion (YES bzw. NO)
- Textart (LOGICAL bzw. VISUAL)

Da die Standardeinstellungen auf verschiedenen Plattformen unterschiedlich sind, kommt es zu Problemen, wenn DB2-Daten von einer Plattform an eine andere gesendet werden. Zum Beispiel verwenden Windows-Plattformen Daten des Typs LOGICAL UNSHAPED, während Daten unter OS/390 oder z/OS in der Regel das Format SHAPED VISUAL aufweisen. Daher werden Daten, die ohne Unterstützung für BIDI-Attribute von DB2 für OS/390 und z/OS an DB2 Connect unter Windows gesendet werden, nicht ordnungsgemäß angezeigt.

Wenn Daten zwischen DB2 Connect und einer Datenbank auf einem Server ausgetauscht werden, führt in der Regel der Empfänger die Umsetzung der eingehenden Daten aus. Diese Vereinbarung gilt normalerweise auch für die BIDI-Layoutumsetzung, die zusätzlich zur gewöhnlichen Codepage-Umsetzung stattfindet. Gegenwärtig unterstützt jedoch kein DB2-Produkt auf einem Host BIDI-spezifische IDs für codierten Zeichensatz (CCSID) oder die BIDI-Layoutumsetzung. Daher wurde DB2 Connect durch eine wahlfreie Funktion erweitert, die eine BIDI-Layoutumsetzung für Daten ausführt, die an die Serverdatenbank gesendet werden sollen und die von der Serverdatenbank empfangen werden.

Damit DB2 Connect die BIDI-Layoutumsetzung für Daten ausführen kann, die an eine Serverdatenbank abgehen, muss die bidirektionale ID für codierten Zeichensatz der Serverdatenbank überschrieben werden. Dies wird durch die Verwendung des Parameters BIDI im Feld PARMS des DCS-Datenbankverzeichniseintrags für die Serverdatenbank erzielt.

Die Verwendung dieser Funktion lässt sich am besten anhand eines Beispiels verdeutlichen.

Angenommen, Sie arbeiten mit einem hebräischen IBM Data Server-Client mit der CCSID 62213 (BIDI-Zeichenfolgetyp 5), und Sie möchten auf eine DB2-Hostdatenbank mit der CCSID 424 (BIDI-Zeichenfolgetyp 4) zugreifen. Ihnen ist jedoch bekannt, dass die in der DB2-Hostdatenbank gespeicherten Daten stattdessen auf CCSID 62245 (BIDI-Zeichenfolgetyp 10) basieren.

In dieser Situation gibt es zwei Probleme. Zum einen kennt die DB2-Hostdatenbank nicht den Unterschied zwischen den BIDI-Zeichenfolgetypen mit den CCSIDs 424 und 62245. Zum anderen erkennt die DB2-Hostdatenbank die IBM Data Server-Client-CCSID 62213 nicht. Sie unterstützt nur CCSID 62209 (BIDI-Zeichenfolgetyp 10), die auf derselben Codepage basiert wie CCSID 62213.

Sie müssen sicherstellen, dass die an die DB2-Hostdatenbank gesendeten Daten von Anfang an im BIDI-Zeichenfolgetyp 6 vorliegen, und Sie müssen DB2 Connect mitteilen, dass eine BIDI-Layoutumsetzung für die von der DB2-Hostdatenbank empfangenen Daten ausgeführt werden soll. Verwenden Sie die folgende Katalogisierung für die DB2-Hostdatenbank:

```
catalog dcs database nydb1 as TELAVIV parms ",,,,,,,,,BIDI=62245"
```

Hiermit wird DB2 Connect angewiesen, die CCSID 424 der DB2-Hostdatenbank durch 62245 zu überschreiben. Diese Überschreibung umfasst die folgenden Verarbeitungsschritte:

1. DB2 Connect stellt eine Verbindung zur DB2-Hostdatenbank her und verwendet dabei die CCSID 62209 (BIDI-Zeichenfolgetyp 10).
2. DB2 Connect führt eine BIDI-Layoutumsetzung für die Daten durch, die an die DB2-Hostdatenbank gesendet werden sollen. Dabei wird CCSID 62213 (BIDI-Zeichenfolgetyp 5) in CCSID 62209 (BIDI-Zeichenfolgetyp 10) umgesetzt.
3. DB2 Connect führt eine BIDI-Layoutumsetzung für die Daten durch, die es von der DB2-Hostdatenbank empfängt. Dabei wird CCSID 62245 (BIDI-Zeichenfolgetyp 10) in CCSID 62213 (BIDI-Zeichenfolgetyp 5) umgesetzt.

Anmerkung:

1. Die Umgebungsvariable bzw. der Registrierungswert DB2BIDI muss auf YES gesetzt werden, damit der Parameter BIDI angewendet wird. DB2BIDI muss auf der DB2 Connect-Workstation definiert werden, auf der der DCS-Datenbankverzeichniseintrag katalogisiert ist. Für Anwendungen, die auf einem fernen Client eines DB2 Connect-Servers ausgeführt werden, muss die DB2BIDI-Variable auch auf dem jeweiligen Client definiert werden.
2. Wenn DB2 Connect eine Layoutumsetzung für die Daten ausführen soll, die an die DB2-Hostdatenbank gesendet werden sollen, ohne dass die entsprechende ID für codierten Zeichensatz überschrieben werden muss, müssen Sie trotzdem den BIDI-Parameter im Feld PARMS des DCS-Datenbankverzeichnisses hinzufügen. In diesem Fall müssen Sie den CCSID-Standardwert der DB2-Hostdatenbank angeben.

- In einigen Fällen wird die SQL-Abfrage durch eine bidirektionale ID für codierten Zeichensatz derart geändert, dass sie vom DB2-Server nicht erkannt wird. Versuchen Sie, vor allem IDs für codierten Zeichensatz des Typs IMPLICIT CONTEXTUAL und IMPLICIT RIGHT-TO-LEFT zu vermeiden, wenn eine andere Zeichenfolgenart verwendet werden kann. Bei IDs für codierten Zeichensatz des Typs CONTEXTUAL kann es zu unvorhersehbaren Ergebnissen kommen, wenn die SQL-Abfrage Zeichenfolgen in Anführungszeichen enthält. Verwenden Sie in SQL-Anweisungen keine Zeichenfolgen in Anführungszeichen, sondern stattdessen Hostvariablen, wo möglich.

Wenn eine bestimmte bidirektionale ID für codierten Zeichensatz Probleme verursacht, die durch diese Empfehlungen nicht behoben werden können, setzen Sie die Umgebungsvariable oder den Registrierungsdatenbankwert DB2BIDI auf NO.

Angeben von Parameterzeichenfolgen

Nachfolgend sind Beispiele für DCS-Parameter aufgeführt (jede Zeile stellt eine Parametergruppe dar):

```
NOMAP
/u/username/sql1lib/map/dcs1new.map,D
,D
,,INTERRUPT_ENABLED
NOMAP,D,INTERRUPT_ENABLED,,,SYSPLEX,LOCALDATE="JJMMTT",,
```

Sie können aber auch die Standardwerte akzeptieren, indem Sie keine Parameterzeichenfolge angeben.

Anmerkung: Verwenden Sie das Escapezeichen "\" (umgekehrter Schrägstrich) des Betriebssystems, wenn Sie den Befehlszeilenprozessor in der Betriebssystembefehlszeile von UNIX-Systemen verwenden. Dies ist erforderlich, da bei der Angabe der Maske LOCALDATE in der Parameterzeichenfolge zwei Paar doppelte Anführungszeichen angegeben werden müssen. Beispiel:

```
db2 catalog dcs db x as y parms "\",,,,,,LOCALDATE=\"\"JJMMTT\"\"\\""
```

Dies ergibt folgenden DCS-Verzeichniseintrag:

Eintrag für DCS 1:

```
Name der lokalen Datenbank           = X
Name der Zieldatenbank               = Y
Name des Anwendungsrequesters        =
DCS-Parameter                        = ,,,,,,LOCALDATE="JJMMTT"
Kommentar                             =
Release-Stand des DCS-Verzeichnisses = 0x0100
```

Kapitel 5. DB2 Connect - Sicherheit

Bei der Verwendung von DB2 Connect spielt die Authentifizierung von Benutzern eine wichtige Rolle, da die Benutzer für DB2 Connect und die Datenbank mit den Daten, auf die zugegriffen werden soll, sowohl lokal als auch fern sein können. Gesicherte Verbindungen und Kerberos-Unterstützung werden zusammen mit Sicherheitsaspekten für Datenbanken auf Hostmaschinen beschrieben.

Gesicherte Verbindungen über DB2 Connect

Einige DB2-Datenbankserver unterstützen gesicherte Kontexte. Mit einem *gesicherten Kontext* kann der Datenbankadministrator unter anderem Bedingungen für die Erstellung einer gesicherten Verbindung durch eine Clientanwendung definieren. Eine *gesicherte Verbindung* verfügt über Rechte, die eine normale Verbindung nicht hat.

Es gibt zwei Typen von gesicherten Verbindungen: implizite und explizite Verbindungen. Ob es sich bei der Erstellung einer Verbindung um eine explizite gesicherte Verbindung, eine implizite gesicherte Verbindung oder eine normale Verbindung handelt, hängt davon ab, ob eine gesicherte Verbindung angefordert wird und ob die Verbindung die im gesicherten Kontext auf dem Server definierten Kriterien erfüllt (siehe Tabelle 4).

Tabelle 4. Welche Verbindungstypen aus verschiedenen Kombinationen von Aktionen resultieren

	Die Verbindung erfüllt die Kriterien des Servers	Die Verbindung erfüllt die Kriterien des Servers nicht
Sie fordern eine gesicherte Verbindung an.	Explizit gesicherte Verbindung	Normale Verbindung und Rückgabe der Warnung SQL20360W (SQLSTATE-Wert 01679)
Sie fordern keine gesicherte Verbindung an.	Implizit gesicherte Verbindung	Normale Verbindung

Der Unterschied zwischen einer *implizit gesicherten Verbindung* und einer normalen Verbindung liegt darin, dass die implizit gesicherte Verbindung dem Benutzer für die Dauer der Verbindung temporäre Zugriffsrechte für Rollen erteilt. Die erteilten Zugriffsrechte für Rollen (sofern vorhanden) sind in dem gesicherten Kontext angegeben, der die Verbindung als gesichert definiert hat.

Implizit gesicherte Verbindungen können von allen Anwendungen erstellt werden, die zur Herstellung der Verbindung DB2 Connect verwenden. Implizit gesicherte Verbindungen werden wie normale Verbindungen erstellt und verwendet. Das bedeutet, dass für vorhandene Anwendungen keine Codeänderungen erforderlich sind, sofern die Anwendung die Verbindung mit DB2 Connect herstellt.

Die Vorgehensweise bei der Erteilung von temporären Zugriffsrechten für Rollen ist für *explizit gesicherte Verbindungen* und implizit gesicherte Verbindungen identisch. Mit einer explizit gesicherten Verbindung können Sie darüber hinaus die Berechtigungs-ID wechseln, während über diese Verbindung Aktionen ausgeführt werden. Das Wechseln der Berechtigungs-ID über eine explizit gesicherte Verbindung wird als *Benutzerwechsel* bezeichnet. Im gesicherten Kontext, der die Erstel-

lung der gesicherten Verbindung zulässt, wird die betreffende Berechtigungs-ID definiert und außerdem festgelegt, ob für diese Berechtigungs-ID ein Kennwort erforderlich ist.

Mit einem Benutzerwechsel lässt sich der Systemaufwand deutlich reduzieren, wenn mehrere Benutzer gemeinsam eine Verbindung verwenden. Das gilt insbesondere für Benutzernamen, die kein Kennwort erfordern, weil der Datenserver in einem solchen Fall die Berechtigungs-ID nicht authentifizieren muss. Bei Verwendung dieser Funktion müssen Sie jedoch sicherstellen, dass die Anwendung ohne Prüfung und Authentifizierung keinen Wechsel der Berechtigungs-ID zulässt. Die Folge wäre sonst eine Lücke im Sicherheitssystem.

Explizit gesicherte Verbindungen können erstellt und Benutzer gewechselt werden, wenn die Verbindung über DB2 Connect mithilfe von CLI oder JDBC hergestellt wird; das gilt auch für mit XA hergestellte Verbindungen. Für die Erstellung einer explizit gesicherten Verbindung und den Benutzerwechsel müssen spezielle Verbindungsattribute festgelegt werden. Vorhandene Anwendungen müssen also modifiziert werden, um explizit gesicherte Verbindungen nutzen zu können.

Abgesehen von den genannten Unterschieden können gesicherte Verbindungen (implizit und explizit) wie normale Verbindungen verwendet werden. Wenn eine explizit gesicherte Verbindung nicht mehr benötigt wird, muss diese jedoch explizit beendet werden, auch wenn die Verbindung bereits unterbrochen wurde. Andernfalls werden von der Verbindung verwendete Ressourcen möglicherweise nicht freigegeben. Dieses Problem tritt bei implizit gesicherten Verbindungen nicht auf.

Anmerkung:

1.

Wichtig: Bei einem Benutzerwechsel ohne Kennwortangabe wird die Authentifizierung des Datenbankservers übergangen. Die Anwendung darf einen Wechsel zu einer Berechtigungs-ID ohne Kennwortangabe nur dann zulassen, wenn die Anwendung die Berechtigungs-ID bereits geprüft und authentifiziert hat. Andernfalls entsteht eine Lücke im Sicherheitssystem.

2. Der Authentifizierungstyp CLIENT sollte von explizit gesicherten Verbindungen nicht verwendet werden. Dies gilt nicht für implizit gesicherte Verbindungen.
3. Anwendungen, die explizit gesicherte Verbindungen verwenden, sollten auf sicheren Maschinen ausgeführt werden, die kennwortgeschützt sind und nur von berechtigten Mitarbeitern verwendet werden. Dies gilt nicht für implizit gesicherte Verbindungen.

Erstellen und Beenden einer gesicherten Verbindung über CLI

Bei Verwendung von CLI können Sie eine explizit gesicherte Verbindung erstellen, wenn der Datenbankserver, zu dem die Verbindung hergestellt wird, entsprechend konfiguriert ist.

Diese Vorgehensweise setzt voraus, dass kein XA-Transaktionsmanager verwendet wird. Wenn Sie einen XA-Transaktionsmanager verwenden, müssen Sie lediglich sicherstellen, dass der Konfigurationswert TCTX bei Aufruf von xa_open auf TRUE gesetzt wird. Wenn diese Voraussetzung erfüllt ist, kann jede geeignete Verbindung als explizit gesicherte Verbindung definiert werden. Wenn Sie prüfen möchten, ob eine Verbindung eine explizit gesicherte Verbindung ist, lesen Sie die Informationen in Schritt 3.

- Die Datenbank, zu der Sie die Verbindung herstellen, muss gesicherte Kontexte unterstützen.
- Es muss ein gesicherter Kontext definiert sein, der den Client als vertrauenswürdig erkennt.
- Sie müssen die Systemberechtigungs-ID kennen, die im gesicherten Kontext angegeben ist. Die Systemberechtigungs-ID einer gesicherten Verbindung ist die Berechtigungs-ID, die bei Erstellung der Verbindung als Benutzername auf dem Server angegeben wird. Damit die Verbindung von einem bestimmten gesicherten Kontext als vertrauenswürdig anerkannt wird, muss die Systemberechtigungs-ID in diesem gesicherten Kontext angegeben sein. Eine gültige Systemberechtigungs-ID und das zugehörige Kennwort erhalten Sie vom zuständigen Sicherheitsadministrator.

Die Beispiele in diesen Anweisungen verwenden die Programmiersprache C. 'verbindung' ist ein Zeiger auf eine gültige Verbindungskennung, für die aber noch keine Verbindung hergestellt wurde. Die Variable 'rc' ist vom Datentyp SQL-RETURN.

1. Legen Sie neben den Einstellungen für die Verbindungsattribute, die Sie auch bei einer normalen Verbindung vornehmen würden, für das Verbindungsattribut SQL_ATTR_USE_TRUSTED_CONTEXT den Wert SQL_TRUE fest. Dazu rufen Sie die Funktion SQLSetConnectAttr auf.

```
rc = SQLSetConnectAttr(
    verbindung,
    SQL_ATTR_USE_TRUSTED_CONTEXT, SQL_TRUE, SQL_IS_INTEGER
);
```

2. Stellen Sie eine Verbindung zur Datenbank her. Gehen Sie dabei wie bei einer normalen Verbindung vor, indem Sie zum Beispiel die Funktion SQLConnect aufrufen. Verwenden Sie die Systemberechtigungs-ID als Benutzernamen und das zugehörige Kennwort als Kennwort. Achten Sie auf Fehler und Warnungen, besonders solche, die in Tabelle 5 aufgeführt sind.

Tabelle 5. Fehler, die darauf hinweisen, dass die Erstellung einer gesicherten Verbindung fehlschlug

SQLCODE	SQLSTATE	Erläuterung
SQL20360W	01679	Die Verbindung konnte nicht als gesicherte Verbindung hergestellt werden. Stattdessen wurde eine normale Verbindung hergestellt.

Die Verbindung wurde hergestellt und ist eine explizit gesicherte Verbindung, sofern keine anderslautenden Fehler oder Warnungen auftreten.

3. (Optional) Sie können prüfen, ob es sich bei einer hergestellten Verbindung um eine explizit gesicherte Verbindung handelt, indem Sie den Wert des Verbindungsattributs SQL_ATTR_USE_TRUSTED_CONTEXT unter Verwendung der Funktion SQLGetConnectAttr überprüfen. Wenn der Wert SQL_TRUE lautet, handelt es sich um eine explizit gesicherte Verbindung.
4. Wenn eine Verbindung nicht mehr benötigt wird, muss diese auf jeden Fall explizit beendet werden, auch wenn die Verbindung bereits unterbrochen wurde. Wenn Sie eine explizit gesicherte Verbindung nicht explizit beenden, werden möglicherweise bestimmte von der Verbindung verwendete Ressourcen nicht freigegeben.

Anmerkung:

1. Der Authentifizierungstyp CLIENT sollte von explizit gesicherten Verbindungen nicht verwendet werden. Dies gilt nicht für implizit gesicherte Verbindungen.

2. Anwendungen, die explizit gesicherte Verbindungen verwenden, sollten nur auf sicheren Computern ausgeführt werden, die kennwortgeschützt sind und nur von berechtigten Mitarbeitern verwendet werden. Dies gilt nicht für implizit gesicherte Verbindungen.

Wechseln von Benutzern über eine gesicherte Verbindung mit CLI

Benutzer können über eine gesicherte Verbindung mithilfe der Befehlszeilenschnittstelle (CLI) gewechselt werden. Informationen zum Wechseln von Benutzern können Sie über den entsprechenden Link in der Liste der zugehörigen Links aufrufen.

- Die Verbindung wurde erfolgreich als explizit gesicherte Verbindung erstellt.
- Die gesicherte Verbindung darf sich nicht in einer Transaktion befinden.
- Der gesicherte Kontext, der die Erstellung der explizit gesicherten Verbindung ermöglichte, muss so konfiguriert sein, dass ein Wechsel zur gewünschten Berechtigungs-ID möglich ist.

Die Beispiele in diesen Anweisungen verwenden die Programmiersprache C. **verbindung** ist ein Zeiger auf eine aktive, explizit gesicherte Verbindung. Die Variable *rc* ist vom Datentyp SQLRETURN. Die Variable *neuerbenutzer* ist ein Zeiger auf eine Zeichenfolge, die die Berechtigungs-ID enthält, zu der Sie wechseln möchten. Die Variable *kennwort* ist ein Zeiger auf eine Zeichenfolge, die das Kennwort für die Berechtigungs-ID enthält.

1. Rufen Sie die Funktion SQLSetConnectAttr auf, und legen Sie einen Wert für das Attribut SQL_ATTR_TRUSTED_CONTEXT_USERID fest. Geben Sie die Berechtigungs-ID an, zu der Sie wechseln möchten.

```
rc = SQLSetConnectAttr(
    verbindung,
    SQL_ATTR_TRUSTED_CONTEXT_USERID, neuerbenutzer, SQL_NTS
);
//Prüfen Sie, ob Fehler aufgetreten sind.
```

Achten Sie auf Fehler und Warnungen, besonders solche, die in Tabelle 6 aufgeführt sind.

Tabelle 6. Fehler, die darauf hinweisen, dass beim Benutzerwechsel keine neue Berechtigungs-ID festgelegt wurde

SQLCODE	Erläuterung
CLI0106E	Die Verbindung wurde nicht hergestellt.
CLI0197E	Die Verbindung ist keine gesicherte Verbindung.
CLI0124E	Der angegebene Wert ist fehlerhaft. Der Wert darf z. B. nicht null sein, und er darf nicht zu lang sein.
CLI0196E	Die Verbindung ist an einer UOW (Unit of Work, Arbeitseinheit) beteiligt, die den Benutzerwechsel verhindert. Für einen Benutzerwechsel darf sich die Verbindung nicht in einer Transaktion befinden.

2. (Optional, sofern der gesicherte Kontext, der die gesicherte Verbindung zugelassen hat, kein Kennwort für die neue Berechtigungs-ID erfordert) Rufen Sie die Funktion SQLSetConnectAttr auf, und legen Sie einen Wert für das Attribut SQL_ATTR_TRUSTED_CONTEXT_PASSWORD fest. Geben Sie als Wert das Kennwort für die neue Berechtigungs-ID an.


```
rc = SQLSetConnectAttr(
    verbindung,
    SQL_ATTR_TRUSTED_CONTEXT_PASSWORD, kennwort, SQL_NTS
);
//Prüfen Sie, ob Fehler aufgetreten sind.
```

Achten Sie auf Fehler und Warnungen, die in Tabelle 6 auf Seite 48 und in Tabelle 7 aufgelistet sind.

Tabelle 7. Fehler, die darauf hinweisen, dass beim Benutzerwechsel kein neues Kennwort festgelegt wurde

SQLCODE	Erläuterung
CLI0198E	Es wurde noch kein Wert für das Attribut SQL_ATTR_TRUSTED_CONTEXT_USERID festgelegt.

- Fahren Sie wie bei einer normalen Verbindung fort. Wenn Sie einen XA-Transaktionsmanager verwenden, wird der Benutzerwechsel im Rahmen der nächsten Anforderung versucht. Andernfalls wird der Benutzerwechsel kurz vor Einleitung des nächsten Funktionsaufrufs versucht, der auf die Datenbank zugreift (z. B. SQLExecDirect). Prüfen Sie in beiden Fällen, ob Fehler und Warnungen aufgetreten sind, und achten Sie insbesondere auf die in Tabelle 8 aufgelisteten Fehler. Die Fehler in Tabelle 8 weisen darauf hin, dass der Benutzerwechsel fehlgeschlagen ist.

Tabelle 8. Fehler, die sich auf ein Fehlschlagen des Benutzerwechsels beziehen

SQLCODE	Erläuterung
SQL1046N	Der gesicherte Kontext, der die Erstellung der gesicherten Verbindung ermöglichte, ist nicht für einen Wechsel zur gewünschten Berechtigungs-ID konfiguriert. Die Berechtigungs-ID kann erst dann gewechselt werden, wenn der gesicherte Kontext geändert wird.
SQL30082N	Das angegebene Kennwort ist für die gewünschte Berechtigungs-ID ungültig.
SQL0969N mit dem nativen Fehler -20361	Es liegt eine Einschränkung auf Datenbankebene vor, die den Benutzerwechsel verhindert.

Wenn der Benutzerwechsel fehlschlägt, befindet sich die Verbindung im Status 'Nicht verbunden', bis erfolgreich zu einem anderen Benutzer gewechselt wird. Benutzer können über eine gesicherte Verbindung im Status 'Nicht verbunden' gewechselt werden; ein Zugriff auf den Datenbankserver ist jedoch nicht möglich. Eine Verbindung im Status 'Nicht verbunden' bleibt in diesem Status, bis der Benutzerwechsel erfolgreich war.

Anmerkungen:

- Wichtig:** Bei einem Benutzerwechsel ohne Kennwortangabe wird die Authentifizierung des Datenbankservers übergangen. Die Anwendung darf einen Wechsel zu einer Berechtigungs-ID ohne Kennwortangabe nur dann zulassen, wenn die Anwendung die Berechtigungs-ID bereits geprüft und authentifiziert hat. Andernfalls entsteht eine Lücke im Sicherheitssystem.
- Die Angabe eines Nullwerts für das Attribut SQL_ATTR_TRUSTED_CONTEXT_USERID entspricht der Angabe der Systemberechtigungs-ID für den gesicherten Kontext (die bei der Erstellung der explizit gesicherten Verbindung verwendete Benutzer-ID).

3. Wenn Sie den Wert des Verbindungsattributs `SQL_ATTR_TRUSTED_CONTEXT_USERID` für eine explizit gesicherte Verbindung erfolgreich festlegen, wird die Verbindung sofort zurückgesetzt. Die Verbindung entspricht dann einer neuen Verbindung, die mit den ursprünglichen Verbindungsattributen dieser Verbindung erstellt wird. Das Zurücksetzen findet auch dann statt, wenn der Wert, auf den Sie das Verbindungsattribut setzen, die Systemberechtigungs-ID, null oder der aktuelle Wert des Attributs ist.
4. Wenn das Attribut `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` festgelegt wurde, wird das Kennwort während der Verarbeitung des Benutzerwechsels authentifiziert, auch wenn der gesicherte Kontext, der die gesicherte Verbindung ermöglichte, bei einem Benutzerwechsel für diese Berechtigungs-ID keine Authentifizierung erfordert. Daraus resultiert ein unnötiger Systemaufwand. Diese Regel gilt nicht für die Systemberechtigungs-ID des gesicherten Kontexts. Wenn ein Wechsel zur Systemberechtigungs-ID des gesicherten Kontexts erfolgt und dazu keine Authentifizierung erforderlich ist, wird die Systemberechtigungs-ID nicht authentifiziert, auch wenn ein Kennwort angegeben wurde.

Überlegungen zur DB2-Authentifizierung

Als DB2 Connect-Administrator können Sie zusammen mit Ihrem Datenbankadministrator für den Host oder das System *i* festlegen, wo Benutzernamen und Kennwörter auf ihre Gültigkeit hin überprüft werden sollen.

- Auf dem Client
- Auf dem Host- oder System *i*-Server
- Einmalige Anmeldung und Gültigkeitsprüfung über ein System anderer Lieferanten (Kerberos)

Anmerkung: Wurde für den fernen Client kein Authentifizierungstyp angegeben, nimmt der Client standardmäßig den Wert `SERVER_ENCRYPT` an. Wird dieser Typ vom Server nicht akzeptiert, versucht der Client, die Aktion unter Verwendung eines vom Server zurückgegebenen geeigneten Werts zu wiederholen. Geben Sie zwecks Leistungsoptimierung stets den Authentifizierungstyp auf dem Client an, um diesen zusätzlichen Datenfluss über das Netzwerk zu vermeiden.

Ab DB2 Connect Version 8.2.2 (entspricht Version 8.1 FixPak 9) ist das Gateway kein passiver Teilnehmer der Authentifizierungsvereinbarung mehr. Stattdessen übernimmt das Gateway eine aktive Rolle. Der im Datenbankverzeichniseintrag angegebene Authentifizierungstyp überschreibt den auf dem Client katalogisierten Authentifizierungstyp. Client, Gateway und Server müssen kompatible Typen angeben. Wurde der katalogisierte Authentifizierungstyp auf dem Gateway im Datenbankverzeichniseintrag nicht angegeben, ist `SERVER` der für den Server erforderliche Standardauthentifizierungstyp. Die Vereinbarung zwischen Client und Server findet auch dann statt, wenn der Server den Authentifizierungstyp `SERVER` nicht unterstützt. Dieses Verhalten unterscheidet sich vom Client, der standardmäßig den Wert `SERVER_ENCRYPT` annimmt, wenn kein Authentifizierungstyp angegeben wurde.

Der auf dem Gateway katalogisierte Authentifizierungstyp wird nicht verwendet, wenn auf dem Client `DB2NODE` oder die Option `SQL_CONNECT_NODE` von `SET CLIENT API` festgelegt wurde. In diesen Fällen findet die Vereinbarung ausschließlich zwischen Client und Server statt.

Die folgenden Authentifizierungstypen sind für DB2 Connect zulässig:

CLIENT

Der Benutzername und das Kennwort werden auf dem Client überprüft.

SERVER

Der Benutzername und das Kennwort werden auf dem Host- oder System i-Datenbankserver geprüft.

SERVER_ENCRYPT

Wie beim Authentifizierungstyp SERVER werden der Benutzername und das Kennwort auf dem Host- oder System i-Datenbankserver geprüft. Die übertragenen Kennwörter werden jedoch auf dem Client verschlüsselt.

DATA_ENCRYPT

Ermöglicht die Verschlüsselung von Benutzerdaten während der Client/Serverkommunikation.

KERBEROS

Ermöglicht dem Client, sich über die Kerberos-Authentifizierung am Server anzumelden, anstatt die herkömmliche Kombination aus ID und Kennwort zu verwenden. Dieser Authentifizierungstyp setzt voraus, dass sowohl der Server als auch der Client für Kerberos aktiviert ist.

Das Einzigartige an der Authentifizierung über Kerberos besteht darin, dass der Client keine Benutzer-ID und kein Kennwort direkt an den Server übergeben muss. Stattdessen fungiert Kerberos als Authentifizierungsmechanismus Dritter. Der Benutzer gibt am Client-Terminal einmal eine ID und ein Kennwort ein, und Kerberos überprüft diese Anmeldung auf ihre Gültigkeit hin. Anschließend übergibt Kerberos die Berechtigung des betreffenden Benutzers automatisch und sicher an alle angeforderten lokalen Services und Netzwerkservices. Dies bedeutet, dass der Benutzer seine ID und sein Kennwort zum Anmelden an einem fernen DB2-Server nicht erneut einzugeben braucht. Die von der Kerberos-Authentifizierung zur Verfügung gestellte Funktion der einmaligen Anmeldung setzt voraus, dass sowohl DB2 Connect als auch der Datenbankserver, zu dem eine Verbindung hergestellt wird, Kerberos-Unterstützung bereitstellen.

Anmerkung: Für den Authentifizierungstyp GSSPLUGIN ist keine Unterstützung verfügbar.

Kerberos-Unterstützung

Die Kerberos-Schicht für die Authentifizierung, die das Ticketsystem verwaltet, ist in den Mechanismus des aktiven Verzeichnisses (Active Directory) von Windows 2000 integriert. Die Client- und Serverseite einer Anwendung kommunizieren mit den Client- bzw. Servermodulen von Kerberos SSP (Security Support Provider). Die SSP-Schnittstelle (SSPI) ist eine leistungsfähige Schnittstelle zu Kerberos SSP und anderen Sicherheitsprotokollen.

Typische Konfiguration

Um DB2 für die Verwendung der Authentifizierung mit Kerberos zu konfigurieren, ist Folgendes zu definieren:

- Eine Berechtigungsrichtlinie für DB2 (als Service) in dem in einem Netzwerk gemeinsam benutzten aktiven Verzeichnis (Active Directory) sowie
- Eine Trust-Beziehung zwischen Kerberos-KDCs (Key Distribution Centers)

Im einfachsten Szenario ist mindestens eine KDC-Trust-Beziehung zu konfigurieren, d. h. die Beziehung zwischen dem KDC zur Steuerung der Client-Workstation und dem System i-, OS/390- oder z/OS-System. OS/390 Version 2 Release 10 oder

z/OS Version 1 Release stellt die Verarbeitung von Kerberos-Tickets über die entsprechende RACF-Funktion zur Verfügung, die es dem Host ermöglicht, als UNIX-KDC zu fungieren.

DB2 Connect stellt wie gewohnt die Router-Funktionalität in der dreischichtigen Konfiguration zur Verfügung. Wird die Kerberos-Sicherheit verwendet, übernimmt DB2 Connect im Rahmen der Authentifizierung keinen Aufgabenbereich, sondern übergibt lediglich das Sicherheitstoken des Clients an DB2 für i5/OS oder an DB2 für OS/390 und z/OS. Der DB2 Connect-Gateway braucht nicht Teil des Clients oder des Kerberos-Realms des Hosts zu sein.

Kompatibilität auf PC-Ebene

DB2-Minimalanforderungen für die Kerberos-Unterstützung sind die folgenden:

IBM Data Server-Client:

Version 8

DB2 Connect:

Version 8

DB2 Universal Database (UDB) für OS/390 und z/OS:

Version 7

Hinweise und Tipps zur Sicherheit unter OS/390 und z/OS

Dieser Abschnitt enthält einige Sicherheitshinweise und Tipps für Verbindungen von DB2 Connect zu einem Datenbankserver unter DB2 für OS/390 und z/OS.

Feld für erweiterte Sicherheit

Stellen Sie sicher, dass das Feld für erweiterte Sicherheit (Extended Security Field) unter DB2 OS/390 und z/OS auf YES eingestellt ist. Dieses Feld befindet sich in der Anzeige DSNTIPR von DB2 für OS/390 und z/OS.

Erweiterte Sicherheitscodes

Bis DB2 Universal Database für z/OS und OS/390 Version 5.1 konnte es vorkommen, dass Verbindungsanforderungen, die Benutzer-IDs oder Kennwörter angaben, mit SQL30082 Ursachencode 0 fehlschlugen, aber keine andere Nachricht mit Angaben zur Fehlerursache ausgegeben wurde.

In DB2 Universal Database für z/OS und OS/390 Version 5.1 wurde eine Erweiterung eingeführt, die Unterstützung für erweiterte Sicherheitscodes zur Verfügung stellt. Durch das Angeben erweiterter Sicherheitscodes werden zusätzliche Diagnoseinformationen, beispielsweise Informationen zu einem abgelaufenen Kennwort (PASSWORD EXPIRED), zusätzlich zum Ursachencode geliefert.

Um diese Möglichkeiten nutzen zu können, muss der DB2 Universal Database für z/OS und OS/390-Installationsparameter ZPARM für erweiterte Sicherheit auf YES eingestellt werden. Mit der Installationsanzeige DSN6SYSP von DB2 Universal Database für z/OS und OS/390 kann EXTSEC=YES eingestellt werden. Sie können dafür auch DDF-Anzeige 1 (DSNTIPR) verwenden. Der Standardwert ist EXTSEC=NO. Falls das Kennwort nicht mehr gültig ist, erhalten Windows-, Linux-, UNIX- und Webanwendungen, die DB2 Connect verwenden, die Fehlernachricht SQL30082.

TCP/IP-Sicherheit bereits geprüft

Wenn Sie Unterstützung für die Sicherheitsoption AUTHENTICATION=CLIENT von DB2 wünschen, geben Sie in der Installationsanzeige DSNTIP4 von DB2 Universal Database für z/OS und OS/390 (DDF-Anzeige 2) mit YES an, dass die TCP/IP-Sicherheit bereits überprüft ist.

Sicherheit für ODBC- und Java-Anwendungen auf Workstations

ODBC- und Java-Anwendungen auf Workstations verwenden dynamisches SQL. Dies kann bei einigen Installationen Sicherheitsfragen aufwerfen. DB2 Universal Database für z/OS und OS/390 führt eine neue Bindeoption, DYNAMICRULES(BIND), ein, die die Ausführung von dynamischem SQL unter der Berechtigung des Eigners oder des Binders zulässt.

DB2 und DB2 Connect bieten einen neuen CLI/ODBC-Konfigurationsparameter, CURRENTPACKAGESET, in der Konfigurationsdatei DB2CLI.INI. Dieser sollte auf einen Schemennamen eingestellt werden, der die geeigneten Zugriffsrechte hat. Eine SQL-Anweisung SET CURRENT PACKAGESET schema wird automatisch nach jeder Verbindungsherstellung für die Anwendung abgesetzt.

Mit dem ODBC-Manager können Sie DB2CLI.INI aktualisieren.

Unterstützung für Kennwortänderungen

Wenn das Kennwort einer Benutzer-ID abgelaufen ist, gibt die SQL-Anweisung CONNECT eine Fehlernachricht zurück, z. B. SQLCODE -30082 Ursachencode 1. DB2 Connect erlaubt die Änderung des Kennworts über Remotezugriff. Mithilfe von DRDA kann DB2 Universal Database für z/OS und OS/390 das Kennwort für Sie ändern. Setzen Sie dazu die folgende Anweisung CONNECT ab:

```
CONNECT TO <datenbank> USER <benutzer-id> USING <kennwort>  
NEW <neues_kennwort> CONFIRM <neues_kennwort>
```

Sie können das Kennwort auch mit dem Dialog "Kennwort ändern" des DB2-Konfigurationsassistenten ändern.

Mit DB2 Connect unterstützte Sicherheitseinstufungen

In diesem Abschnitt werden die verschiedenen Kombinationen von Authentifizierungs- und Sicherheitseinstellungen aufgelistet, die von DB2 Connect unterstützt werden.

Sicherheitseinstufungen für TCP/IP-Verbindungen

Das Übertragungsprotokoll TCP/IP unterstützt keine Sicherheitsoptionen auf der Netzwerkprotokollebene. Der Authentifizierungstyp legt fest, wo die Authentifizierung stattfindet. Nur die in der Tabelle aufgeführten Kombinationen werden von DB2 Connect unterstützt. Die Authentifizierungseinstellungen befinden sich im Datenbankverzeichniseintrag auf dem DB2 Connect-Server.

Tabelle 9. Gültige Sicherheitsszenarios

Szenario	Authentifizierungseinstellung	Gültigkeitsprüfung
1	CLIENT	Client
2	SERVER	Host- oder System i-Datenbankserver
3	SERVER_ENCRYPT	Host- oder System i-Datenbankserver

Tabelle 9. Gültige Sicherheitsszenarios (Forts.)

Szenario	Authentifizierungseinstellung	Gültigkeitsprüfung
4	KERBEROS	Kerberos-Sicherheit
5	DATA_ENCRYPT	Host- oder System i-Datenbankserver

Erörterung der Sicherheitseinstufungen

Die folgenden Erörterungen gelten für die oben beschriebenen Verbindungen, die in Tabelle 9 auf Seite 53 aufgeführt sind. Jedes Szenario wird nachfolgend detaillierter beschrieben:

- In Szenario 1 werden der Benutzername und das Kennwort nur auf dem fernen Client überprüft. Bei einem lokalen Client werden Benutzername und Kennwort nur auf dem DB2 Connect-Server geprüft.
Es wird erwartet, dass der Benutzer an der Stelle authentifiziert wird, an der er sich das erste Mal anmeldet. Die Benutzer-ID wird über das Netzwerk gesendet, jedoch nicht das Kennwort. Diese Sicherheitseinstufung sollte nur verwendet werden, wenn alle Client-Workstations über entsprechende zuverlässige Sicherheitseinrichtungen verfügen.
- In Szenario 2 werden der Benutzername und das Kennwort nur auf dem Host- oder System i-Datenbankserver überprüft. Die Benutzer-ID und das Kennwort werden über das Netzwerk vom fernen Client an den DB2 Connect-Server und vom DB2 Connect-Server an den Host- oder System i-Datenbankserver gesendet.
- Szenario 3 entspricht Szenario 2, mit dem Unterschied, dass Benutzer-ID und Kennwort verschlüsselt werden.
- In Szenario 4 erhält der Client ein Kerberos-Ticket von Kerberos KDC. Das Ticket wird unverändert über DB2 Connect an den Server gesendet, wo es vom Server überprüft wird.
- Szenario 5 entspricht Szenario 3, mit dem Unterschied, dass die Benutzerdaten ebenfalls verschlüsselt werden.

Kapitel 6. Binden von Anwendungen und Dienstprogrammen (DB2 Connect)

Anwendungsprogramme, die mit eingebettetem SQL entwickelt wurden, müssen an jede Datenbank gebunden werden, mit der sie arbeiten sollen. Sofern diese Funktionen auf den entsprechenden Plattformen zur Verfügung stehen, können Sie dies über die Steuerzentrale und den Konfigurationsassistenten durchführen.

Das Binden muss einmal pro Anwendung für jede Datenbank durchgeführt werden. Bei dem Bindeprozess werden Datenbankzugriffspläne für jede SQL-Anweisung gespeichert, die ausgeführt wird. Diese Zugriffspläne werden von den Anwendungsentwicklern bereitgestellt und in *Bindedateien* gespeichert, die während der Vorkompilierung erstellt werden. Die Verarbeitung dieser Bindedateien durch einen Host- oder System i-Datenbankserver wird als Binden bezeichnet.

Da einige der mit DB2 Connect bereitgestellten Dienstprogramme mit eingebetteter SQL entwickelt wurden, müssen sie an einen Host- oder System i-Datenbankserver gebunden werden, bevor sie mit diesem System verwendet werden können. Wenn Sie die Dienstprogramme und Schnittstellen von DB2 Connect nicht verwenden, müssen sie auch nicht an Ihre Host- oder System i-Datenbankserver gebunden werden. Die folgenden Dateien enthalten die Liste der Bindedateien, die von diesen Dienstprogrammen benötigt werden:

- ddcsmvs.lst für OS/390 und z/OS
- ddcsvse.lst für VSE
- ddcsvm.lst für VM
- ddcs400.lst für OS/400

Durch Binden einer dieser Dateilisten an eine Datenbank wird jedes einzelne Dienstprogramm an die betreffende Datenbank gebunden.

Wenn ein DB2 Connect-Serverprodukt installiert ist, müssen die Dienstprogramme von DB2 Connect an jeden Host- oder System i-Datenbankserver gebunden werden, bevor sie mit dem entsprechenden System verwendet werden können. Wenn alle Clients auf demselben Fixpackstand sind, müssen die Dienstprogramme - unabhängig von der Anzahl der Clientplattformen - nur einmal gebunden werden.

Wenn Sie z. B. 10 Windows-Clients und 10 AIX-Clients haben, die über DB2 Connect Enterprise Server Edition auf einem Windows-Server mit DB2 Universal Database (UDB) für OS/390 und z/OS verbunden sind, gehen Sie wie folgt vor:

- Binden Sie ddcsmvs.lst von einem der Windows-Clients aus.
- Binden Sie ddcsmvs.lst von einem der AIX-Clients aus.
- Binden Sie ddcsmvs.lst vom DB2 Connect-Server aus.

Dieses Beispiel setzt Folgendes voraus:

- Alle Clients haben dieselbe Servicestufe. Ist dies nicht der Fall, müssen Sie sie unter Umständen zusätzlich von jedem Client einer bestimmten Servicestufe binden.
- Der Server ist auf derselben Servicestufe wie die Clients. Wenn das nicht der Fall ist, müssen Sie auch vom Server aus binden.

Zusätzlich zu Dienstprogrammen von DB2 Connect müssen alle anderen Anwendungen, die eingebettetes SQL verwenden, an jede Datenbank gebunden werden, mit der sie arbeiten sollen. Eine Anwendung, für die kein Bindevorgang ausgeführt wurde, erzeugt bei der Ausführung normalerweise die Fehlermeldung SQL0805N. Es ist in vielen Fällen sinnvoll, eine zusätzliche Bindelistendatei für alle Anwendungen zu erstellen, für die ein Bindevorgang erforderlich ist.

Führen Sie für jeden Host- oder System i-Datenbankserver, an den eine Anwendung gebunden wird, folgende Schritte aus:

1. Stellen Sie sicher, dass Sie über die erforderliche Berechtigung für das Verwaltungssystem des verwendeten Host- oder System i-Datenbankservers verfügen:

OS/390 oder z/OS

Folgende Berechtigungen sind erforderlich:

- SYSADM oder
- SYSCTRL oder
- BINDADD *und* CREATE IN COLLECTION NULLID

Anmerkung: Die Zugriffsrechte BINDADD und CREATE IN COLLECTION NULLID sind **nur** dann ausreichend, wenn die Pakete noch nicht existieren (z. B. wenn Sie sie zum ersten Mal erstellen).

Wenn sie bereits existieren und erneut gebunden werden, hängt die für diesen Vorgang bzw. diese Vorgänge erforderliche Berechtigung davon ab, wer den ursprünglichen Bindevorgang ausführte.

A) Wenn Sie selbst den ursprünglichen Bindevorgang ausführten und den Bindevorgang nun erneut ausführen, ist eine beliebige der oben aufgeführten Berechtigungen zum Ausführen des Bindevorgangs ausreichend.

B) Wenn ein anderer Benutzer den ursprünglichen Bindevorgang ausführte und Sie den zweiten Bindevorgang ausführen, wird hierfür die Berechtigung SYSADM oder die Berechtigung SYSCTRL benötigt. Mit den Berechtigungen BINDADD und CREATE IN COLLECTION NULLID allein können Sie den Bindeprozess nicht durchführen. Ein Paket kann auch dann erstellt werden, wenn weder die Berechtigung SYSADM noch die Berechtigung SYSCTRL vorhanden ist. In dieser Situation benötigen Sie die Berechtigung BIND für jedes der vorhandenen Pakete, das Sie ersetzen wollen.

VSE oder VM

Eine DBA-Berechtigung ist erforderlich. Wenn Sie die Option GRANT mit dem Befehl BIND verwenden wollen (um zu vermeiden, dass der Zugriff für jedes DB2 Connect-Paket einzeln erteilt werden muss), muss die Benutzer-ID NULLID die Berechtigung zum Erteilen der Berechtigung für die folgenden Tabellen an andere Benutzer haben:

- system.syscatalog
- system.syscolumns
- system.sysindexes
- system.systabauth
- system.syskeycols
- system.syssynonyms

- system.syskeys
- system.syscolauth

Auf dem VSE- oder VM-System kann folgender Befehl abgesetzt werden:

```
grant select on tabelle to nullid with grant option
```

OS/400

Berechtigung *CHANGE oder eine höhere Berechtigung in der Objektgruppe NULLID

2. Führen Sie Befehle aus, die den folgenden ähneln:

```
db2 connect to db-aliasname user benutzer-id using kennwort
db2 bind pfad@ddcsmvs.lst blocking all
      sqlerror continue messages ddcsmvs.msg grant public
db2 connect reset
```

Hierbei gilt: *db-aliasname*, *benutzer-id* und *kennwort* beziehen sich auf den Host- oder System i-Datenbankserver, 'ddcsmvs.lst' ist die Bindelistendatei für z/OS, und *pfad* ist die Speicherposition der Bindelistendatei.

Beispiel: *laufwerk*:\sql1lib\bnd\ gilt für alle Windows-Betriebssysteme und *INSTHOME*/sql1lib/bnd/ für alle Linux- und UNIX-Betriebssysteme. Hierbei ist *laufwerk* das logische Laufwerk, auf dem DB2 Connect installiert ist, und *INSTHOME* ist das Benutzerverzeichnis der DB2 Connect-Instanz.

Mit der Option GRANT des Befehls BIND kann das Zugriffsrecht EXECUTE an PUBLIC oder einen angegebenen Benutzernamen bzw. eine angegebene Gruppen-ID erteilt werden. Wenn die Option GRANT des Befehls BIND nicht verwendet wird, muss GRANT EXECUTE (RUN) für jeden einzelnen Fall ausgeführt werden.

Die Paketnamen für die Bindedateien können mit folgendem Befehl ermittelt werden:

```
ddcspkgn @bindedatei.lst
```

Beispiel:

```
ddcspkgn @ddcsmvs.lst
```

Dadurch würde die folgende Ausgabe erstellt:

Bindedatei	Paketname
f:\sql1lib\bnd\db2ajgrt.bnd	SQLAB6D3

Führen Sie zum Ermitteln dieser Werte für DB2 Connect das Dienstprogramm *ddcspkgn* aus, z. B.:

```
ddcspkgn @ddcsmvs.lst
```

Dieses Dienstprogramm kann wahlfrei zum Ermitteln des Paketnamens einzelner Bindedateien verwendet werden, z. B.:

```
ddcspkgn bindedat.bnd
```

Anmerkung:

- a. Die Verwendung der Bindeoption `sqlerror continue` ist erforderlich. Diese Option wird jedoch automatisch für Sie angegeben, wenn Sie Anwendungen mit DB2-Tools oder dem Befehlszeilenprozessor (CLP) binden. Diese Option wandelt Bindefehler in Warnungen um, sodass beim Binden einer Datei mit Fehlern immer noch ein Paket erstellt werden kann.

Dies erlaubt es wiederum, eine Bindedatei für mehrere Server zu verwenden, selbst wenn eine bestimmte Serverimplementierung die SQL-Syntax einer anderen als ungültig markiert. Daher ist zu erwarten, dass das Binden einer der Listendateien `ddcsxxx.lst` für einen bestimmten Host- oder System i-Datenbankserver einige Warnungen ergibt.

- b. Wenn Sie die Verbindung zu einer Datenbank von DB2 über DB2 Connect herstellen, verwenden Sie die Bindeliste `db2ubind.lst` und geben Sie nicht `sqlerror continue` an, da dies nur gültig ist, wenn eine Verbindung zu einem Host- oder System i-Datenbankserver hergestellt wird. Für die Verbindung zu einer Datenbank unter DB2 wird außerdem empfohlen, die mit DB2 ausgelieferten DB2-Clients und nicht DB2 Connect zu verwenden.
3. Verwenden Sie ähnliche Anweisungen, um jede Anwendung oder Liste von Anwendungen zu binden.
4. Wenn ferne Clients aus einem früheren Release von DB2 vorhanden sind, ist es möglicherweise erforderlich, die Dienstprogramme auf diesen Clients an DB2 Connect zu binden.

Kapitel 7. Aktualisierungen auf mehreren Systemen

Das Aktualisieren auf mehreren Systemen, auch als DUOW (Distributed Unit of Work, verteilte Arbeitseinheit) und zweiphasiges Commit bezeichnet, ist eine Funktion, die es Ihren Anwendungen ermöglicht, Daten auf mehreren fernen Datenbankservern zu aktualisieren und dabei deren Integrität zu gewährleisten. Ein Beispiel hierfür ist eine Banktransaktion, bei der Geld von einem Konto auf ein anderes auf einem anderen Datenbankserver übertragen wird.

Bei einer solchen Transaktion ist es wichtig, dass Aktualisierungen, die ein Konto belasten, erst festgeschrieben werden, nachdem die Aktualisierungen, die für die Verarbeitung der Gutschrift auf dem anderen Konto erforderlich sind, festgeschrieben wurden. Die Aktualisierung auf mehreren Systemen ist dann in Betracht zu ziehen, wenn die Daten für diese Konten auf zwei verschiedenen Datenbankservern verwaltet werden.

Die DB2-Produkte bieten eine umfassende Unterstützung für Aktualisierungen auf mehreren Systemen. Diese Unterstützung ist für Anwendungen verfügbar, die mit regulärer SQL entwickelt wurden, sowie für Anwendungen, die Transaktionsverarbeitungsmonitore (TP-Monitore) verwenden, die die X/Open XA-Schnittstellenspezifikation implementieren. TP-Monitore sind z. B. IBM TxSeries (CICS und Encina), IBM Message and Queuing Series, IBM Component Broker Series, IBM San Francisco Project sowie Microsoft Transaction Server (MTS), BEA Tuxedo und verschiedene andere. Je nachdem, ob für die Aktualisierung auf mehreren Systemen systemeigenes SQL oder ein TP-Monitor verwendet wird, variieren die Installationsanforderungen.

Sowohl die Verfahren, die für die Aktualisierung auf mehreren Systemen systemeigenes SQL verwenden, als auch die auf TP-Monitoren basierenden Programme müssen unter Angabe der Optionen `CONNECT 2 SYNCPOINT TWOPHASE` vorkompiliert werden. Beide Verfahren können über die SQL-Anweisung `CONNECT` angeben, welche Datenbank für die folgenden SQL-Anweisungen verwendet werden soll. Wenn kein TP-Monitor vorhanden ist, der DB2 mitteilt, dass er die Transaktion koordiniert (z. B., wenn DB2 die `xa_open`-Aufrufe des TP-Monitors zum Aufbau einer Datenbankverbindung empfängt), wird die Transaktion von der DB2-Software koordiniert.

Wenn Sie für die Aktualisierung auf mehreren Systemen einen TP-Monitor verwenden, muss von der Anwendung mithilfe der API des TP-Monitors, z. B. `CICS SYNCPOINT`, `Encina Abort()`, `MTS SetAbort()`, ein Commit oder ein Rollback angefordert werden. Bei der Aktualisierung auf mehreren Systemen mit systemeigenem SQL müssen die normalen SQL-Anweisungen `COMMIT` und `ROLLBACK` verwendet werden.

Die Aktualisierung auf mehreren Systemen mit einem TP-Monitor kann eine Transaktion koordinieren, die sowohl auf DB2- als auch auf Nicht-DB2-Ressourcenmanager wie Oracle, Informix oder SQLServer zugreift. Die Aktualisierung auf mehreren Systemen mit systemeigenem SQL wird nur mit DB2-Servern verwendet.

Damit eine Aktualisierungstransaktion auf mehreren Systemen durchgeführt werden kann, müssen alle Datenbanken, die an einer verteilten Transaktion beteiligt

sind, DUOWs unterstützen. Zum Zeitpunkt der Erstellung dieses Handbuchs unterstützen folgende DB2-Server DUOWs und können somit an verteilten Transaktionen beteiligt werden:

- DB2 für Linux, UNIX und Windows Version 8 oder höher.
- DB2 Universal Database (UDB) für OS/390 und z/OS Version 7
- DB2 für z/OS Version 8.
- Für DB2 für System i ist OS/400 Version 5 Release 1 oder höher erforderlich.

In einer verteilten Transaktion kann eine beliebige Zusammenstellung aus unterstützten Datenbankservern aktualisiert werden. So kann Ihre Anwendung beispielsweise mit einer einzigen Transaktion verschiedene Tabellen in einer Datenbank von DB2 unter Windows, einer Datenbank von DB2 für OS/390 und z/OS sowie einer Datenbank von DB2 für i5/OS aktualisieren.

Aktivieren von Aktualisierungen auf mehreren Systemen über die Steuerzentrale

Sie können Aktualisierungen auf mehreren Systemen über die Steuerzentrale durchführen.

Um Aktualisierungen auf mehreren Systemen zu aktivieren, gehen Sie wie folgt vor:

1. Starten Sie die Steuerzentrale.
2. Klicken Sie das Pluszeichen [+] an, um die Baumstruktursicht zu erweitern.
3. Klicken Sie nun mit der rechten Maustaste die Instanz an, die Sie konfigurieren wollen. Daraufhin wird ein Kontextmenü geöffnet.
4. Wählen Sie den Menüpunkt **Aktualisierung auf mehreren Systemen** —> **Konfigurieren** aus. Der Assistent für Aktualisierungen auf mehreren Systemen wird geöffnet.
5. Wählen Sie **Unten genannten TP-Monitor verwenden** aus, und geben Sie einen TP-Monitor (Transaktionsprogramm-Monitor) an. Dieses Feld enthält die von Ihnen aktivierten Standardwerte für den TP-Monitor. Wenn Sie keinen TP-Monitor verwenden möchten, wählen Sie **Keinen TP-Monitor verwenden** aus. Klicken Sie **Weiter** an.
6. Wenn Sie einen TP-Monitor verwenden, geben Sie die Einstellungen für den Synchronisationspunktmanager (SPM) an. Wenn Sie keinen TP-Monitor verwenden, geben Sie Ihre Transaktionsmanagerdatenbank an.
7. Klicken Sie **Fertig stellen** an.

Testen der Aktualisierung auf mehreren Systemen über die Steuerzentrale

Sie können Ihre Konfiguration für die Aktualisierung auf mehreren Systemen über die Steuerzentrale testen.

Um die Aktualisierung auf mehreren Systemen zu testen, gehen Sie wie folgt vor:

1. Klicken Sie die Instanz mit der rechten Maustaste an, und wählen Sie anschließend im Kontextmenü die Menüoption **Aktualisierung auf mehreren Systemen** —> **Test** aus. Das Fenster 'Aktualisierung auf mehreren Systemen testen' wird geöffnet.

2. Wählen Sie die zu testenden Datenbanken aus den verfügbaren Datenbanken aus, die im Listenfenster **Verfügbar** angezeigt werden. Mithilfe der in der Mitte angezeigten Pfeilkнопfe (> und >>) können Sie Ihre Auswahl aus dem bzw. in das Listenfenster **Ausgewählt** bewegen. Darüber hinaus können Sie die ausgewählte Benutzer-ID und das ausgewählte Kennwort ändern, indem Sie sie direkt im Listenfenster **Ausgewählt** editieren.
3. Wenn Sie Ihre endgültige Auswahl getroffen haben, klicken Sie **OK** an. Das Fenster 'Testergebnis für Aktualisierung auf mehreren Systemen' wird geöffnet.
4. Das Fenster 'Testergebnis für Aktualisierung auf mehreren Systemen' zeigt an, für welche der ausgewählten Datenbanken der Aktualisierungstest erfolgreich war und für welche er fehlschlug. Das Fenster zeigt für die fehlgeschlagenen Tests SQL-Codes und Fehlernachrichten an. Klicken Sie **Schließen** an, um das Fenster zu schließen.
5. Klicken Sie **Schließen** an, um das Fenster 'Aktualisierung auf mehreren Systemen testen' zu schließen.

Aktualisierungen auf mehreren Systemen und Synchronisationspunktmanager

Für Host- und System i-Datenbankserver ist DB2 Connect zur Teilnahme an einer verteilten Transaktion erforderlich, die von Linux-, Windows-, UNIX- oder Webanwendungen ausgeht. Zusätzlich erfordern viele Szenarios für die Aktualisierung auf mehreren Systemen, an denen Host- und System i-Datenbankserver beteiligt sind, die Konfiguration des Synchronisationspunktmanagers (SPM). Der DB2-SPM wird beim Erstellen einer DB2-Instanz automatisch mit Standardeinstellungen konfiguriert.

Ob der Synchronisationspunktmanager (SPM) tatsächlich benötigt wird, hängt von der Auswahl des Protokolls (TCP/IP) und der Verwendung des TP-Monitors ab. Eine Auflistung aller Szenarios, für die der SPM erforderlich ist, finden Sie in der folgenden Tabelle. Aus der Tabelle geht außerdem hervor, ob für den Zugriff auf den Host oder auf System i über Intel- oder UNIX-Maschinen DB2 Connect erforderlich ist. Der Synchronisationspunktmanager von DB2 Connect ist für eine Aktualisierung auf mehreren Systemen erforderlich, wenn ein TP-Monitor verwendet wird.

Tabelle 10. Szenarios für die Aktualisierung auf mehreren Systemen, für die SPM erforderlich ist - TCP/IP

Wird ein TP-Monitor verwendet?	Ist SPM erforderlich?	Erforderliches Produkt (wählen Sie eines aus)	Unterstützte Host- und System i-Datenbanken
Ja	Ja	DB2 Connect-Serverprodukt DB2 Enterprise Server Edition mit angewendeter DB2 Connect-Lizenz	DB2 Universal Database (UDB) für OS/390 und z/OS Version 7 DB2 UDB für z/OS Version 8 oder höher

Tabelle 10. Szenarios für die Aktualisierung auf mehreren Systemen, für die SPM erforderlich ist - TCP/IP (Forts.)

Wird ein TP-Monitor verwendet?	Ist SPM erforderlich?	Erforderliches Produkt (wählen Sie eines aus)	Unterstützte Host- und System i-Datenbanken
Nein	Nein	DB2 Connect Personal Edition DB2 Connect-Serverprodukt DB2 Enterprise Server Edition mit angewendeter DB2 Connect-Lizenz	DB2 UDB für OS/390 und z/OS Version 7 DB2 UDB für z/OS Version 8 oder höher

Anmerkung: In einer verteilten Transaktion kann eine beliebige Zusammenstellung aus unterstützten Datenbankservern aktualisiert werden. So kann Ihre Anwendung beispielsweise mit einer einzigen Transaktion verschiedene Tabellen in einer Datenbank von DB2 unter Windows, einer Datenbank von DB2 für OS/390 sowie einer Datenbank von DB2 für i5/OS aktualisieren.

Konfiguration von DB2 Connect mit einem XA-kompatiblen Transaktionsmanager

Dieser Abschnitt beschreibt die Konfigurationsschritte, die erforderlich sind, um S/390-, System i- und zSeries-Datenbankserver im Rahmen Ihres TP-Monitors verwenden zu können.

Sie müssen über einen betriebsbereiten TP-Monitor verfügen, DB2 Connect installiert haben sowie eine Verbindung zum Host- oder System i-Datenbankserver konfiguriert und getestet haben.

Um DB2 Connect für die Verwendung von S/390-, System i- und zSeries-Datenbankservern im Rahmen Ihres TP-Monitors zu konfigurieren, führen Sie die folgenden Schritte aus:

1. Konfigurieren Sie den TP-Monitor, sodass er auf den DB2 XA-Schalter zugreifen kann. Der DB2 XA-Schalter stellt dem TP-Monitor die Adressen der XA-APIs von DB2 Connect zur Verfügung. Jeder TP-Monitor führt dies auf eine andere Weise durch.
2. Konfigurieren Sie den TP-Monitor mit der DB2-Zeichenfolge XA_OPEN. Jeder TP-Monitor führt dies auf eine andere Weise durch. Die Dokumentation Ihres TP-Monitors enthält Informationen dazu, wie die DB2-Zeichenfolge XA_OPEN zur Verwendung durch den TP-Monitor konfiguriert wird.
3. Ändern Sie gegebenenfalls die Standardkonfigurationsparameter des Synchronisationspunktmanagers (SPM) von DB2 Connect. Host- und System i-Datenbankserver mit Version 5 Release 3 und früher unterstützen die XA-Schnittstelle noch nicht. System i verfügt ab Version 5 Release 4 über eine vollständige XA-Unterstützung.

Der SPM ist eine DB2 Connect-Komponente, die das XA-Protokoll des zweiphasigen Commits dem entsprechenden Protokoll des zweiphasigen Commits zuordnet, das von Host- und System i-Datenbankservern verwendet wird. Standardmäßig verfügt die DB2-Instanz über vordefinierte Werte für die SPM-Konfigurationsparameter. Der wichtigste Parameter des Datenbankmanagers ist der Konfigurationsparameter SPM_NAME. Der Standardwert ist eine Variante der ersten sieben Zeichen des TCP/IP-Hostnamens.

Wenn Sie eine Verbindung zu DB2 für OS/390 und z/OS über TCP/IP herstellen, sollte eine Änderung der Standardeinstellungen nicht erforderlich sein. In diesem Fall ist keine SPM-Konfiguration erforderlich, da der SPM bereits funktionsfähig ist.

DB2 Connect-Unterstützung für lose gekoppelte Transaktionen

Die in DB2 Connect integrierte Unterstützung für lose gekoppelte Transaktionen ist für Benutzer gedacht, die verteilte XA-Anwendungen implementieren, die auf DB2 Universal Database (UDB) für i5/OS Version 5 Release 4 oder höher oder DB2 UDB für OS/390 und z/OS Version 7 oder höher zugreifen. Diese Unterstützung ermöglicht es verschiedenen Zweigen derselben globalen Transaktion, einen Sperrbereich unter DB2 UDB für OS/390 und z/OS gemeinsam zu nutzen.

Unterstützung für lose gekoppelt Transaktionen ist für .NET- und COM+-Anwendungen vorgesehen.

Diese Funktion reduziert das Zeitfenster, in dem ein Zweig einer verteilten Transaktion aufgrund eines anderen Zweigs innerhalb derselben globalen Transaktion auf Situationen wie ein Zeitlimit für Sperren oder einen Deadlock trifft.

Kapitel 8. Versetzen von Daten mit DB2 Connect

Wenn Sie in einer komplexen Umgebung arbeiten, in der Sie Daten zwischen einem Hostdatenbanksystem und einer Workstation versetzen müssen, können Sie DB2 Connect, das Gateway für die Datenübertragung zwischen Host und Workstation, verwenden (siehe Abb. 8).

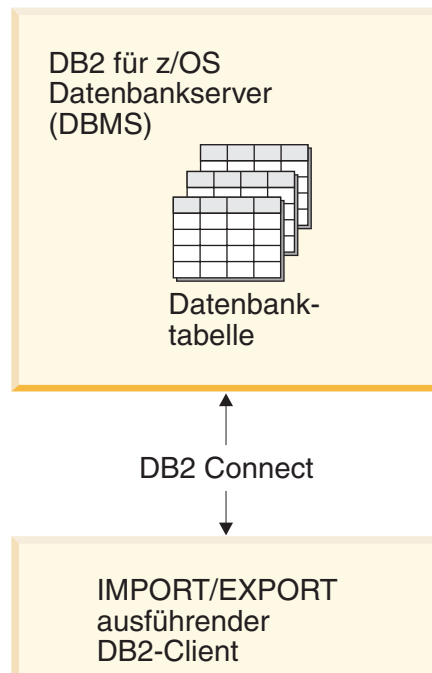


Abbildung 8. Import/Export über DB2 Connect

Mit den DB2-Dienstprogrammen IMPORT und EXPORT können Daten aus einer Datenbank auf einem Host- oder System i-Server in eine Datei auf der DB2 Connect-Workstation und umgekehrt versetzt werden. Anschließend können diese Daten mit jeder anderen Anwendung oder jedem anderen Verwaltungssystem für relationale Datenbanken verwendet werden, von der/dem das betreffende Import-/Exportformat unterstützt wird. Beispielsweise können Sie Daten aus einer Datenbank auf einem Host- oder System i-Server in eine PC/IXF-Datei exportieren und diese Daten anschließend in eine DB2 für Windows-Datenbank importieren.

Sie können Export- und Importfunktionen von einem Datenbankclient oder von der DB2 Connect-Workstation aus ausführen.

Anmerkung:

1. Die zu importierenden oder exportierenden Daten dürfen die Beschränkungen der beiden Datenbanken hinsichtlich der Größe und Datentypen nicht verletzen.
2. Zur Verbesserung der Leistung beim Importieren können zusammengesetzte Abfragen verwendet werden. Geben Sie beim Dienstprogramm IMPORT den Änderungswert `compound` für den Dateityp an, um eine angegebene Anzahl von

Abfrageanweisungen zu einem Block zusammenzufassen. Hierdurch können der Systemaufwand für das Netzwerk verringert und die Antwortzeiten verbessert werden.

Bei DB2 Connect müssen Export- und Importoperationen die folgenden Bedingungen erfüllen:

- Der Dateityp muss PC/IXF sein.
- Es muss eine Zieltabelle auf dem Zielsever erstellt werden, deren Attribute mit denen der Daten kompatibel sind, bevor der Importvorgang gestartet werden kann. Mit dem Dienstprogramm db2look können Sie die Attribute der Quellentabelle abrufen. Beim Import durch DB2 Connect kann keine Tabelle erstellt werden, weil INSERT die einzige unterstützte Option ist.

Wenn eine dieser Bedingungen nicht erfüllt ist, schlägt die Operation fehl, und eine Fehlermeldung wird zurückgegeben.

Anmerkung: Indexdefinitionen werden weder beim Export gespeichert noch beim Import verwendet.

Beim Importieren und Exportieren gemischter Daten (d. h. Spalten, die sowohl SBCS- als auch DBCS-Daten enthalten) muss Folgendes beachtet werden:

- Auf Systemen, die Daten im EBCDIC-Format speichern (MVS, OS/390, OS/400, VM und VSE), markieren DBCS-Startzeichen und DBCS-Endezeichen den Anfang und das Ende der DBCS-Daten. Bei der Definition von Spaltenlängen für die Datenbanktabellen muss daher genügend Platz für diese Zeichen gelassen werden.
- Wenn die Daten in einer Spalte kein einheitliches Muster haben, werden Spalten für Zeichendaten variabler Länge empfohlen.

Versetzen von Daten von einer Workstation auf einen Host-Server

Gehen Sie wie folgt vor, um Daten in eine Datenbank auf einem Host- oder System i-Server zu versetzen:

1. Exportieren Sie die Daten aus einer DB2-Tabelle in eine PC/IXF-Datei.
2. Importieren Sie die PC/IXF-Datei mit der Option INSERT in eine kompatible Tabelle der Datenbank auf dem Host-Server.

Gehen Sie wie folgt vor, um Daten aus einer Datenbank auf dem Host-Server auf eine Workstation zu versetzen:

1. Exportieren Sie die Daten aus der Tabelle der Datenbank auf dem Host-Server in eine PC/IXF-Datei.
2. Importieren Sie die PC/IXF-Datei in eine DB2-Tabelle.

Beispiel

Das folgende Beispiel veranschaulicht, wie Sie Daten von einer Workstation in eine Datenbank auf einem Host- oder System i-Server versetzen.

Exportieren Sie die Daten in ein externes IXF-Format, indem Sie den folgenden Befehl absetzen:

```
db2 export to staff.ixf of ixf select * from userid.staff
```

Setzen Sie den folgenden Befehl ab, um eine DRDA-Verbindung zur DB2-Ziel-datenbank herzustellen:

```
db2 connect to cbc664 user admin using xxx
```

Erstellen Sie die Zieltabelle auf der DB2-Ziel-datenbankinstanz, wenn sie dort noch nicht vorhanden ist:

```
CREATE TABLE mydb.staff (ID SMALLINT NOT NULL, NAME VARCHAR(9),  
    DEPT SMALLINT, JOB CHAR(5), YEARS SMALLINT, SALARY DECIMAL(7,2),  
    COMM DECIMAL(7,2))
```

Setzen Sie den folgenden Befehl ab, um die Daten zu importieren:

```
db2 import from staff.ixf of ixf insert into mydb.staff
```

Jede Datenzeile wird aus der Datei im IXF-Format gelesen, und eine SQL-Anweisung INSERT wird abgesetzt, um die betreffende Zeile in die Tabelle 'mydb.staff' einzufügen. Es werden so lange einzelne Zeilen eingefügt, bis alle Daten in die Zieltabelle versetzt wurden.

Ausführliche Informationen enthält das Handbuch "Moving Data Across the DB2 Family", eine Veröffentlichung aus der Reihe der IBM Redbooks. Diese Redbooks-Veröffentlichung steht unter der URL <http://www.redbooks.ibm.com/redbooks/SG246905> zur Verfügung.

Kapitel 9. SQLCODE-Zuordnung

Verschiedene IBM Produkte für relationale Datenbanken erzeugen nicht immer dieselben SQLCODE-Werte für ähnliche Fehler. Auch wenn der SQLCODE derselbe ist, kann er von Token begleitet sein, die auf unterschiedliche Weise angegeben werden. Die Token-Liste wird im Feld SQLERRMC des SQL-Kommunikationsbereichs (SQLCA) übergeben. Standardmäßig ordnet DB2 Connect die SQLCODE-Werte und Token aller Host- oder System i-Datenbankserver den entsprechenden SQLCODE-Werten von DB2 zu.

Wenn Sie die SQLCODE-Zuordnung inaktivieren wollen, müssen Sie **NOMAP** in der Parameterzeichenfolge des DCS-Verzeichnisses angeben.

Wenn Sie eine Anwendung direkt von einem Host- oder System i-Datenbankserver (beispielsweise DB2 Universal Database für OS/390 und z/OS) portieren, ist es unter Umständen sinnvoll, die SQLCODE-Zuordnung auszuschalten. Dies ermöglicht den Einsatz der Anwendung ohne Änderung der SQLCODE-Werte, auf die sie verweist.

Inaktivieren der SQLCODE-Zuordnung

Wenn Sie die SQLCODE-Zuordnung inaktivieren wollen, müssen Sie **NOMAP** in der Parameterzeichenfolge des DCS-Verzeichnisses angeben.

Wenn Sie eine Anwendung direkt von einem Host- oder System i-Datenbankserver (beispielsweise DB2 Universal Database (UDB) für OS/390 und z/OS) portieren, ist es unter Umständen sinnvoll, die SQLCODE-Zuordnung auszuschalten. Dies ermöglicht den Einsatz der Anwendung ohne Änderung der SQLCODE-Werte, auf die sie verweist.

Anpassung der SQLCODE-Zuordnung

Standardmäßig ordnet DB2 Connect die SQLCODE-Werte und Token aller Host- oder System i-Datenbankserver den entsprechenden SQLCODE-Werten von DB2 zu. Die folgenden Dateien sind genaue Kopien der standardmäßigen SQLCODE-Zuordnung:

- dcs1dsn.map ordnet SQLCODE-Werte von DB2 Universal Database (UDB) für OS/390 und z/OS und von DB2 für z/OS zu.
- dcs1ari.map ordnet SQLCODE-Werte von DB2 UDB für VSE und VM zu.
- dcs1qsq.map ordnet SQLCODE-Werte von DB2 UDB für iSeries und von DB2 für i5/OS zu.

Für DB2 unter Linux- oder UNIX-Betriebssystemen ist keine Zuordnung erforderlich.

1. Wenn Sie die SQLCODE-Standardzuordnung überschreiben wollen oder wenn Sie einen Host- bzw. System i-Datenbankserver verwenden, der über keine SQLCODE-Zuordnung verfügt (einen Datenbankserver, der nicht von IBM stammt), können Sie eine dieser Dateien kopieren und als Grundlage für Ihre neue SQLCODE-Zuordnungsdatei verwenden. Wenn die Datei nicht direkt edi-

tiert, sondern kopiert wird, ist sichergestellt, dass die ursprüngliche SQLCODE-Zuordnung später immer noch zur Verfügung steht.

2. Der Name der neuen SQLCODE-Zuordnungsdatei wird in der Parameterzeichenfolge des DCS-Verzeichnisses angegeben.
3. Zuordnungsdateien sind ASCII-Dateien, die mit einem ASCII-Editor erstellt und editiert werden können. Bei der Erstinstallation wird die Zuordnungsdatei im Verzeichnis `map` im Installationspfad gespeichert.

Die Datei kann folgende besondere Zeilenarten enthalten:

&& Der logische Dateibeginn. Alle Zeilen vor dem ersten Auftreten von **&&** werden als unformatierte Kommentare angesehen und ignoriert. Wenn in der Datei nach **&&** keine Angaben enthalten sind, wird keine SQLCODE-Zuordnung durchgeführt. Die SQLCODE-Zuordnung kann auch mit dem Parameter `NOMAP` ausgeschaltet werden (siehe oben).

***** Ein Stern als erstes Zeichen in einer Zeile kennzeichnet einen Kommentar.

W Ein **W** als einziges Zeichen in einer Zeile zeigt an, dass Warnungsmarkierungen neu zugeordnet werden sollten. Standardmäßig werden die ursprünglichen Warnungsmarkierungen übergeben. Das **W** muss in Großschreibung angegeben werden.

Alle anderen Zeilen nach **&&** müssen entweder leer sein oder Zuordnungsanweisungen im folgenden Format enthalten:

```
eingabecode [, ausgabecode [, tokenliste]]
```

eingabecode steht für eine der folgenden Angaben:

sqlcode Der SQLCODE-Wert vom Host- oder System i-Datenbankserver.

U Alle nicht definierten negativen SQLCODE-Werte (die nicht in dieser Datei aufgelisteten Codes) werden dem angegebenen *ausgabecode* zugeordnet. Wenn kein *ausgabecode* in dieser Zeile angegeben ist, wird der ursprüngliche SQLCODE verwendet. Das Zeichen muss in Großschreibung angegeben werden.

P Alle nicht definierten positiven SQLCODE-Werte (die nicht in dieser Datei aufgelisteten Codes) werden dem angegebenen *ausgabecode* zugeordnet. Wenn kein *ausgabecode* in dieser Zeile angegeben ist, wird der ursprüngliche SQLCODE verwendet. Das Zeichen muss in Großschreibung angegeben werden.

ccnn Der SQLSTATE-Klassencode vom Host- oder System i-Datenbankserver. *nn* ist einer der folgenden Werte:

- | | |
|-----------|---|
| 00 | Nicht weiter qualifizierte, erfolgreiche Beendigung |
| 01 | Warnung |
| 02 | Keine Daten |
| 21 | Ungültige Beziehungsart |
| 22 | Datenfehler |
| 23 | Ungültige Integritätsbedingung |
| 24 | Ungültiger Cursor-Status |
| 26 | Ungültige ID für eine SQL-Anweisung |
| 40 | Rollback der Transaktion |
| 42 | Ungültiger Zugriff |

- 51 Ungültiger Anwendungsstatus
- 55 Objekt nicht in erforderlichem Status
- 56 Sonstiger SQL- oder Produktfehler
- 57 Ressource nicht verfügbar oder Bedienereingriff
- 58 Systemfehler

Der angegebene *ausgabecode* wird für alle SQLCODE-Werte mit dem betreffenden Klassencode verwendet, die nicht ausdrücklich in der Zuordnungsdatei angegeben sind. Wenn kein *ausgabecode* in dieser Zeile angegeben ist, wird der ursprüngliche SQLCODE sich selbst zugeordnet, wobei keine Token kopiert werden.

Die Zeichen *cc* müssen in Kleinschreibung angegeben werden.

Wenn der gleiche *eingabecode* mehrmals in der Zuordnungsdatei aufgeführt ist, wird das erste Auftreten verwendet. *ausgabecode* gibt den ausgegebenen SQLCODE-Wert an. Wenn kein Wert angegeben wird, wird der ursprüngliche SQLCODE verwendet.

Wenn ein *Ausgabecode* angegeben wird, kann auch eine der folgenden Angaben gemacht werden:

- (s) Der eingegebene SQLCODE sowie die Produkt-ID (ARI, DSN oder QSQ) werden in das Nachrichten-Token-Feld des SQL-Kommunikationsbereichs geschrieben.

Der ursprüngliche SQLCODE wird als einziges Token zurückgegeben. Diese Option dient zum Verarbeiten nicht definierter SQLCODE-Werte mit Ausnahme von +965 und -969. Wenn der *ausgabecode* +965 oder -969 ist, umfasst die im Feld SQLERRMC des SQL-Kommunikationsbereichs zurückgegebene Token-Liste den ursprünglichen SQLCODE, gefolgt von der Produkt-ID, gefolgt von der ursprünglichen Token-Liste.

Das Zeichen *s* muss in Kleinschreibung angegeben werden.

(*tokenliste*)

Eine Liste durch Kommas getrennter Token. Wenn ein bestimmtes Token übersprungen werden soll, darf nur ein Komma angegeben werden. Die Angabe (*t2,,t4*) bedeutet z. B., dass das erste und dritte Ausgabe-Token leer sind.

Jedes Token besteht aus einer Zahl (*n*), der wahlweise *c* vorangeht und auf die wahlweise *c* oder *i* folgt, und wird wie folgt interpretiert:

- c* Der Datentyp des Tokens an dieser Stelle ist CHAR (Standardwert). Wenn *c* vor *n* steht, verweist es auf das Eingabe-Token, wenn es nach *n* steht, verweist es auf das Ausgabe-Token. Das Zeichen *c* muss in Kleinschreibung angegeben werden.
- i* Der Datentyp des Tokens an dieser Stelle ist INTEGER. Wenn *i* nach *n* steht, verweist es auf das Ausgabe-Token. *i* sollte nicht vor *n* stehen, da Datenbankserverprodukte auf dem IBM Host oder System *i* lediglich Token mit dem Datentyp CHAR unterstützen. Das Zeichen *i* muss in Kleinschreibung angegeben werden.
- n* Zahlen (eine oder mehrere), die angeben, welche Datenbankservertoken auf dem Host bzw. dem System *i* verwendet werden. Sie werden in der Reihenfolge angeordnet, die für die Platzierung im SQL-Kommunikationsbereich für die Ausgabe

gewünscht wird. Die Zahl gibt die Datenbankservertoken auf dem Host oder System i an; die Anordnung gibt die Reihenfolge an, in der die Token in den SQL-Kommunikationsbereich gestellt werden.

Beispiel: Der Host- oder System i-Datenbankserver gibt unter Umständen zwei Token zurück: 1 und 2. Soll Token 2 im SQL-Kommunikationsbereich für die Ausgabe vor Token 1 erscheinen, geben Sie (2,1) an.

Mehrere Token-Nummern können zu einem Ausgabe-Token mit dem Datentyp CHAR kombiniert werden, indem sie durch Punkte miteinander verbunden werden.

Zur Trennung von Ausgabe-Token werden Kommas verwendet. Wenn kein Token vor einem Komma angegeben ist, enthält der SQL-Kommunikationsbereich für die betreffende Stelle kein Ausgabe-Token. Alle Token, die im SQL-Kommunikationsbereich für die Ausgabe nach dem letzten angegebenen Token erscheinen, werden einem leeren Token zugeordnet.

Abb. 9 zeigt ein Beispiel einer SQLCODE-Zuordnungsdatei

&&				
	-007	,	-007	, (1)
	-010			
	-060	,	-171	, (2)
...				
	-204	,	-204	, (c1.2c)
...				
	-633	,	-206	, (,c1i)
	-30021	,	-30021	, (c1c,c2c)
	cc00	,	+000	
...				
	U	,	-969	, (s)
	P	,	+965	, (s)

Abbildung 9. Eine SQLCODE-Zuordnungsdatei

Die folgenden Beschreibungen beziehen sich auf die entsprechenden Zeilennummern in der vorangehenden Abbildung:

1. Der SQLCODE wird von -007 bis -007 zugeordnet. Das erste vom Host- oder System i-Datenbankserver empfangene Eingabe-Token wird als erstes Ausgabe-Token verwendet und hat standardmäßig den Datentyp CHAR. Es werden keine anderen Token übertragen.
2. Der SQLCODE wird von -010 bis -010 zugeordnet (es wird kein Ausgabe-SQLCODE angegeben). Es werden keine Token im SQL-Kommunikationsbereich für die Ausgabe gespeichert.
3. Der SQLCODE wird von -060 bis -171 zugeordnet. Das erste vom Host- oder System i-Datenbankserver empfangene Eingabe-Token wird gelöscht. Das zweite Token wird im SQL-Kommunikationsbereich für die Ausgabe als erstes Token verwendet und hat den Datentyp CHAR. Es gibt kein zweites Token im SQL-Kommunikationsbereich für die Ausgabe.

4. Der SQLCODE wird von -204 bis -204 zugeordnet. Das erste und zweite vom Host- oder System i-Datenbankserver empfangene Token haben den Datentyp CHAR. Diese beiden Eingabe-Token werden zu einem Ausgabe-Token mit dem Datentyp CHAR kombiniert, das zum ersten Ausgabe-Token im SQL-Kommunikationsbereich wird.
5. Der SQLCODE wird von -633 bis -206 zugeordnet. Das erste vom Host- oder System i-Datenbankserver empfangene Eingabe-Token hat den Datentyp CHAR. Es wird in den Typ INTEGER umgesetzt und als zweites Token im SQL-Kommunikationsbereich für die Ausgabe verwendet. Das erste Token im SQL-Kommunikationsbereich für die Ausgabe ist ein leeres Token, was durch ein Komma angezeigt wird.
6. Der SQLCODE wird von -30021 bis -30021 zugeordnet. Das erste und zweite vom Host- oder System i-Datenbankserver empfangene Eingabe-Token haben den Datentyp CHAR und werden als erstes und zweites Token im SQL-Kommunikationsbereich für die Ausgabe verwendet.
7. Alle SQLCODE-Werte in SQL-Kommunikationsbereichen mit SQLSTATE-Werten in der Klasse 00 werden SQLCODE +000 zugeordnet.
8. Für alle nicht definierten SQLCODE-Werte wird -969 zugeordnet. Diese Option sollte nur verwendet werden, wenn alle Codes, die zugeordnet werden können, einschließlich derer, die identisch sind und keine Zuordnung benötigen, aufgelistet sind. Die Option (**s**) gibt an, dass die Token-Liste, die im Feld SQLERRMC des SQL-Kommunikationsbereichs zurückgegeben werden soll, den ursprünglichen SQLCODE enthält, gefolgt von dem Produkt, in dem der Fehler auftrat, gefolgt von der ursprünglichen Token-Liste. Wenn der Eintrag **U** nicht vorgenommen wurde, werden alle nicht aufgeführten Codes ohne Zuordnung übergeben.
9. Für alle nicht definierten positiven SQLCODE-Werte wird +965 zugeordnet. Diese Option sollte nur verwendet werden, wenn alle Codes, die zugeordnet werden können, einschließlich derer, die identisch sind und keine Zuordnung benötigen, aufgelistet sind. Die Option (**s**) gibt an, dass die Token-Liste, die im Feld SQLERRMC des SQL-Kommunikationsbereichs zurückgegeben werden soll, den ursprünglichen SQLCODE enthält, gefolgt von dem Produkt, in dem die Warnung auftrat, gefolgt von der ursprünglichen Token-Liste. Wenn der Eintrag **P** nicht vorgenommen wurde, werden alle nicht aufgeführten positiven Codes ohne Zuordnung übergeben.

Kapitel 10. Datenbanksystemüberwachung und DB2 Connect

Hier finden Sie Erläuterungen zu einer Reihe von Möglichkeiten für die Verbindungs- und Leistungsüberwachung in einer Umgebung, in der DB2 Connect verwendet wird. Die Art der Überwachung ist für das jeweilige Betriebssystem spezifisch.

Überwachung von Verbindungen für ferne Clients

Sie können den Datenbanksystemmonitor mit einem DB2 Connect-Serverprodukt (zum Beispiel DB2 Connect Enterprise Edition) verwenden, um Verbindungen zu fernen Clients zu überwachen. Um Clients zu überwachen, die für den DB2 Connect-Server als lokal definiert sind, d. h. die auf dem Server selbst ausgeführt werden, muss die folgende Variable definiert werden:

```
db2set DB2CONNECT_IN_APP_PROCESS=NO
```

Wenn es zum Beispiel zu einem Fehler auf dem Host- oder System i-System kommt, kann der Systemadministrator feststellen, ob der Fehler auf der DB2 Connect-Workstation auftrat. Der Datenbanksystemmonitor ordnet die folgenden Angaben zu:

- Das DRDA-Korrelations-Token (CRRTKN) (beim ungeschützten Datenaustausch)
- Die ID der Arbeitseinheit (UOWID) (bei zweiphasigen Verbindungen, die durch den DRDA-3-Synchronisationspunktmanager (wie bei Verwendung über TCP/IP-Verbindungen) geschützt sind)
- Die Verbindungs-ID von DB2 Connect (die Anwendungs-ID)

Diese Informationen zeigen, welche Verbindung unter DB2 Connect das Problem verursachte, sodass der Systemadministrator die einzelne Client-Anwendung zwangsweise vom System abmelden kann, ohne die anderen Clients zu beeinträchtigen, die dieselbe Verbindung unter DB2 Connect verwenden.

Auflisten des Status von Monitorschaltern

Mit dem Befehl `db2 get monitor switches` kann der Status der Monitorschalter aufgelistet werden.

Überwachung der Leistung mit Windows Performance Monitor

Windows-Betriebssysteme stellen ein nützliches Tool zur Überwachung der Leistung Ihrer DB2-Anwendungen zur Verfügung. Performance Monitor, eines der Verwaltungs-Tools von Windows, zeigt eine grafische Darstellung der Systemleistung an. Sie können eine Reihe von Aspekten im Hinblick auf das System, die Datenbank und die Kommunikation auswählen, die überwacht und in einer grafischen Darstellung zusammengefasst werden sollen.

Die Berichte beispielsweise, die über die Befehle `GET SNAPSHOT FOR ALL DCS DATABASES` bzw. `GET SNAPSHOT FOR ALL DCS APPLICATIONS` zur Verfügung stehen, können mit dem Monitor in Echtzeit grafisch dargestellt und direkt mit Werten wie der CPU-Auslastung verglichen werden. Sie können die Auswirkungen verschiedener Einstellungen auf die Datenbank- oder Kommunikationsleistung direkt vergleichen. Sie haben die Möglichkeit, angepasste Konfigurationseinstellungen in PMC-Dateien zu speichern und später wieder abzurufen.

In der Abbildung unten beispielsweise werden mehrere DB2-Maßnahmen im Hinblick auf die CPU-Auslastung grafisch dargestellt. Die Zusammenstellung der Werte in der Darstellung wurde in der Datei db2chart.pmc gespeichert. Sie können so viele PMC-Dateien speichern, wie Sie möchten, um verschiedene Querschnitte der Systemleistung zu erfassen.

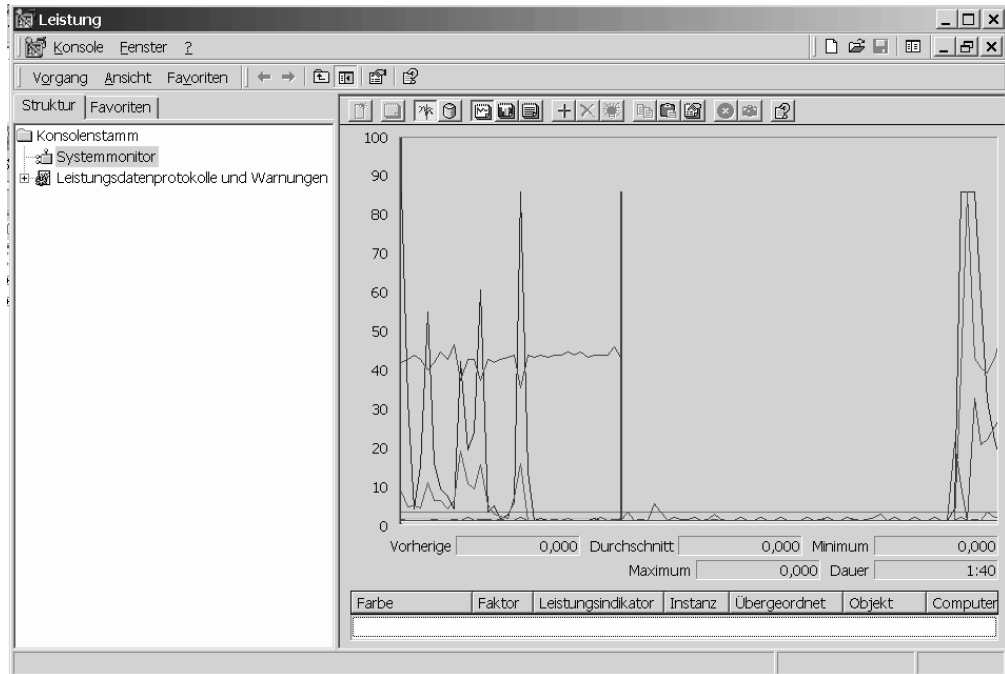


Abbildung 10. Performance Monitor

Um die Überwachung lokaler Anwendungen zu aktivieren, müssen Sie die Umgebungsvariable DB2CONNECT_IN_APP_PROCESS inaktivieren.

Verwendung der GET SNAPSHOT-Befehle

Der DB2-Monitor verwaltet eine aktive Aufstellung wertvoller Systeminformationen. Durch Ausgabe des Befehls GET SNAPSHOT können Sie jederzeit eine Zusammenfassung des Systemstatus abrufen. Sie können Momentaufnahmen des Monitors erstellen, wenn Sie für die Instanz des Datenbankmanagers, die Sie überwachen wollen, über die Berechtigung SYSMAINT, SYSCTRL oder SYSADM verfügen.

Fünf Befehle für Momentaufnahmen sind bei der Überwachung von DCS-Informationen von Nutzen:

- GET SNAPSHOT FOR ALL DCS DATABASES
- GET SNAPSHOT FOR ALL DCS APPLICATIONS
- GET SNAPSHOT FOR DCS APPLICATION ...
- GET SNAPSHOT FOR DCS DATABASE ON db_alias
- GET SNAPSHOT FOR DCS APPLICATIONS ON db_alias

Jeder Befehl für Momentaufnahmen führt zur Erstellung eines Berichts über den von Ihnen angeforderten Bereich.

Wenn Sie beispielsweise den Befehl GET SNAPSHOT FOR DCS DATABASE ON DCSDb, wird der folgende Bericht erstellt:

Momentaufnahme der DCS-Datenbank

```

DCS-Datenbankname           = DCSDB
Host-Datenbankname          = GILROY
Zeitmarke der ersten Datenbankverbindung = 12-15-2001 10:28:24.596495
Letzte Dauer des Verbindungsaufbaus     = 0,950561
Letzte Verbindungsdauer              = 0,000000
Host-Antwortzeit (sek,ms)            = 0,000000
Zeitmarke für letzte Zurücksetzung     =
Anzahl versuchter SQL-Anweisungen     = 2
Versuchte Commits                = 1
Versuchte Rollbacks              = 0
Fehlgeschlagene Anweisungsoperationen = 0
Gesamtanzahl der Gateway-Verbindungen = 1
Aktuelle Anzahl der Gateway-Verbindungen = 1
Auf Host-Antwort wartende Gateway-Verb. = 0
Auf Client-Anford. wartende Gateway-Verb. = 1
Gateway-Kommunikationsfehler zu Host   = 0
Zeitmarke letzter Kommunikationsfehler = Keine
Obere Grenze für Gateway-Verbindungen = 1
Ausgewählte Zeilen                = 0
Abgehende, gesendete Byte          = 140
Abgehende, empfangene Byte         = 103

```

Dieser Bericht liefert Informationen zu Datenbankverbindungen, zur Leistung, zu Fehlern und zum Durchsatz von SQL-Anforderungen. Die Momentaufnahmen von DB2 Monitor können weitaus detaillierter sein. Wenn Sie beispielsweise den Befehl GET SNAPSHOT FOR ALL DCS APPLICATIONS ausgeben, erhalten Sie einen ähnlichen Bericht wie den folgenden:

Momentaufnahme der DCS-Anwendung

```

Client-Anwendungs-ID        = 09150F74.B6A4.991215152824
Folgenummer                 = 0001
Berechtigungs-ID            = SMITH
Anwendungsname              = db2bp
Anwendungskennzeichen       = 1
Anwendungsstatus            = Warten auf Anforderung
Statusänderungszeit         = 12-15-2001 10:29:06.707086
Client-Knoten                = sys143
Client-Release-Stufe         = SQL06010
Client-Plattform             = AIX
Clientprotokoll              = TCP/IP
Client-Codepage              = 850
Prozess-ID der Client-Anwendung = 49074
Client-Anmelde-ID            = smith
Host-Anwendungs-ID          = G9150F74.B6A5.991215152825
Folgenummer                 = 0000
Aliasname für Datenbank auf Gateway = MVSDB
DCS-Datenbankname           = DCSDB
Host-Datenbankname          = GILROY
Host-Release-Stufe           = DSN05012
Host-ID für codierten Zeichensatz = 500

Adresse für abgehende Kommunikation = 9.21.21.92 5021
Protokoll für abgehende Kommunikation = TCP/IP
Adresse für eingehende Kommunikation = 9.21.15.116 46756
Zeitmarke der ersten Datenbankverbindung = 12-15-2001 10:28:24.596495
Host-Antwortzeit (sek,ms)            = 0,000000
Zeit für Gateway-Verarbeitung         = 0,000000
Zeitmarke für letzte Zurücksetzung     =
Ausgewählte Zeilen                = 0
Anzahl versuchter SQL-Anweisungen     = 2
Fehlgeschlagene Anweisungsoperationen = 0
COMMIT-Anweisungen              = 1
ROLLBACK-Anweisungen            = 0
Eingehende, empfangene Byte         = 404

```

```

Abgehende, gesendete Byte           = 140
Abgehende, empfangene Byte          = 103
Eingehende, gesendete Byte          = 287
Anzahl geöffneter Cursor            = 0
Leerlaufzeit der Anwendung          = 1 Minute und 32 Sekunden

Beendigungsstatus für UOW           =
Zeitmarke für Beendigung der vorigen UOW = 12-15-2001 10:28:25.592631
Zeitmarke für Start der UOW          = 12-15-2001 10:29:06.142790
Zeitmarke für Stopp der UOW         =
Antwortzeit letzte beendete UOW (sek,ms) = 0,034396

Vorherige Operation                  = Anweisung EXECUTE IMMEDIATE
Zeitmarke für Start d. vorherigen Operation= 12-15-2001 10:29:06.142790
Zeitmarke für Stopp d. vorherigen Operation= 12-15-2001 10:29:06.707053

Anweisung                            = Anweisung EXECUTE IMMEDIATE
Abschnittsnummer                      = 203
Anwendungsersteller                   = NULLID
Paketname                             = SQLC2C07
SQL-Compiler-Aufwandsschätzung in Timeron = 0
SQL-Compiler-Kardinalitätsschätzung    = 0
Zeitmarke für Start der Anweisung      = 12-15-2001 10:29:06.142790
Zeitmarke für Stopp der Anweisung      = 12-15-2001 10:29:06.707053
Host-Antwortzeit (sek,ms)             = 1,101612
Antw.zeit letzte beendete Anweis. (sek,ms) = 0,564263
Abgerufene Zeilen                     = 0
Zeit für Gateway-Verarbeitung          = 0,013367
Eingehende, empfangene Byte für Anweisung = 220
Abgehende, gesendete Byte für Anweisung = 130
Abgehende, empfangene Byte für Anweisung = 49
Eingehende, gesendete Byte für Anweisung = 27
Text der SQL-Anweisung:
create table t12 (col1 int, col2 char)

```

DCS-Anwendungsstatus

Der Systemmonitor stellt die folgenden drei Formate für den Befehl LIST DCS APPLICATIONS zur Verfügung:

- LIST DCS APPLICATIONS
- LIST DCS APPLICATIONS SHOW DETAIL
- LIST DCS APPLICATIONS EXTENDED

In der im Folgenden aufgeführten Ausgabe kann das Format der Hostanwendungs-ID und der Clientanwendungs-ID je nach Version der Host- oder System i-Datenbank und der Stufe der TCP/IP-Unterstützung unterschiedlich sein.

Tabelle 11. Format der Anwendungs-ID in Abhängigkeit von der Hostversion und der Stufe der TCP/IP-Unterstützung

Szenario	Format der Anwendungs-ID
Clients greifen auf Datenserver mit einer Unterstützungsstufe für RDB Manager kleiner als 7 zu	G91A0D3A.P8BC.060306212019

Tabelle 11. Format der Anwendungs-ID in Abhängigkeit von der Hostversion und der Stufe der TCP/IP-Unterstützung (Forts.)

Szenario	Format der Anwendungs-ID
Clients greifen über TCP/IP Version 4 auf Datenserver mit einer Unterstützungsstufe von 8 oder größer für RDB Manager zu	9.26.13.61.65289.060306213816
Clients greifen über TCP/IP Version 6 auf Datenserver mit einer Unterstützungsstufe von 8 oder größer für RDB Manager zu	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

LIST DCS APPLICATIONS

Die vom Monitor auf Anwendungsebene bereitgestellten Informationen können Sie mit dem Befehl DB2 LIST DCS APPLICATIONS anzeigen.

Es werden folgende Informationen zu einer TCP/IP-Verbindung (zwischen DB2 Connect und DB2 Universal Database für z/OS und OS/390) zurückgegeben:

Ber.ID	Anwendungsname	Anw. kenn.	Hostanwendungs-ID
NEWTON	db2cli.exe	7	G91A0D3A.P8BC.060306212019
NEWTON	db2cli.exe	25	9.26.13.61.65289.060306213816
NEWTON	db2cli.exe	20	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

Ber.ID Die Berechtigungs-ID, unter der die Anmeldung beim Host- oder System i-Datenbankserver erfolgt ist. Diese gibt an, wer die Anwendung ausführt.

Anwendungsname

Der Name der Anwendung, die auf dem Client ausgeführt wird, wie er DB2 Connect bekannt ist. Nur die ersten 20 Byte nach dem letzten Pfadtrennzeichen sind verfügbar.

Anwendungskennung

Der Agent, der auf der DB2 Connect-Workstation ausgeführt wird. Dieses Element kann verwendet werden, um Informationen des Datenbanksystemmonitors mit anderen Diagnoseinformationen zu verbinden. Die Agent-ID ist auch für den Befehl FORCE USERS oder die entsprechende Anwendungsprogrammierschnittstelle (API) erforderlich.

Hostanwendungs-ID

Dies sind:

- Das DRDA-Korrelations-Token (CRRTKN) (beim ungeschützten Datenaustausch)
- Die ID der Arbeitseinheit (UOWID) (bei zweiphasigen Verbindungen, die durch den DRDA-3-Synchronisationspunktmanager (wie bei Verwendung über TCP/IP-Verbindungen) geschützt sind)

Diese eindeutige Kennung wird generiert, wenn die Anwendung eine Verbindung zum Host- oder System i-Datenbankserver herstellt. Dieses Element kann in Verbindung mit der Anwendungs-ID verwendet werden, um die Client- und Serverteile der Anwendungsinformationen einander zuzuordnen.

LIST DCS APPLICATIONS SHOW DETAIL

Wenn das Befehlsformat DB2 LIST DCS APPLICATIONS SHOW DETAIL verwendet wird, werden weitere Informationen angezeigt, darunter:

Ber.ID	Anwendungsname	Anw. kenn.	Clientanwendungs-ID
NEWTON	db2cli.exe	37	2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224

Nr.	Client-DB-Alias	Client-knoten	Client-Release	Client-Codepage	Hostanwendungs-ID
00001	MDB	SAYYID	SQL09000	1252	G91A0D3A.P982.060306214231

Nr.	Hostdatenbankname	Host-Release
00001	MEXICO	DSN08015

ID der Clientanwendung

Identifiziert eindeutig die Anwendung, die mit der DB2 Connect-Workstation verbunden ist. Es gibt verschiedene Formate für die Anwendungs-ID, die vom Protokoll abhängig sind, das für die Übertragung zwischen dem Client und der DB2 Connect-Workstation verwendet wird.

Dieser Wert erlaubt das Korrelieren von Verbindungen von Clients zur DB2 Connect-Workstation und von dort zum Host- oder System i-Datenbankserver.

Clientfolgenummer (Nr.)

Die Folgenummer für den Client ist die Folgenummer der Transaktion. Sie wird als Hilfe bei der Zuordnung einer Transaktion verwendet, die auf verschiedene Systeme verteilt ist.

Client-DB-Alias

Der Aliasname der Datenbank, der von der Anwendung bereitgestellt wird, die eine Verbindung mit der Datenbank herstellt. Dieses Element kann verwendet werden, um die tatsächliche Datenbank zu identifizieren, auf die die Anwendung zugreift. Die Zuordnung zwischen diesem Namen und dem Datenbanknamen könnte durch Verwendung der Datenbankverzeichnisse auf dem Clientknoten und auf dem Serverknoten des Datenbankmanagers erfolgen.

Client-NNAME (Clientknoten)

Gibt den Knoten an, auf dem die Clientanwendung ausgeführt wird. Die Information hängt von dem verwendeten Clientprotokoll ab. Für einen über TCP/IP verbundenen Client ist dies der Hostname.

Clientprodukt-ID (Client-Release)

Das Produkt und die Version, die auf dem Client ausgeführt werden. Die Clientprodukt-IDs lauten:

- SQL07010 für Version 7.1 von DB2 Universal Database und DB2 Connect-Produkten und ihren Clients
- SQL08010 für Version 8.1 von DB2 Universal Database und DB2 Connect-Produkten und ihren Clients
- SQL08020 für Version 8.2 von DB2 Universal Database und DB2 Connect-Produkten und deren Clients.

- SQL09120 für Version 9.1 von DB2-Produkten, DB2 Connect-Produkten und deren Clients

Client-Codepage

Die Kennung für die Codepage auf dem Knoten, auf dem die überwachte Anwendung gestartet wurde.

Anhand dieser Informationen kann sichergestellt werden, dass die Datenumsetzung zwischen der Codepage der Anwendung und der Codepage der Datenbank (oder, bei Datenbanken von Host- oder System i-Datenbankservern, der ID für codierten Zeichensatz des Host- oder System i-Datenbankservers) unterstützt wird.

Wenn die Codepage der Anwendung nicht die Codepage ist, unter der der Datenbanksystemmonitor ausgeführt wird, kann dieses Codepage-Element das manuelle Umsetzen der Daten erleichtern, die von der Anwendung übergeben und vom Datenbanksystemmonitor angezeigt wurden. Z. B. kann es zum Umsetzen des Anwendungsnamens verwendet werden.

Folgenummer für abgehende Daten (Nr.)

Dies ist die Folgenummer der abgehenden Daten. Sie wird zum Korrelieren von Transaktionen auf verschiedenen Systemen verwendet.

Hostdatenbankname

Der wirkliche Name der Datenbank, mit der die Anwendung verbunden ist. Im DCS-Verzeichnis ist dies der *Zieldatenbankname*.

Host-Release

Das Produkt und die Version, die auf dem Server ausgeführt werden. Die Angabe hat das Format *PPPVVRRM*, wobei Folgendes gilt:

- PPP** Gibt das Produkt des Host- oder System i-Datenbankservers an (beispielsweise DSN für DB2 Universal Database für z/OS und OS/390, ARI für DB2 Server for VSE & VM oder QSQ für DB2 für i5/OS).
- VV** Stellt eine zweistellige Versionsnummer dar, z. B. 08.
- RR** Stellt eine zweistellige Releasenummer dar, z. B. 01.
- M** Stellt einen einstelligen Wert für die Modifikationsstufe dar (0 bis 9 oder A bis Z).

LIST DCS APPLICATIONS EXTENDED

Sie können den Befehl LIST DCS APPLICATIONS mit der Option EXTENDED verwenden, um einen erweiterten Bericht zu generieren. Im erweiterten Bericht werden alle Felder aufgelistet, die auch bei Angabe der Option SHOW DETAIL im Befehl aufgelistet werden, plus neun neue Felder:

- DCS-Anwendungsstatus
- Statusänderungszeit
- Clientplattform
- Clientprotokoll
- Host-ID für codierten Zeichensatz
- Clientanmelde-ID
- Prozess-ID der Clientanwendung
- Aliasname der Datenbank am Gateway
- DCS-Datenbankname

Durch die vorhandenen Befehlsoptionen werden die Felder horizontal aufgelistet (eine Zeile pro Anwendung), während sie durch die neue Option vertikal aufgelistet werden (ein Feld pro Zeile).

Die neue Syntax des Befehls sieht wie folgt aus:

```
LIST DCS APPLICATIONS [SHOW DETAIL | EXTENDED ]
```

Beispielausgabedaten dieses Befehls bei Verwendung der neuen Option EXTENDED:

Liste der DCS-Anwendungen - Erweiterter Bericht

```
Clientanwendung-ID           = 2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224
Folgenummer                   = 00001
Berechtigungs-ID              = NEWTON
Gesicherte Berechtigungs-ID   =
Anwendungsname                 = db2cli.exe
Anwendungskennung             = 37
Anwendungsstatus              = Warten auf Anforderung
Statusänderungszeit           = Nicht gesammelt
Clientknoten                   = SAYIID
Client-Release-Stufe          = SQL09000
Clientplattform                = NT
Clientprotokoll                = TCP/IP
Client-Codepage                = 1252
Prozess-ID der Clientanwendung = 1192
Clientanmelde-ID              = ISAYIID
Hostanwendungs-ID             = G91A0D3A.P982.060306214231
Folgenummer                    = 00001
Aliasname für Datenbank auf Gateway = MDB
DCS-Datenbankname             = MDB
Hostdatenbankname              = MEXICO
Host-Release-Stufe            = DSN08015
Host-ID für codierten Zeichensatz = 1208
```

Das Feld für den Anwendungsstatus enthält einen der drei folgenden Werte:

1. **CONNECT anstehend - abgehend:** Dies bedeutet, dass die Anforderung zum Verbinden zu einer Host- oder System i-Datenbank abgesetzt wurde und dass DB2 Connect auf die Verbindungsherstellung wartet.
2. **Warten auf Anforderung:** Dies bedeutet, dass die Verbindung zur Host- oder System i-Datenbank hergestellt wurde und dass DB2 Connect auf eine SQL-Anweisung von der Clientanwendung wartet.
3. **Warten auf Antwort.** Dies bedeutet, dass die SQL-Anweisung an die Host- oder System i-Datenbank gesendet wurde.

Zudem wird die Statusänderungszeit (State Change Time) nur dann im Bericht angezeigt, wenn der Schalter der UOW für den Systemmonitor während der Verarbeitung aktiviert wurde. Andernfalls wird "Nicht gesammelt" angezeigt.

Diagnosemonitor und Alerts

Der DB2 für z/OS-Diagnosemonitor wertet Richtlinien zur Objektverwaltung in regelmäßigen Abständen aus. Stellt der Diagnosemonitor fest, dass Objektverwaltungsmaßnahmen erforderlich sind, werden Diagnosealerts generiert. Aktionen als Antwort auf Diagnosealerts können angezeigt, zur Ausführung übergeben und gespeichert werden.

DB2 für z/OS-Diagnosemonitor - Übersicht

Auf z/OS-Systemen wird der DB2 für z/OS-Diagnosemonitor für jedes zu überwachende DB2-Subsystem oder auf einem dedizierten Mitglied einer Gruppe zur gemeinsamen Datennutzung (Data Sharing Group) als Task gestartet.

Der DB2 für z/OS-Diagnosemonitor löst die Auswertung von Objektverwaltungsrichtlinien zu geplanten Zeiten und in festgelegten Intervallen wie in der Richtlinie definiert aus. Die Objektverwaltungsrichtlinien werden mit Hilfe des Assistenten 'Objektverwaltungsrichtlinie erstellen' der DB2-Steuerzentrale erstellt. Bei jeder Richtlinienauswertung werden die Kriterien zur Empfehlung von Verwaltungsaktionen anhand der in der Objektverwaltungsrichtlinie festgelegten Schwellenwerte überprüft, um die Notwendigkeit von Objektverwaltungsaktionen zu bestimmen. Das heißt, es wird festgestellt, ob die Ausführung von COPY, REORG, RUNSTATS, STOSPACE, ALTER TABLESPACE oder ALTER INDEX erforderlich ist. Darüber hinaus werden Statusbedingungen des eingeschränkten Betriebs, wie zum Beispiel CHKP, für Tabellenbereichs-, Index- und Speichergruppenobjekte (falls zutreffend) ermittelt. Wenn bei der Richtlinienauswertung Objekte im Alertstatus ermittelt werden, werden die Ansprechpartner für den Diagnosealert der Richtlinie unter Ihren E-Mail-Adressen oder Pagernummern benachrichtigt. Die Liste der Ansprechpartner bei Diagnosealerts für die einzelnen DB2-Subsysteme wird in der Steuerzentrale definiert und verwaltet.

Nach dem Start des Diagnosemonitors wird zu Anfang eine Momentaufnahme des Auswertungszeitplans für die Richtlinien erfasst, mit deren Hilfe der Diagnosemonitor den Zeitpunkt bestimmt, zu dem Richtlinienauswertungen auszulösen sind. Diese Zeitplanmomentaufnahme wird zu dem Aktualisierungszeitpunkt aktualisiert, der beim Starten des Diagnosemonitors angegeben wurde, oder wenn der Diagnosemonitor einen Aktualisierungsbefehl (Refresh) empfängt. Alle Änderungen am Auswertungszeitplan einer Richtlinie werden vom Diagnosemonitor berücksichtigt, wenn eine Aktualisierung (Refresh) des Zeitplans erfolgt.

Der Diagnosemonitor wird über die Konsole mit Hilfe der MVS-Systembefehle START und STOP gestartet bzw. gestoppt.

Ein Beispiel für eine katalogisierte Prozedur (DSNHMONP), die einen DB2-Diagnosemonitor startet, und ein Beispiel für eine katalogisierte Prozedur (DSNHMONA), die mehrere DB2-Diagnosemonitore innerhalb eines MVS-Systems oder Parallel Sysplex startet, werden vom Installationsjob DSNTIJHM in einer Prozedurenbibliothek angelegt.

Sichten, Tabellen, Dateien, katalogisierte Prozeduren, gespeicherte Prozeduren, benutzerdefinierte Funktionen und die Ergebnismengentabelle, die vom DB2-Diagnosemonitor bzw. bei den zugehörigen Tasks (siehe unten) verwendet werden, werden durch die Installationsjobs DSNTIJCC und DSNTIJHM erstellt und installiert. Die Jobs DSNTIJCC und DSNTIJHM werden mit FMIDs JDB771D und JDB881D geliefert.

Protokoll für Richtlinienauswertungen

Richtlinienauswertungen, die vom DB2-Diagnosemonitor ausgelöst werden, werden in der Tabelle DSNACC.HM_EVAL_LOG protokolliert. Es wird ein Eintrag protokolliert, wenn eine Richtlinienauswertung beginnt und wenn eine Richtlinienauswertung endet. Die Protokolleinträge werden über sieben Tage aufbewahrt. Anschließend werden sie aus der Tabelle gelöscht. Die DB2-Sicht DSNACC.HM_ALERT_PO_EV, die durch den Installationsjob DSNTIJCC für diese Tabelle erstellt wurde, kann zum Anzeigen aller Richtlinien verwendet werden, deren letzte Auswertungsiteration nicht erfolgreich war.

Starten, Stoppen und Aktualisieren des DB2 für z/OS-Diagnosemonitors

Der DB2 für z/OS-Diagnosemonitor wird auf dem z/OS-System für jedes zu überwachende DB2-Subsystem bzw. auf jedem dedizierten Mitglied einer Gruppe zur gemeinsamen Datennutzung (Data Sharing Group) als Task gestartet.

- Zum Starten eines DB2-Diagnosemonitors führen Sie den folgenden MVS-Systembefehl START aus:

```
S teildateiname, DB2SSN=ssid, JOBNAME=HMONssid, TRACE=trace, REFRESH=nn
```

Die Parameter TRACE und REFRESH sind optional.

teildateiname

Gibt eine Teildatei einer Prozedurenbibliothek an, die zum Starten des DB2-Diagnosemonitors ausgeführt wird. In diesem Fall ist dies DSNHMONP. Diese katalogisierte Prozedur wird durch den Installationsjob DSNTIJHM erstellt.

ssid

Gibt den Namen bzw. die Kennung des zu überwachenden DB2-Subsystems an.

trace

Gibt die Tracemarkierung an. Folgende Werte sind gültig:

- ON - Trace aktivieren. Tracesätze werden in die Systemausgabe (SYSOUT) geschrieben.
- OFF - Trace nicht aktivieren.

Der Standardwert ist OFF.

nn

Gibt die Stunde (im 24-Stunden-Format) für den Zeitpunkt an, zu dem der Diagnosemonitor die Momentaufnahme für den Auswertungszeitplan aktualisiert, die er zum Auslösen von Richtlinienauswertungen verwendet. Der Standardwert ist 22.

- Zum Starten mehrerer DB2-Diagnosemonitore führen Sie den folgenden MVS-Systembefehl START aus:

```
S teildateiname
```

teildateiname

Eine Teildatei einer Prozedurenbibliothek, die zum Starten mehrerer DB2-Diagnosemonitore verwendet wird. In diesem Fall ist dies DSNHMONA.

Anmerkung: Vor dem Starten mehrerer DB2-Diagnosemonitore durch einen Befehl START mit DSNHMONA muss die in der Prozedur DSNHMONA angegebene Datei HMONPARM mit der Liste der zu überwachenden Subsysteme gefüllt werden. Die katalogisierte Prozedur und die Datei werden durch den Installationsjob DSNTIJHM erstellt.

- Zur Aktualisierung der Momentaufnahme des Richtlinienauswertungszeitplans, die vom DB2-Diagnosemonitor verwendet wird, um zu bestimmen, wann Richtlinienauswertungen auszulösen sind, führen Sie den folgenden MVS-Systembefehl MODIFY aus:

```
F HMONssid,APPL=REFRESH
```

ssid

Name oder Kennung des DB2-Subsystems, das von dem DB2-Diagnosemonitor, den Sie aktualisieren, überwacht wird.

- Zum Stoppen eines DB2-Diagnosemonitors führen Sie den folgenden MVS-Systembefehl STOP aus:

```
STOP HMONssid oder P HMONssid
```

ssid

Name oder Kennung des DB2-Subsystems, das von dem DB2-Diagnosemonitor, den Sie stoppen, überwacht wird.

Anzeigen, Übergeben und Speichern empfohlener Aktionen

Zum Anzeigen, Übergeben und Speichern der Aktionen, die bei der Richtlinienauswertung für Alertobjekte empfohlen werden, rufen Sie die gespeicherte DB2-Prozedur SYSPROC.DSNACCHR auf, die durch den Installationsjob DSNTIJCC erstellt wird. DSNACCHR ist eine gespeicherte Prozedur, die die empfohlenen Aktionen für Alertobjekte bestimmt, die bei der Richtlinienauswertung ermittelt werden, und einen JCL-Job generiert, der die empfohlenen Aktionen ausführt.

Das folgende Syntaxdiagramm zeigt die SQL-Anweisung CALL für den Aufruf von DSNACCHR. Da für DSNACCHR die Verbindungskonvention GENERAL WITH NULLS gilt, müssen Sie, wenn Sie Parameter in Hostvariablen übergeben, einen Nullanzeiger für jede Hostvariable einfügen. Nullanzeiger für Eingabehostvariablen müssen vor der Ausführung der CALL-Anweisung initialisiert werden.

Syntax

```

▶▶—CALL DSNACCHR (—abfragetyp,—diagnoseanzeiger,—richtlinien-id,—arbeitsgruppe,——————▶
▶—dateiname,—teildateiname,—speicheroption,—tracemarkierung,——————▶
  └─NULL—┘ └─NULL—┘ └─NULL—┘
▶—job-id,—jobname,—jcl-verarbeitungszeit,—tracemarkierung,—letzte_anweisung,——————▶
▶—rückkehrcode,—fehlernachricht )—————▶▶

```

abfragetyp

Gibt an, was mit den Aktionen, die für Objekte empfohlen werden, die bei der Richtlinienauswertung als im Alertstatus befindlich ermittelt wurden, geschehen soll. Folgende Werte sind gültig:

- 0 - Anzeigen empfohlener Aktionen zu Alertobjekten als JCL-Job
- 1 - Übergeben des JCL-Jobs, der die empfohlenen Aktionen für Alertobjekte ausführt
- 2 - Übergeben des JCL-Jobs, der die empfohlenen Aktionen für Alertobjekte ausführt, und Einreihen des Jobs in die Warteschlange
- 3 Speichern der empfohlenen Aktionen für Alertobjekte als JCL-Job in einer Bibliotheksteildatei

Der Parameter *abfragetyp* ist ein Eingabeparameter des Typs INTEGER.

diagnoseanzeiger

Gibt den Typ von Alert an, den DSNACCHR in den JCL-Job einschließt. Folgende Werte sind gültig:

- RS - Status des eingeschränkten Betriebs (Restricted State)
- EX - Speicherbereiche überschritten (Extents Exceeded)
- RR - Reorganisation erforderlich (REORG Required)
- CR - COPY erforderlich (COPY Required)
- RT - RUNSTATS erforderlich (RUNSTATS Required)
- SS - STOSPACE erforderlich (STOSPACE Required)

Der Parameter *diagnoseanzeiger* ist ein Eingabeparameter des Typs VARCHAR(4).

richtlinien-id

Gibt eine Objektverwaltungsrichtlinie an. Der Parameter *richtlinien-id* ist ein Eingabeparameter des Typs VARCHAR(7).

arbeitsgruppe

Gibt die Arbeitsgruppe einer Objektverwaltungsrichtlinie an, die die Alertobjekte ermittelt hat, die DSNACCHR in den JCL-Job einfügt. Diese Arbeitsgruppe muss mit der Richtlinie im Parameter *richtlinien-id* und dem Typ von Alert im Parameter *diagnoseanzeiger* zusammen angegeben werden. Der Parameter *arbeitsgruppe* ist ein Eingabeparameter des Typs INTEGER.

dateiname

Gibt den vollständig qualifizierten Namen einer partitionierten Datei (PDS) oder einer erweiterten partitionierten Datei (PDSE) an. Dieser Wert muss angegeben werden, wenn für *abfragetyp* der Wert 3 angegeben wird. Der Parameter *dateiname* ist ein Eingabeparameter des Typs VARCHAR(44).

teildateiname

Gibt eine Teildatei der partitionierten Datei (PDS) bzw. der erweiterten partitionierten Datei (PDSE) an, die im Parameter *dateiname* angegeben wird, in der der JCL-Objektverwaltungsjob gespeichert wird. Dieser Wert muss angegeben werden, wenn für *abfragetyp* der Wert 3 angegeben wird. Der Parameter *teildateiname* ist ein Eingabeparameter des Typs VARCHAR(8).

speicheroption

Gibt an, wie der JCL-Objektverwaltungsjob zu speichern ist. Dieser Wert muss angegeben werden, wenn für *abfragetyp* der Wert 3 angegeben wird. Folgende Werte sind gültig:

- R - Ersetzen
- A - Anhängen
- NM - Neue Teildatei

Der Parameter *speicheroption* ist ein Eingabeparameter des Typs VARCHAR(2).

tracemarkierung

Gibt an, ob die Tracefunktion aktiviert wird oder nicht. Folgende Werte sind gültig:

- Y - Trace aktivieren
- N - Trace inaktivieren

Der Eingabeparameter *tracemarkierung* hat den Typ CHAR(1).

job-id

Gibt die Job-ID des übergebenen Jobs an, wenn für *abfragetyp* der Wert 1 oder 2 angegeben wird. Der Parameter *job-id* ist ein Ausgabeparameter des Typs VARCHAR(8).

jobname

Gibt den Namen des übergebenen Jobs an, wenn für *abfragetyp* der Wert 1 oder 2 angegeben wird. Der Parameter *jobname* ist ein Ausgabeparameter des Typs VARCHAR(8).

jcl-verarbeitungszeit

Gibt die Zeit an, zu der die Anforderung verarbeitet wurde. Der Parameter *jcl-verarbeitungszeit* ist ein Ausgabeparameter des Typs TIMESTAMP.

letzte_anweisung

Wenn DSNACCHR einen schwer wiegenden Fehler (Rückkehrcode 12) zurückgibt, enthält dieses Feld die SQL-Anweisung, die ausgeführt wurde, als der Fehler auftrat. Der Parameter *letzte_anweisung* ist ein Ausgabeparameter des Typs VARCHAR(2500).

rückkehrcode

Der Rückkehrcode der Ausführung von DSNACCHR. Folgende Werte sind gültig:

- 0 - DSNACCHR wurde erfolgreich ausgeführt.
- 12 - DSNACCHR wurde mit schwer wiegenden Fehlern beendet. Der Parameter *fehlnachricht* enthält eine Nachricht, die den Fehler beschreibt. Der Parameter *letzte_anweisung* enthält die SQL-Anweisung, die ausgeführt wurde, als der Fehler auftrat.

Der Parameter *rückkehrcode* ist ein Ausgabeparameter des Typs INTEGER.

fehlnachricht

Wenn DSNACCHR einen schwer wiegenden Fehler (Rückkehrcode 12) zurückgibt, enthält dieses Feld Fehlnachrichten, einschließlich des formatierten SQL-Kommunikationsbereichs (SQLCA). Der Parameter *fehlnachricht* ist ein Ausgabeparameter des Typs VARCHAR(1331).

DSNACCHR gibt eine Ergebnismenge zurück, wenn der Parameter *abfragetyp* den Wert 0 hat. Die Ergebnismenge enthält den von DSNACCHR generierten JCL-Job. Die Tabelle für die DSNACCHR-Ergebnismenge wird durch den Installationsjob DSNTIJCC erstellt. Tabelle 12 auf Seite 88 zeigt das Format der Ergebnismenge.

Table 12. Format der DSNACCHR-Ergebnismenge

Spaltenname	Datentyp	Beschreibung
JCLSEQNO	INTEGER	Folgenummer der Tabellenzeile (1,...,n)
JCLSTMT	VARCHAR(80)	Gibt eine JCL-Anweisung an.

Anzeigen von Diagnosealertzusammenfassungen

Die Funktion HEALTH_OVERVIEW gibt Informationen aus der Diagnosealertzusammenfassungsdatei VSAM KSDS (Health Alert Summary) in Form einer DB2-Tabelle zurück. Diese Datei wird durch den Installationsjob DSNTIJHM erstellt.

Die Diagnosealertzusammenfassungsdatei enthält Informationen zum Status des DB2-Diagnosemonitors und zusammenfassende Alertstatistikdaten für jedes DB2-Subsystem, das vom Diagnosemonitor auf dem jeweiligen MVS-System bzw. Parallel Sysplex zuvor überwacht wurde bzw. gegenwärtig überwacht wird. Diese Informationen werden in Form einer Zeile für jedes DB2-Subsystem und jede Alertempfehlung an den Client zurückgegeben.

Das Ergebnis der Funktion ist eine DB2-Tabelle mit den folgenden Spalten:

ip-addr

Die IP-Adresse des DB2-Servers. Dies ist eine Spalte des Typs VARCHAR(40).

db2-ssid

Die Subsystem-ID des DB2-Subsystems. Dies ist eine Spalte des Typs VARCHAR(4).

health-ind

Der Typ von Alert. Folgende Werte sind gültig:

- RS - Status des eingeschränkten Betriebs (Restricted State)
- EX - Speicherbereiche überschritten (Extents Exceeded)
- RR - Reorganisation erforderlich (REORG Required)
- CR - COPY erforderlich (COPY Required)
- RT - RUNSTATS erforderlich (RUNSTATS Required)
- SS - STOSPACE erforderlich (STOSPACE Required)
- PO - Richtlinienbewertung fehlgeschlagen (Failed Policy Evaluation)
- HM - Diagnosemonitorstatus (Health Monitor State)

Die Spalte *health-ind* ist vom Typ VARCHAR(4).

host-name

Der vollständig qualifizierte Domänenname des DB2-Servers. Dies ist eine Spalte des Typs VARCHAR(255).

summary-stats

Der Status des DB2-Diagnosemonitors, wenn *health-ind* 'HM' ist. Folgende Werte sind gültig:

- 0 - Diagnosemonitor ist nicht gestartet.
- 1 - Diagnosemonitor ist gestartet.
- -1 - Status des Diagnosemonitors ist unbekannt.

Ansonsten die Gesamtzahl von Alertobjekten mit dem in *health-ind* angegebenen Alerttyp. Dies ist eine Spalte des Typs INTEGER.

alert-state

Der Status des in *health-ind* angegebenen Alerts. Folgende Werte sind gültig:

- 5 - Alarm
- 4 - Achtung
- 3 - Warnung
- 0 - Normal

Der Wert von *alert-state* ist immer 0, wenn *health-ind* 'HM' ist. Dies ist eine Spalte des Typs INTEGER.

Der externe Programmname für die Funktion lautet HEALTH_OVERVIEW. Der spezifische Name ist DSNACC.DSNACCHO. Die Funktion wird durch den Installationsjob DSNTIJCC erstellt.

Beispiel: Ermitteln der Gesamtzahl von Alertobjekten, die eine COPY-Operation für das DB2-Subsystem 'ABCD' erfordern:

```
SELECT SUMMARYSTATS FROM TABLE (DSNACC.HEALTH_OVERVIEW()) AS T
      WHERE DB2SSID = 'ABCD'
      AND HEALTHIND = 'CR';
```

Anzeigen von Diagnosealertobjekten

Die Alertobjekte, die bei der letzten erfolgreichen Iteration einer Richtlinienauswertung ermittelt wurden, werden nach Objekttyp in folgenden Repository-Tabellen für Alertobjekte gespeichert.

Folgende Alertobjekte sind verfügbar:

- DSNACC.HM_MAINT_TS für Tabellenbereiche
- DSNACC.HM_MAINT_IX für Indizes
- DSNACC.HM_MAINT_SG für Speichergruppen

DB2 erstellt eine Reihe von Sichten für diese Repository-Tabellen für Alertobjekte. Die Sichten und Repository-Tabellen für Alertobjekte werden durch den Installationsjob DSNTIJCC erstellt. Tabelle 13 enthält eine Liste der Sichten mit den Tabellen, für die die einzelnen Sichten definiert sind, sowie Beschreibungen der Sichten. Alle Sichtnamen und Tabellennamen haben das Qualifikationsmerkmal DSNACC.

Tabelle 13. Sichten für Diagnosealertobjekte

Name der Sicht	Für Tabelle	Beschreibung der Sicht
HM_ALERT_TS_RS	HM_MAINT_TS	Zeigt alle Tabellenbereiche im Status des eingeschränkten Betriebs an.

Tabelle 13. Sichten für Diagnosealertobjekte (Forts.)

Name der Sicht	Für Tabelle	Beschreibung der Sicht
HM_ALERT_TS_EX	HM_MAINT_TS	Zeigt alle Tabellenbereiche an, deren (EXTENTSIZE großen) Speicherbereiche einen benutzerdefinierten Grenzwert überschritten haben.
HM_ALERT_TS_RR	HM_MAINT_TS	Zeigt alle Tabellenbereiche an, die eine Reorganisation (REORG) erfordern.
HM_ALERT_TS_CR	HM_MAINT_TS	Zeigt alle Tabellenbereiche an, die die Ausführung von COPY erfordern.
HM_ALERT_TS_RT	HM_MAINT_TS	Zeigt alle Tabellenbereiche an, die die Ausführung von RUNSTATS erfordern.
HM_ALERT_IX_RS	HM_MAINT_IX	Zeigt alle Indizes im Status des eingeschränkten Betriebs an.
HM_ALERT_IX_EX	HM_MAINT_IX	Zeigt alle Indizes an, deren (EXTENTSIZE großen) Speicherbereiche einen benutzerdefinierten Grenzwert überschritten haben.
HM_ALERT_IX_RR	HM_MAINT_IX	Zeigt alle Indizes an, die eine Reorganisation (REORG) erfordern.
HM_ALERT_IX_CR	HM_MAINT_IX	Zeigt alle Indizes an, die die Ausführung von COPY erfordern.
HM_ALERT_IX_RT	HM_MAINT_IX	Zeigt alle Indizes an, die die Ausführung von RUNSTATS erfordern.
HM_ALERT_SG_SS	HM_MAINT_SG	Zeigt alle Speichergruppen an, die STOSPACE erfordern.

Teil 3. Hohe Verfügbarkeit und DB2 Connect

In einer Umgebung, in der DB2 Connect verwendet wird, müssen bestimmte Aspekte hinsichtlich der Hochverfügbarkeit berücksichtigt werden. Wenn ein Datenbankserver in einem Netz aus einem bestimmten Grund nicht mehr verfügbar ist, ist die Fähigkeit, eine Client-Workstation an einen alternativen Server im Netz umzuleiten, von großer Bedeutung.

Kapitel 11. Hohe Verfügbarkeit und Lastausgleich für die Konnektivität von Hostdatenbanken

Auf dem heutigen IT-Markt besteht eine hohe Nachfrage nach Datenverfügbarkeit rund um die Uhr. Diese Nachfrage muss befriedigt werden, damit ein Unternehmen im Wettbewerb bestehen und kontinuierliches Wachstum generieren kann. Viele der heutigen Web-, e-business- und Tabellenkalkulationsanwendungen setzen den Zugriff auf Unternehmensdaten voraus. Hierfür müssen zuverlässige, schnelle und sichere Verbindungen zu Host- und System i-Datenbanken hergestellt werden. Diese Verbindungen müssen rund um die Uhr verfügbar und in der Lage sein, die hohe Nachfrage nach Verbindungen auch bei starker Arbeitsbelastung befriedigen zu können. Hierbei stellt sich die Frage, wie sich solche Verbindungen aufbauen lassen.

Szenario für hohe Verfügbarkeit

Ein Unternehmen verfügt über mehrere Workstations und Anwendungsserver, die unter Windows, Linux und UNIX ausgeführt werden. Diese Maschinen benötigen Zugriff auf Daten, die sich auf mehreren Host- und System i-Datenbanken befinden. Anwendungen, die auf diesen Maschinen ausgeführt werden, brauchen schnelle und zuverlässige Verbindungen zu den Datenbanken. Das gesamte System ist durch ein Ethernet-Netzwerk über TCP/IP verbunden.

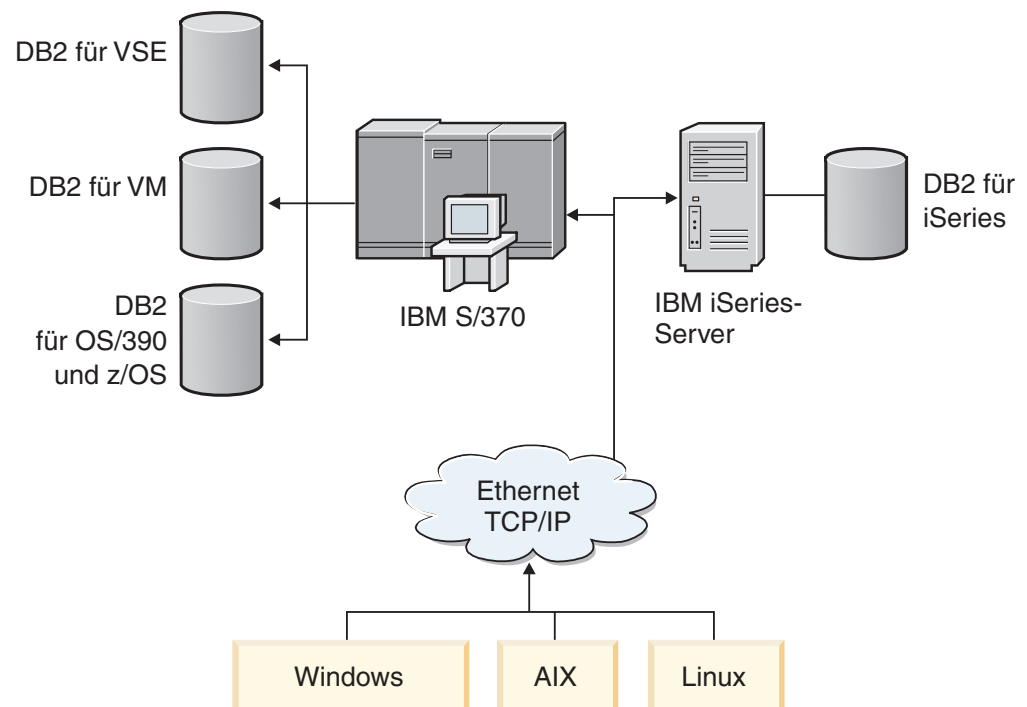


Abbildung 11. Beispielnetzwerkszenario

Damit die Workstations und Anwendungsserver auf Host- und System i-Datenbanken zugreifen können, benötigen Sie eine Konnektivitätskomponente als Vermittler. Diese Komponente muss jederzeit verfügbare, zuverlässige und schnelle Verbindungen zu den Host- und System i-Datenbanken zur Verfügung stellen. Außerdem muss sie skalierbar sein, um einem zukünftigen Anstieg des Verbindungsvolumens gewachsen zu sein.

Über die Links zu diesem Themen können Sie Informationen zu einer Lösung mit DB2 Connect und automatischer Clientweiterleitung aufrufen.

Kapitel 12. Automatische Clientweiterleitung - Beschreibung und Einrichtung (DB2 Connect)

Die Aufgabe der Funktion für automatische Clientweiterleitung besteht in erster Linie darin, einer Anwendung des IBM Data Server-Clients die Wiederstellung einer Verbindung nach einer Unterbrechung zu ermöglichen, damit die Anwendung den Betrieb schnellstmöglich fortsetzen kann. Die Weiterleitung ist für die Sicherstellung des Dauerbetriebs unerlässlich. Sie ist jedoch nur möglich, wenn eine alternative Position verfügbar ist, die der Clientverbindung bekannt ist. In einer Umgebung, bei der es sich nicht um eine DB2 Connect-HADR-Umgebung handelt, wird die Datenbank, auf die zugegriffen wird, normalerweise zwischen dem ursprünglichen DB2-Server und dem alternativen DB2-Server synchronisiert, wobei eine Reihe verschiedener Methoden verwendet wird, wie z. B. HACMP (High Availability Cluster Multiprocessor) oder HADR (High Availability Disaster Recovery).

Im Fall des DB2 Connect-Servers, bei dem keine Synchronisation der lokalen Datenbanken erforderlich ist, muss jedoch lediglich sichergestellt werden, dass sowohl auf dem ursprünglichen als auch auf dem alternativen DB2 Connect-Server die Zielhost- bzw. System i-Datenbank so katalogisiert ist, dass über einen identischen Datenbankaliasnamen darauf zugegriffen werden kann.

Anmerkung: In einer DB2 Connect-Serverumgebung kann ein alternativer DB2 Connect-Server angegeben werden, um die automatische Weiterleitung zwischen einem Client und dem DB2 Connect-Server zu ermöglichen. Damit eine Weiterleitung zwischen den DB2 Connect Personal Edition- oder DB2 Connect-Serverprodukten und einem Host oder System i-Datenbankserver stattfinden kann, muss der ferne Server eine oder mehrere alternative Adressen für sich selbst bereitstellen. Bei DB2 für z/OS sind mehrere Adressen bekannt, wenn es sich bei der Datenbank um eine Sysplex-Umgebung mit gemeinsamer Datennutzung handelt.

Die Weiterleitungsfunktion für Sysplex kann zwischen DB2 Connect und dem Hostdatenbankserver konfiguriert werden, falls die Sysplex-Unterstützung aktiviert ist. Die Weiterleitungsfunktion für Sysplex ist eine DB2 Connect-Funktionalität, die es DB2 Connect ermöglicht, die Verbindungsherstellung mit anderen Mitgliedern der Sysplex-Gruppe zu versuchen, wenn die Kommunikation mit dem ursprünglichen Mitglied unterbrochen wird. Zur Aktivierung der Weiterleitungsfunktion für Sysplex in DB2 Connect muss der alternative Server nicht unbedingt im Datenbankverzeichnis katalogisiert sein. Standardmäßig ist die Weiterleitungsfunktion für Sysplex aktiviert, wenn die Sysplex-Unterstützung aktiviert ist.

Damit ein IBM Data Server-Client nach einer Unterbrechung der Verbindung zu einem DB2 Connect-Server mithilfe der automatischen Clientweiterleitung wiederhergestellt werden kann, muss die Position eines alternativen DB2 Connect-Servers angegeben werden, bevor die Unterbrechung der Kommunikation auftritt. Der Befehl UPDATE ALTERNATE SERVER FOR DATABASE wird dazu verwendet, die Position des alternativen DB2 Connect-Servers für eine bestimmte Host- oder System i-Datenbank zu definieren. Der Hostname und die Portnummer des alternativen Servers werden mit dem Befehl angegeben. Die Position wird in der Datei des Systemdatenbankverzeichnisses auf dem DB2 Connect-Server gespeichert.

Um sicherzustellen, dass die Position des alternativen DB2 Connect-Servers für diese Datenbank für alle Clients gültig ist, muss die Position des alternativen Servers auf dem DB2 Connect-Server angegeben werden. Der alternative Server wird ignoriert, wenn er auf der Clientinstanz angegeben wird.

Beispiel: Angenommen, eine Host- oder System i-Datenbank wird mit dem Datenbankaliasnamen db1 auf dem DB2 Connect-Server S1 (mit dem Hostnamen db2conn1 und der Portnummer 122) katalogisiert. Der Datenbankadministrator möchte einen alternativen DB2 Connect-Server S2 mit dem Hostnamen db2conn2 und der Portnummer 123 angeben. Der Befehl, den der Datenbankadministrator auf dem DB2 Connect-Server S1 ausführen muss, lautet wie folgt:

```
db2 update alternate server for database db1 using hostname db2conn2 port 123
```

Nach der Angabe der Position des alternativen DB2 Connect-Servers für den Datenbankaliasnamen db1 auf dem DB2 Connect-Server S1 wird die Information über die Position des alternativen Servers als Teil des Verbindungsprozesses an den IBM Data Server-Client zurückgegeben. Wenn die Kommunikation zwischen dem IBM Data Server-Client und dem DB2 Connect-Server S1 aus einem bestimmten Grund (normalerweise aufgrund eines Kommunikationsfehlers, wie z. B. SQL-Code -30081 oder SQL-Code -1224) unterbrochen wird, versucht der IBM Data Server-Client, die Verbindung zu db1 über den ursprünglichen DB2 Connect-Server (S1) oder über den alternativen DB2 Connect-Server (S2) wiederherzustellen, wobei zwischen den beiden Servern abgewechselt wird. Das Zeitintervall zwischen den Versuchen ist zu Beginn kurz und wird mit jedem Versuch allmählich länger.

Nach erfolgreicher Herstellung einer Verbindung wird der SQL-Code -30108 zurückgegeben. Dieser Wert gibt an, dass eine Datenbankverbindung nach einem Kommunikationsfehler wiederhergestellt wurde. Der Hostname bzw. die IP-Adresse und der Servicename bzw. die Portnummer werden zurückgegeben. Der IBM Data Server-Client gibt die Ursache für den ursprünglichen Kommunikationsfehler nur dann an die Anwendung zurück, wenn die Wiederherstellung der Clientkommunikation weder für den ursprünglichen noch für den alternativen Server möglich ist.

Die folgenden Aspekte hinsichtlich der Konnektivität über alternative Server in einer DB2 Connect-Serverumgebung sind ebenfalls zu beachten:

- Wenn Sie einen DB2 Connect-Server dazu verwenden, Zugriff auf eine Host- oder System i-Datenbank sowohl für ferne als auch für lokale Clients bereitzustellen, kann es bei den Konnektivitätsinformationen für den alternativen Server in einem Systemdatenbankverzeichniseintrag zu Unklarheiten kommen. Zur Vermeidung dieser Unklarheiten sollten im Systemdatenbankverzeichnis für eine Host- oder System i-Datenbank zwei Einträge katalogisiert werden. Katalogisieren Sie einen Eintrag für ferne Clients und einen Eintrag für lokale Clients.
- Die von einem DB2 für z/OS-Zielservers zurückgegebenen SYSPLEX-Informationen werden nur im Cache auf dem DB2 Connect-Server gespeichert. Nur ein einziger alternativer Server wird auf die Platte geschrieben. Wenn mehrere alternative oder mehrere aktive Server vorhanden sind, sind die Informationen nur im Hauptspeicher vorhanden und gehen beim Beenden des Prozesses verloren.

Kapitel 13. Konfigurieren der automatischen Clientweiterleitung für die Distributortechnologie für die Clientverbindung

Distributor- oder Dispatchertechnologien wie z. B. WebSphere EdgeServer verteilen Verbindungswiederholungsanforderungen der Clientanwendung an eine definierte Gruppe von Systemen, wenn ein primärer Datenbankserver fehlschlägt. Wenn Sie Distributortechnologie zusammen mit der automatischen Clientweiterleitung von DB2 verwenden, müssen Sie den Distributor selbst als alternativen Server für die automatische Clientweiterleitung von DB2 angeben.

Sie arbeiten möglicherweise mit Distributortechnologie in einer Umgebung, die der folgenden Umgebung ähnlich ist:

Client → Distributortechnologie → (DB2 Connect Server 1 oder DB2 Connect Server 2) → DB2 z/OS

Dabei gilt:

- Die Komponente der Distributortechnologie hat den TCP/IP-Hostnamen 'DHostname'
- Der DB2 Connect Server 1 hat den TCP/IP-Hostnamen 'GWYhostname1'.
- Der DB2 Connect Server 2 hat den TCP/IP-Hostnamen 'GWYhostname2'.
- Der DB2 z/OS-Server hat den TCP/IP-Hostnamen 'zOShostname'.

Auf dem Client wird **DHostname** katalogisiert, damit die Distributortechnologie für den Zugriff auf einen der DB2 Connect-Server genutzt werden kann. Die zwischengeschaltete Distributortechnologie entscheidet darüber, ob **GWYhostname1** oder **GWYhostname2** verwendet wird. Nachdem die Entscheidung getroffen ist, verfügt der Client über eine direkte Socketverbindung zu einem dieser beiden DB2 Connect-Gateways. Sobald die Socketverbindung zum gewünschten DB2 Connect-Server besteht, verfügen Sie über die typische Konnektivität zwischen Client, DB2 Connect-Server und DB2 z/OS.

Nehmen Sie zum Beispiel an, dass der Distributor **GWYhostname2** auswählt. Dadurch ergibt sich die folgende Umgebung:

Client -> DB2 Connect Server 2 -> DB2 z/OS

Wenn ein Kommunikationsfehler auftritt, versucht der Distributor nicht, Verbindungen wiederherzustellen. Wenn Sie die Funktion der automatischen Clientweiterleitung für eine Datenbank in einer solchen Umgebung aktivieren möchten, müssen Sie als alternativen Server für die zugeordnete Datenbank (bzw. Datenbanken) auf dem DB2 Connect-Server (DB2 Connect Server 1 oder DB2 Connect Server 2) den Distributor (DHostname) angeben. Wenn dann der DB2 Connect Server 1 aus einem irgendeinem Grund gesperrt wird, wird die automatische Clientweiterleitung ausgelöst, und es wird versucht, erneut eine Clientverbindung herzustellen, wobei der Distributor sowohl als primärer als auch als alternativer Server eingesetzt wird. Diese Option ermöglicht es Ihnen, die Funktionen des Distributors mit der DB2-Funktion zur automatischen Clientweiterleitung zu kombinieren und zu verwalten. Wenn Sie den alternativen Server auf einen anderen Hostnamen setzen als den Hostnamen des Distributors, wird den Clients die Funktion der automatischen Clientweiterleitung weiterhin zur Verfügung gestellt. Die Clients stellen

jedoch direkte Verbindungen zu dem definierten alternativen Server her und umgehen die Distributortechnologie, wodurch der Nutzen des Distributors verloren geht.

Die Funktion für die automatische Clientweiterleitung fängt die folgenden SQL-Codes ab:

- SQLCODE-Wert -20157
- SQLCODE-Wert -1768 (Ursachencode = 7)

Anmerkung: Die Clientweiterleitung wird möglicherweise nicht in zeitgerechter Form über Socketfehler informiert, wenn der Konfigurationsparameter "TCP Keepalive" des Betriebssystems auf einen zu hohen Wert eingestellt ist. (Beachten Sie, dass der Name dieses Konfigurationsparameters je nach Plattform variiert.)

Teil 4. Optimierung und DB2 Connect

In einer Datenbankumgebung, in der DB2 Connect für die Übertragung von Datenbankanforderungen und entsprechenden Antworten zwischen Client-Workstations und Datenbankservern verwendet wird, müssen spezielle Aspekte hinsichtlich der Leistungsoptimierung berücksichtigt werden. Es stehen mehrere Möglichkeiten zur Verfügung, das Leistungsniveau in einer solchen Umgebung aufrechtzuerhalten bzw. zu verbessern.

Kapitel 14. DB2 Connect - Leistungsinformationen

Leistung ist das Verhalten eines Datenverarbeitungssystems bei einer bestimmten Auslastung. Sie wird von den verfügbaren Ressourcen und deren Verwendung sowie dem gemeinsamen Zugriff auf sie beeinflusst. Wenn die Leistung verbessert werden soll, muss zuerst klar sein, was mit Leistung gemeint ist. Es kann eine Vielzahl unterschiedlicher *Leistungsmessdaten* verwendet werden, wie z. B.:

Antwortzeit

Der zeitliche Abstand zwischen dem Senden der Datenbankanforderung und dem Empfangen einer Antwort durch die Anwendung.

Transaktionsdurchsatz

Die Anzahl von UOWs, die pro Zeiteinheit ausgeführt werden können. Eine UOW kann einfach sein wie das Abrufen und Aktualisieren einer Zeile oder sehr kompliziert, sodass sie Hunderte von SQL-Anweisungen umfasst.

Datenübertragungsgeschwindigkeit

Die Anzahl an Datenbyte, die pro Zeiteinheit zwischen der DB2 Connect-Anwendung und der Host- oder System i-Datenbank übertragen werden.

Die Leistung wird durch die verfügbaren Hardware- und Softwareressourcen begrenzt. CPU, Hauptspeicher und Netzwerkkadapters sind Beispiele für Hardwareressourcen. Kommunikationssysteme, Pagingssysteme und mbuf für AIX sind Beispiele für Softwareressourcen.

Datenfluss

Abb. 12 auf Seite 102 zeigt den Datenfluss zwischen dem Host- oder System i-Datenbankserver und der Workstation über DB2 Connect.

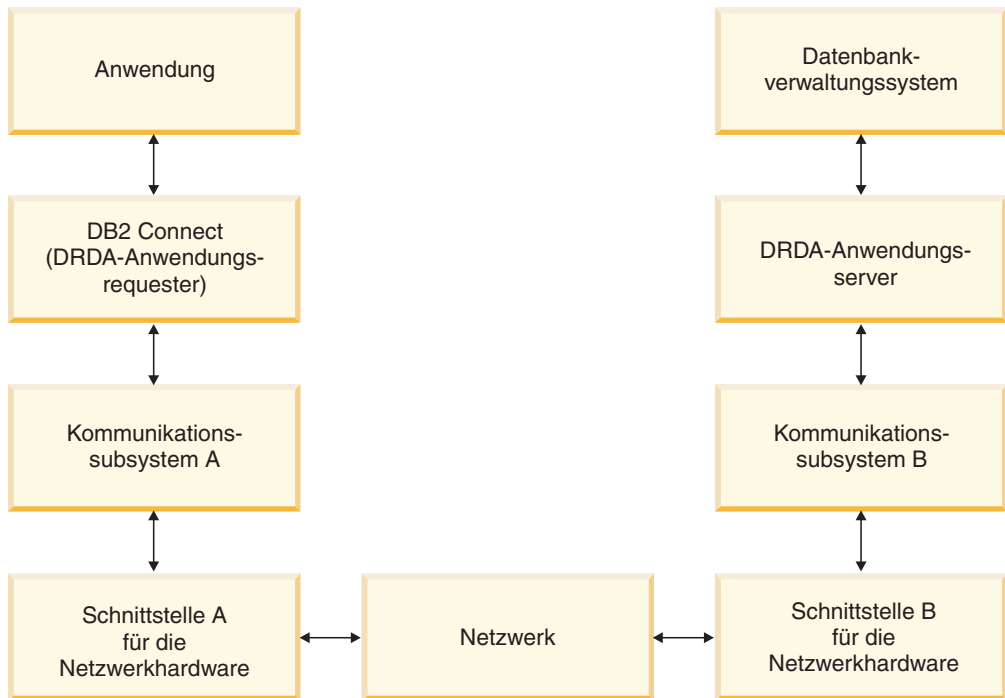


Abbildung 12. Datenfluss in DB2 Connect

- Die Host- oder System i-Datenbank und ein Teil des Kommunikations-subsystems B werden normalerweise auf demselben System ausgeführt. Dieses System besteht aus einer oder mehreren CPU(s), Hauptspeicher, einem E/A-Subsystem, DASD und einem Betriebssystem. Da andere Programme gemeinsam auf diese Komponenten zugreifen können, können durch Ressourcenkonkurrenz Probleme bei der Leistung entstehen.
- Das Netzwerk besteht aus einer Kombination von Kabeln, Hubs, Übertragungsleitungen, Schaltern und anderen Kommunikationssteuereinheiten. Beispielsweise könnte die Schnittstelle B für die Netzwerkhardware aus Kommunikationssteuereinheiten wie der Steuereinheit IBM 3745 oder 3172 oder einem Token-Ring-Adapter für einen System i-Server bestehen. Es kann auch mehr als ein Übertragungsmedium zwischen den Schnittstellen A und B für die Netzwerk-Hardware verwendet werden.
- Die Schnittstelle A für die Netzwerk-Hardware kann ein Token-Ring-Adapter, ein Ethernet-Adapter oder ein anderer LAN-Adapter bzw. ein Adapter sein, der das SDLC- oder X.25-Protokoll unterstützt.
- DB2 Connect und das Kommunikationssystem A befinden sich normalerweise auf demselben System. Im Rahmen dieser Erörterung wird davon ausgegangen, dass die Anwendung sich ebenfalls auf demselben System befindet.

Engpässe bei der Leistung

Der Transaktionsdurchsatz hängt von der langsamsten Komponente im System ab. Wenn ein Engpass bei der Leistung festgestellt wird, kann das Problem oft durch Ändern von Konfigurationsparametern, Zuordnen zusätzlicher Ressourcen zur Problemkomponente, Aufrüsten/Aktualisieren der Komponente oder Entlasten der Komponente durch Hinzufügen einer neuen Komponente gemildert bzw. behoben werden.

Es stehen verschiedene Tools zur Verfügung, mit denen ermittelt werden kann, wie viel Zeit in jeder Komponente für eine Abfrage benötigt wird. Hierdurch können Sie eine Vorstellung bekommen, welche Komponenten zur Verbesserung der Leistung optimiert oder aufgerüstet/aktualisiert werden sollten. Wenn z. B. festgestellt wird, dass eine Abfrage 60% ihrer Zeit in der DB2 Connect-Maschine zubringt, ist es möglicherweise sinnvoll, DB2 Connect zu optimieren oder (wenn ferne Clients vorhanden sind) dem Netzwerk eine weitere DB2 Connect-Maschine hinzuzufügen.

Ausführung von Vergleichstest

Bei der Ausführung von *Vergleichstests* wird die Leistung in einer Umgebung mit der Leistung in einer anderen Umgebung verglichen. Das Ausführen von Vergleichstests kann mit dem Ausführen der Testanwendung in einer normalen Umgebung beginnen. Während ein Leistungsproblem immer weiter eingegrenzt wird, können spezialisierte Anwendungsbeispiele entwickelt werden, um den Bereich der getesteten und beobachteten Funktion zu verringern.

Die ausgeführten Vergleichstests müssen nicht komplex sein. Spezialisierte Anwendungsbeispiele müssen nicht eine gesamte Anwendung emulieren, um wichtige Informationen zu liefern. Beginnen Sie mit einfachen Messungen, und erhöhen Sie die Komplexität nur, wenn dies erforderlich ist.

Merkmale guter Vergleichspunkte:

- Jeder Test ist wiederholbar.
- Jede Wiederholung eines Tests wird in einem identischen Systemstatus gestartet.
- Die Hardware und die Software, die für die Vergleichstests verwendet werden, entsprechen Ihrer Produktionsumgebung.
- Im System sind ausschließlich die gemessenen Funktionen oder Anwendungen aktiv. (Es sei denn, der Test verlangt, dass eine andere Aktivität im System aktiv ist.)

Anmerkung: Anwendungen, die gestartet wurden, belegen Speicher, selbst wenn sie auf Symbolgröße verkleinert wurden oder momentan inaktiv sind. Dies könnte Paging verursachen und die Ergebnisse des Tests verzerren.

Leistungsanalyseprogramme

In den folgenden Tabellen sind einige Tools aufgelistet, die beim Messen der Systemleistung helfen können. Da diese Tools selbst Systemressourcen belegen, sollten sie nicht ständig aktiv sein.

Tabelle 14. Leistungsanalysetools für die CPU-Belastung und die Speicherbelegung

System	Tool	Beschreibung
AIX	vmstat, time, ps, tprof	Bieten Informationen zu Konkurrenzsituationen beim Zugriff auf die CPU oder den Hauptspeicher auf der DB2 Connect-Workstation und fernen Clients.
HP-UX	vmstat, time, ps, monitor und glance, wenn diese verfügbar sind.	
Windows	Microsoft Performance Monitor	

Tabelle 15. Leistungsanalysetools für die Datenbankaktivität

System	Tool	Beschreibung
Alle	Datenbankmonitor	Stellt fest, ob das Problem von der Datenbank ausgeht.
OS/390 oder zSeries	DB2PM (IBM), OMEGAMON/DB2 (Candle), TMON (Landmark), INSIGHT (Goal Systems) und DB2AM (BMC)	
Windows	Microsoft Performance Monitor	

Tabelle 16. Leistungsanalysetools für Netzwerkaktivitäten

System	Tool	Beschreibung
AIX	netpmon	Erstellt einfache Systemstatistiken für das Netzwerk, einschließlich TCP/IP-Statistiken, wie die Anzahl pro Sekunde empfangener Pakete oder Rahmen.
Netzwerksteuereinheit, z. B. Steuereinheit 3745	NetView Performance Monitor	Erstellt Berichte über die Auslastung von Kommunikationssteuerung und VTAM.
Linux und UNIX	netstat	Verwaltet den TCP/IP-Verkehr.

Kapitel 15. Optimierung des ODBC-Zugriffs

DB2 stellt besondere Optimierungsmöglichkeiten zur Verbesserung der Kommunikationsleistung über ODBC zur Verfügung. Diese Erweiterungen sind für Microsoft Access, Lotus Approach und Visual Basic verfügbar. Sie haben den Vorteil eines schnelleren ODBC-Durchsatzes mithilfe des Konfigurationsassistenten von DB2.

Um die ODBC-Optimierung zu aktivieren, gehen Sie wie folgt vor:

- Beim Definieren einer neuen Verbindung:
 1. Starten Sie den DB2-Konfigurationsassistenten (CA).
 2. Öffnen Sie das Menü **Ausgewählt**, und wählen Sie **Datenbank mit Assistent hinzufügen...** aus.
 3. Gehen Sie durch die Seiten des Assistenten, bis Sie auf die Seite **Datenquelle** gelangen.
 4. Aktivieren Sie **Datenbank für CLI/ODBC registrieren**.
 5. Geben Sie an, wie CLI/ODBC-Anwendungen, die auf diese Datenbank zugreifen, registriert werden sollen:
 - **Als Systemdatenquelle** bedeutet, dass alle Benutzer auf dem System auf die Datenquelle zugreifen können.
 - **Als Benutzerdatenquelle** bedeutet, dass Sie als einziger Benutzer auf die Datenquelle zugreifen können.
 - **Als Dateidatenquelle** bedeutet, dass eine Datei mit den Informationen aus der Datenquelle erstellt wird. Diese Datenquellendatei kann über eine TCP/IP-Verbindung mit mehreren Workstations gemeinsam genutzt werden. Ohne TCP/IP-Verbindung lässt sie sich nur auf diesem Computer nutzen.
 6. Geben im Feld **Name der Datenquelle** einen Namen ein.
 7. (Optional) Wählen Sie aus der Liste **Für Anwendung optimieren** eine Anwendung aus, damit die Einstellungen der Datenquelle für diese Anwendung optimiert werden.
 8. Klicken Sie **OK** an, und verlassen Sie den Konfigurationsassistenten.
- Beim Aktualisieren einer bestehenden Verbindung:
 1. Starten Sie den DB2-Konfigurationsassistenten (CA).
 2. Klicken Sie den Aliasnamen der zu optimierenden Datenbank doppelt an.
 3. Klicken Sie **Datenquelle** an.
 4. Aktivieren Sie **Datenbank für CLI/ODBC registrieren**.
 5. Geben Sie an, wie CLI/ODBC-Anwendungen, die auf diese Datenbank zugreifen, registriert werden sollen:
 - **Als Systemdatenquelle** bedeutet, dass alle Benutzer auf dem System auf die Datenquelle zugreifen können.
 - **Als Benutzerdatenquelle** bedeutet, dass Sie als einziger Benutzer auf die Datenquelle zugreifen können.
 - **Als Dateidatenquelle** bedeutet, dass eine Datei mit den Informationen aus der Datenquelle erstellt wird. Diese Datenquellendatei kann über eine TCP/IP-Verbindung mit mehreren Workstations gemeinsam genutzt werden. Ohne TCP/IP-Verbindung lässt sie sich nur auf diesem Computer nutzen.

6. Geben im Feld **Name der Datenquelle** einen Namen ein.
7. (Optional) Wählen Sie aus der Liste **Für Anwendung optimieren** eine Anwendung aus, damit die Einstellungen der Datenquelle für diese Anwendung optimiert werden.
8. Klicken Sie **OK** an, und verlassen Sie den Konfigurationsassistenten.

Kapitel 16. Anwendungsentwurf

Beim Erstellen einer Anwendung stehen verschiedene Methoden zur Leistungssteigerung zur Verfügung.

Compound-SQL-Anweisungen und gespeicherte Prozeduren

Bei Anwendungen, die viele Befehle und Antworten senden und empfangen, kann der Systemaufwand des Netzwerks beträchtlich sein. Compound-SQL-Anweisungen und gespeicherte Prozeduren stellen zwei Arten dar, diesen Systemaufwand zu verringern.

Wenn eine Anwendung mehrere SQL-Anweisungen ohne dazwischenliegende Programmlogik sendet, kann eine Compound-SQL-Anweisung verwendet werden. Wenn Programmlogik in den Gruppen von SQL-Anweisungen erforderlich ist, können gespeicherte Prozeduren verwendet werden.

Alle ausführbaren Anweisungen mit Ausnahme der folgenden können in einer Compound-SQL-Anweisung enthalten sein:

```
CALL
    FETCH
    CLOSE
    OPEN
    Compound SQL
    Connect
    Prepare
    Release
    Describe
    Rollback
    Disconnect
    Set connection
    execute immediate
```

Mit gespeicherten Prozeduren kann der Datenaustausch auf dem Netzwerk reduziert werden, weil die Programmlogik auf den Server verlagert wird. Die Daten können beim Verlassen der Prozedur automatisch festgeschrieben werden. Sie können auch Ergebnismengen zurückgeben, die die Anwendungslogik auf dem Client minimieren.

Gruppierung von Anforderungen

Das Gruppieren zusammengehöriger Datenbankanforderungen (SQL-Anweisungen) zu einer Datenbankanforderung kann die Anzahl über das Netzwerk übertragener Anforderungen und Antworten verringern.

Wenn z. B. die beiden folgenden Anweisungen:

```
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=1
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=2
```

zu einer Anweisung gruppiert werden:

```
SELECT COL1, COL2, COL5, COL6 FROM TABLEA WHERE ROW_ID=1 OR ROW_ID=2
```

werden weniger Anforderungen über das Netzwerk gesendet.

Es können auch Schlüsselwörter wie IN und BETWEEN verwendet werden, um die Anzahl zurückgegebener Zeilen zu verringern. Außerdem können die Schlüsselwörter WHERE, IN und BETWEEN in den Anweisungen UPDATE und DELETE verwendet werden.

Prädikatenlogik

Prädikatenlogik kann verwendet werden, um nur die Zeilen und Spalten anzufordern, die tatsächlich benötigt werden. Hierdurch wird der Datenaustausch auf dem Netzwerk und die CPU-Belastung für die Datenübertragung verringert.

Folgende Abfrage sollte beispielsweise nicht verwendet werden:

```
SELECT * FROM TABLEA
```

wenn nur die erste Zeile von TABLEA with ROW_ID=1 benötigt wird oder wenn nur Spalte 1 und Spalte 2 benötigt werden.

Datenblockung

Es sollte Datenblockung verwendet werden, wenn große Datenmengen vom Server erwartet werden. Durch Blockung wird die Ausnutzung der Netzwerkbandbreite verbessert und die CPU-Belastung sowohl des Host- oder System i-Datenbankservers als auch des DB2 Connect-Servers verringert. Jede gesendete und empfangene Nachricht bewirkt unabhängig von ihrer Größe einen bestimmten Grad von CPU-Belastung und eine bestimmte Menge an Systemaufwand im Netzwerk. Durch Datenblockung wird die Anzahl von Nachrichten verringert, die für eine bestimmte Menge übertragener Daten erforderlich ist.

Wenn Blockung verwendet wird, wird die erste Zeile der Daten einer Abfrage erst dann an die Anwendung übergeben, wenn der erste Block empfangen wird. Durch Blockung wird die Abrufzeit für die erste Zeile vergrößert, aber die Abrufzeit für nachfolgende Zeilen wird verkürzt.

Eine andere Überlegung bezieht sich auf die verwendete Menge von Speicher. Die Menge eingelagerter Seiten erhöht sich normalerweise, wenn die Blockung aktiviert wird.

In DB2 Connect kann die Datenmenge gesteuert werden, die mit jedem Block übertragen wird.

Zum Aktivieren der Blockung kann die Option BLOCKING des Befehls prep oder bind verwendet werden. Die Blockung ist aktiviert, wenn Folgendes zutrifft:

- Auf den Cursor besteht nur Lesezugriff.
- Der Cursor ist mehrdeutig, und während der Ausführung des Befehls prep bzw. bind wird die Blockung angegeben.

Anmerkung: Bei der Verwendung von dynamischem SQL ist der Cursor immer mehrdeutig.

SQL-Anweisungen mit Blockung

Aktualisierbare SELECT-Anweisungen (die Anweisungen UPDATE/DELETE WHERE CURRENT OF verwenden) stellen nicht geblockte Abfragen dar, daher sollten sie nur verwendet werden, wenn dies absolut erforderlich ist.

Eine aktualisierbare SELECT-Anweisung stellt sicher, dass die Zeile zwischen dem Zeitpunkt des Beendens von SELECT und dem Zeitpunkt der Eingabe von UPDATE/DELETE nicht geändert wird. Wenn diese Stufe des gemeinsamen Zugriffs für Ihre Anwendung nicht wichtig ist, kann stattdessen DELETE oder UPDATE mit Suchkriterien verwendet werden, die auf von einer nicht aktualisierbaren SELECT-Anweisung übergebenen Werten basieren.

Für SELECT mit Lesezugriff geben Sie FOR FETCH ONLY an (außer unter VM und VSE, da es dort nicht unterstützt wird).

Statisches und dynamisches SQL

Verwenden Sie so oft wie möglich statisches SQL. Hierdurch werden mehrdeutige Cursor und die Vorbereitung von SQL-Abschnitten zur Laufzeit vermieden. Wenn dynamisches SQL sich nicht vermeiden lässt, können folgende Maßnahmen ergriffen werden, damit der Datenaustausch auf dem Netzwerk minimiert und die Leistung verbessert wird:

- Wenn die Anweisung eine Anweisung SELECT ist und eine Vorbereitung erforderlich ist, führen Sie PREPARE ... INTO für einen SQL-Deskriptorbereich durch. Für den SQL-Deskriptorbereich sollte die vollständige Größe zugeordnet werden, die für die verwendeten Einstellungen erforderlich ist. Wenn die maximale Anzahl von Spalten x beträgt und sich voraussichtlich nichts ändern wird, sollte ein SQL-Deskriptorbereich mit x SQLVARs zugeordnet werden. Wenn die Anzahl möglicher Spalten unsicher ist (und Hauptspeicher keinen Engpass darstellt), sollte die maximale Anzahl von SQLVARs (256) verwendet werden.

Wenn der zugeordnete Wert für den SQL-Deskriptorbereich nicht groß genug zum Speichern des übergebenen SQL-Deskriptorbereichs ist, muss das Programm eine weitere Anweisung DESCRIBE mit einem SQL-Deskriptorbereich absetzen, der groß genug ist, um das Ergebnis erneut zu speichern. Hierdurch wird der Datenaustausch auf dem Netzwerk erhöht.

Die Folge aus PREPARE und DESCRIBE sollte nicht verwendet werden. Die Verwendung der Anweisung PREPARE.....INTO bietet eine bessere Leistung.

- Binden Sie statische gebundene SQL-Anweisungen COMMIT oder ROLLBACK statt dynamischer Anweisungen COMMIT oder ROLLBACK.
- Wenn es sich nicht um eine Anweisung SELECT, COMMIT oder ROLLBACK handelt, sollte EXECUTE IMMEDIATE statt einer Folge aus PREPARE und EXECUTE zum Ausführen der Anweisung eingegeben werden.
- ODBC-Anwendungen verwenden dynamisches SQL. Sie können die Leistung unter Umständen mithilfe der statischen CLI/ODBC-Profilierungsfunktion verbessern. Mit dieser Funktion können Sie ODBC-Aufrufe erfassen und in statische Anweisungen umsetzen, die in einem Datenbankpaket gespeichert werden. Die tatsächliche Leistung, die Sie erhalten, hängt von der Komplexität Ihrer Anwendung ab.

Andere Überlegungen zu SQL

Die Verwendung des Befehlszeilenprozessors (CLP) ist im Allgemeinen langsamer als die Verwendung von dynamischer SQL im Programm, weil der CLP die Eingabedaten syntaktisch analysieren muss, bevor die SQL-Anweisungen an die Datenbanksteuerkomponente übergeben werden. Der CLP formatiert die Daten auch bei deren Eingang, was für Ihre Anwendung möglicherweise gar nicht erforderlich ist.

SQL-Anweisungen in einer interpretierten Sprache (z. B. REXX) sind wesentlich langsamer als dieselben SQL-Anweisungen in einer Compilersprache (z. B. C).

Es gibt zwei Typen der Anweisung CONNECT, Typ 1 und Typ 2. Bei Verbindungen des Typs 2 wird bei der Herstellung einer Verbindung zu einer Datenbank die vorherige Verbindung in einen Ruhezustand versetzt, aber nicht abgebrochen. Durch das spätere Umschalten zu einer im Ruhezustand befindlichen Verbindung kann der Systemaufwand für das Laden von Bibliotheken und das Einrichten interner Datenstrukturen umgangen werden. Aus diesem Grund kann die Verwendung von Verbindungen des Typs 2 die Leistung von Anwendungen verbessern, die auf mehrere Datenbanken zugreifen.

Kapitel 17. Verbindungsverwaltung

Die Verbindungsverwaltung besteht aus zwei Operationen: Verbindungspooling und Verbindungskonzentration. Das Verbindungspooling reduziert den Systemaufwand bei Datenbankverbindungen und verwaltet das Verbindungsvolumen. Der Verbindungskonzentrator erhöht die Skalierbarkeit innerhalb der Arbeitsumgebung durch Optimieren der von den Hostdatenbankservern verwendeten Ressourcen. Beide Operationen werden hier erläutert.

Verbindungspooling

DB2 Connect-Serverprodukte, wie z. B. DB2 Connect Enterprise Edition, stellen oft Datenbankverbindungen für Tausende von gleichzeitigen Clientanforderungen zur Verfügung. Das Herstellen und Trennen von Verbindungen zu Datenbankservern ist ein Prozess, der sehr viele Ressourcen verbrauchen kann und einen negativen Einfluss sowohl auf die Leistung von Datenbankservern als auch von DB2 Connect-Servern hat.

Dieses Problem zeigt sich insbesondere in Webumgebungen, in denen bei jedem Besuch auf einer Webseite der Aufbau einer neuen Verbindung zum Datenbankserver, die Ausführung einer Abfrage und die Trennung der Verbindung erforderlich werden kann. Um diesen Systemaufwand zu reduzieren, verwenden DB2 Connect-Serverprodukte das Verbindungspooling, um offene Verbindungen zu Datenbanken in einem leicht verfügbaren Pool zu verwalten.

Die meisten Anwendungen auf der Basis von Webtechnologien verarbeiten große Volumina an kurzen Transaktionen. Eine typische Webtransaktion wird als Teil ihrer eigenen Verbindung ausgeführt. Mit anderen Worten: Das Ausführen einer Transaktion umfasst die Herstellung einer Datenbankverbindung und das Beenden dieser Verbindung nach nur wenigen SQL-Anweisungen. Dieser Prozess des Aufbaus und Trennens einer Verbindung verbraucht sehr viele Ressourcen. Er umfasst das Erstellen eines DB2 Connect-Agenten, die Herstellung einer Netzwerkverbindung zwischen diesem Agenten und dem DB2-Server sowie das Erstellen eines DB2-Threads auf dem Server. Für Verbindungen, die über einen längeren Zeitraum aktiv sind, lohnt sich dieser Systemaufwand im Hinblick auf die vielen Transaktionen, die über diese Verbindung ausgeführt werden. Bei typischen Webtransaktionen übersteigt dieser Systemaufwand jedoch normalerweise den Aufwand für die Ausführung der Transaktion selbst.

Beim Verbindungspooling handelt es sich um eine Methode, die es ermöglicht, eine aufgebaute Verbindungsinfrastruktur für nachfolgende Verbindungen erneut zu verwenden. Beim Start einer DB2 Connect-Instanz wird auch ein Pool von Koordinationsagenten erstellt. Beim Empfang einer Verbindungsanforderung wird dieser Anforderung ein Agent zugeordnet. Der Agent stellt eine Verbindung zum DB2-Server her, und in DB2 wird ein Thread erstellt. Gibt die Anwendung eine Anforderung zum Trennen der Verbindung aus, übergibt der Agent diese Anforderung nicht an den DB2-Server. Stattdessen wird der Agent in den Pool zurückgestellt. Der Agent im Pool ist weiterhin der Eigner seiner Verbindung zum DB2-Server und des entsprechenden DB2-Threads. Gibt eine andere Anwendung eine Verbindungsanforderung aus, wird dieser Agent der neuen Anwendung zugeordnet. Um die Sicherheit der Verarbeitung zu gewährleisten, werden Benutzeridentifikationsinformationen an den DB2-Thread übergeben, der anschließend die Identität des Benutzers überprüft.

Das von DB2 Connect zur Verfügung gestellte Verbindungspooling führt zu einer deutlichen Leistungsverbesserung in solchen Umgebungen. DB2 Connect verwaltet offene Verbindungen zur Datenbank in einem verfügbaren Pool. Wenn ein Client eine Verbindung anfordert, kann diese aus dem Pool der bereiten Verbindungen zur Verfügung gestellt werden. Das Verbindungspooling führt zu einer beträchtlichen Reduzierung des Systemaufwands, der normalerweise für das Öffnen und Schließen dieser Verbindungen erforderlich ist.

Das Verbindungspooling ist für Anwendungen transparent, die über DB2 Connect eine Verbindung zum Host herstellen. Wenn eine Anwendung das Trennen der Verbindung zum Host anfordert, löscht DB2 Connect die eingehende Verbindung mit der Anwendung, belässt die abgehende Verbindung zum Host jedoch in einem Pool. Wenn eine neue Anwendung eine Verbindung anfordert, verwendet DB2 Connect eine Verbindung aus dem bestehenden Pool. Die Verwendung der bereits bestehenden Verbindung reduziert die allgemeine Verbindungszeit sowie den hohen Ressourcenverbrauch für die CPU-Verbindung auf dem Host.

DB2 Connect-Agenten können sich in einem von zwei Status befinden: inaktiv oder aktiv. Ein Agent ist aktiv, wenn er eine Aktion für eine Anwendung ausführt. Sobald diese Aktion beendet ist, geht der Agent in den inaktiven Status über und wartet auf weitere Aktionen von derselben oder einer anderen Anwendung. Alle inaktiven Agenten werden zusammen in einem Pool für inaktive Agenten verwaltet. Die Größe dieses Pools kann mithilfe des Konfigurationsparameters **num_poo-lagents** definiert werden. Dieser Parameter entspricht der maximalen Anzahl der inaktiven Agenten, die das System verwalten soll. Wird der Parameter auf null gesetzt, bedeutet dies, dass die Funktion des Verbindungspoolings ausgeschaltet ist. Standardmäßig wird für diesen Konfigurationsparameter AUTOMATIC mit dem Standardwert 100 definiert. Die Definition von AUTOMATIC bewirkt, dass DB2 Connect die Anzahl der inaktiven Agenten im entsprechenden Pool automatisch verwaltet.

DB2 Connect stellt Verbindungen zur Datenbank erst dann her, wenn die erste Client-Anforderung empfangen wird. Alternativ können Sie den Pool der inaktiven Agenten jedoch füllen, bevor Anforderungen von Clients ausgegeben werden. Der Pool kann beim Start durch die Verwendung des Konfigurationsparameters **num_initagents** gefüllt werden. Dieser Parameter gibt an, wie viele inaktive Agenten beim Initialisieren erstellt werden sollen. Diese inaktiven Agenten haben zunächst keine Verbindung zum Hostdatenbankserver.

Wenn ein Client eine Verbindung zum Host anfordert, versucht DB2 Connect, aus dem Pool einen Agenten abzurufen, der bereits über eine Verbindung zum Hostdatenbankserver verfügt. Schlägt dies fehl, sucht DB2 Connect einen Agenten im Pool der inaktiven Agenten. Wenn der Pool leer ist, erstellt DB2 Connect einen neuen Agenten.

Sie können die maximale Anzahl der gleichzeitig aktiven Agenten über den Konfigurationsparameter **max_coordagents** steuern. Sobald diese Anzahl überschritten wird, schlagen neue Verbindungen mit dem SQL-Fehlercode SQL1226 fehl. (Dieser Code bedeutet, dass die maximale Anzahl der gleichzeitig bestehenden abgehenden Verbindungen überschritten wurde.) Standardmäßig wird für diesen Konfigurationsparameter AUTOMATIC mit dem Standardwert 200 definiert. Die Definition von AUTOMATIC bewirkt, dass DB2 Connect die Anzahl der Koordinatoragenten automatisch verwaltet.

Die DB2-Registrierdatenbankvariable DB2CONNECT_IN_APP_PROCESS lässt zu, dass für Anwendungen, die auf derselben Maschine wie das DB2 Connect-Server-

produkt ausgeführt werden, entweder DB2 Connect innerhalb des Anwendungsprozesses ausgeführt wird (Standardverhalten) oder dass eine Verbindung zum DB2 Connect-Serverprodukt hergestellt wird und die Host-Verbindung dann innerhalb des Agenten ausgeführt wird. Damit eine Anwendung das Verbindungspooling verwenden kann, müssen die Verbindungen zum Host über die Agenten des DB2 Connect-Serverprodukts erfolgen. Demnach muss der Parameter DB2CONNECT_IN_APP_PROCESS auf NO (Nein) gesetzt werden.

DB2 Connect-Verbindungspooling im Vergleich zum Verbindungspooling von Anwendungsservern

Die Funktion des Verbindungspoolings ist unabdingbar für alle Anwendungen auf der Basis von Webtechnologien, die große Transaktionsvolumina unterstützen müssen. Die meisten Webanwendungsserver stellen heute ihre eigenen Methoden zum Pooling von Datenbankverbindungen zur Verfügung. Sowohl Microsoft MTS (COM+) als auch IBM WebSphere ermöglichen das Verbindungspooling.

Die von diesen Servern implementierten Methoden für das Verbindungspooling unterscheiden sich deutlich von der Funktion, die von den DB2 Connect-Servern zur Verfügung gestellt wird. Da Anwendungsserver die Verbindungen nur zur eigenen Verwendung in einem Pool zusammenschließen, gehen sie normalerweise davon aus, dass die Benutzer-IDs, Kennwörter und Isolationsstufen usw. für alle Verbindungen identisch sind. Hierbei ist insbesondere zu beachten, dass Anwendungsserver nur solche Verbindungen zusammenschließen, die von demselben Prozess eingeleitet wurden. Dies bedeutet, dass Verbindungen von anderen Maschinen, Benutzern und Prozessen nicht im Pool zusammengeschlossen werden. Die von Anwendungsservern zur Verfügung gestellten Techniken für das Verbindungspooling sind zwar effektiv, wenn es darum geht, Verbindungen, die von derselben Instanz einer Anwendung aufgebaut wurden, erneut zu verwenden. Für den Zusammenschluss von Verbindungen von mehreren Benutzern, Servern etc. sind diese Techniken allerdings gänzlich ungeeignet.

Die von DB2 Connect-Servern zur Verfügung gestellte Funktion für Verbindungspooling hingegen ist völlig unabhängig von den jeweiligen Anwendungen, Maschinen und Benutzern. Bei Verbindungen von mehreren Clients können alle Anwendungsserver (auch wenn sie über verschiedene Benutzer-IDs verfügen) die Verbindungen aller anderen Server verwenden, was zu einer viel besseren Auslastung der zusammengeschlossenen Ressourcen führt.

Es stellt sich nun die Frage, welcher Typ von Verbindungspooling am besten zu verwenden ist. Die Antwort lautet: beide. Im Allgemeinen können das von DB2 Connect und das von Anwendungsservern zur Verfügung gestellte Verbindungspooling kombiniert verwendet werden. Da sie nicht miteinander konkurrieren, ist dies in jedem Fall eine gute Strategie. Auch wenn das Verbindungspooling über einen Anwendungsserver aktiviert ist, kann das Verbindungspooling über DB2 Connect verwendet werden, um Verbindungen für mehrere Anwendungsserver sowie für andere Clients, die den DB2 Connect-Server verwenden, wiederzuverwenden.

Verbindungskonzentrator

Der Verbindungskonzentrator senkt den Bedarf an Ressourcen, die auf Datenbankservers unter DB2 für OS/390 und z/OS erforderlich sind, um eine große Anzahl von Workstation- und Webbenutzern zu unterstützen. Diese Funktion kann die Skalierbarkeit Ihrer Lösungen unter DB2 für OS/390 und z/OS sowie DB2 Connect

drastisch erhöhen und sorgt gleichzeitig für störungssicheren Lastausgleich auf Betriebs- und Transaktionsebene in Umgebungen unter DB2 für OS/390 und z/OS mit gemeinsamer Datennutzung.

Der Verbindungskonzentrator ermöglicht es Anwendungen, ihre Verbindungen aufrecht zu erhalten, ohne dass auf dem DB2-Hostserver Ressourcen verbraucht werden. Auf diese Weise können Tausende von Benutzern in Anwendungen aktiv sein, während auf dem DB2-Hostserver lediglich einige wenige Threads aktiv sind.

Mithilfe der DB2 Connect-Technologie, die den Einsatz von *Verbindungskonzentratoren* vorsieht, können DB2 Connect-Serverprodukte wie DB2 Connect Enterprise Edition bei einem drastisch gesenkten Ressourcenverbrauch auf den S/390-Host- oder System i-Datenbankservern Unterstützung für Tausende von Benutzern zur Verfügung stellen, die gleichzeitig Geschäftstransaktionen ausführen. Dies wird durch eine Konzentration der Arbeitsbelastung aller Anwendungen auf eine viel kleinere Anzahl von Verbindungen zu S/390-Host- oder System i-Datenbankservern erreicht. Diese Methode scheint der oben beschriebenen Funktion des Verbindungspoolings zu ähneln. Es handelt sich jedoch um eine anspruchsvollere Methode zur Reduzierung des Ressourcenverbrauchs für OLTP-Anwendungen (Online-Transaktionsverarbeitung) mit sehr hohem Volumen.

Beim Verbindungskonzentrator wird das Konzept eines Agenten auf zwei Entitäten aufgeteilt:

- Der logische Agent entspricht der Anwendungsverbindung.
- Der koordinierende Agent ist der Eigner der DB2-Verbindung und des Threads und führt Anwendungsanforderungen aus.

Wenn eine neue Anwendung versucht, eine Verbindung zum Host aufzubauen, wird ihr ein logischer Agent zugewiesen. Ein koordinierender Agent ist erforderlich, um SQL-Anweisungen an die Datenbank weiterzuleiten. Er wird zugewiesen, sobald eine neue Transaktion eingeleitet wird. Entscheidend für diese Architektur sind folgende Punkte:

- Der koordinierende Agent ist vom logischen Agenten getrennt.
- Wenn eine Transaktion mit einer COMMIT-Operation oder einem Rollback beendet wird, kehrt der koordinierende Agent in den Pool zurück.

Ein weiteres wichtiges Merkmal ist die Methode zur Zuordnung von Koordinationsagenten für neue Transaktionen in einer Umgebung mit gemeinsamer Datenbenutzung. DB2 Connect implementiert einen fortgeschrittenen Zeitplanungsalgorithmus, der WLM-Informationen (Work Load Manager, Auslastungsverwaltung) von OS/390 und z/OS verwendet. Diese Informationen werden verwendet, um die Arbeitsbelastung gemäß den in WLM definierten Kriterien auf die Member einer Gruppe zu verteilen, die Daten gemeinsam benutzt. WLM kennt nicht nur die Auslastung der einzelnen Member, sondern auch ihre Verfügbarkeit. Dadurch ist DB2 Connect in der Lage, die Arbeitsbelastung von gestörten oder überlasteten Membern auf betriebsbereite und nicht ausgelastete Member zu verlagern. Der DB2 Connect-Verbindungskonzentrator wird aktiviert, indem die maximal zulässige Anzahl der logischen Agenten (*max_connections*) höher eingestellt wird als die Anzahl der Koordinationsagenten (*max_coordagents*).

Das Verbindungspooling spart Ressourcen bei der Herstellung einer Verbindung, wenn eine verwendet wird, die von einer beendeten Anwendung nicht mehr benötigt wird. Mit anderen Worten: Eine Anwendung muss ihre Verbindung erst trennen, bevor eine andere Anwendung diese in den Pool zurückgestellte Verbindung erneut verwenden kann.

Der Verbindungskonzentrator ermöglicht es DB2 Connect jedoch, einer Anwendung eine Verbindung zur Verfügung zu stellen, sobald eine andere Anwendung eine Transaktion beendet hat. Dazu muss diese andere Anwendung ihre Verbindung nicht trennen. Kurz gesagt: Eine Verbindung zu einem Datenbankserver und die damit verbundenen Host- und DB2 Connect-Ressourcen werden von einer Anwendung nur für den Zeitraum einer aktiven Transaktion verwendet. Sobald die Transaktion beendet wird, stehen die Verbindung und die zugeordneten Ressourcen einer beliebigen anderen Anwendung zur Verfügung, die eine Transaktion ausführen muss.

In früheren Versionen von DB2 Connect verfügte jede aktive Anwendung über eine Engine Dispatchable Unit (EDU), die sowohl die Datenbankverbindung als auch alle Anwendungsanforderungen verwaltete. Diese EDU wurde normalerweise als *Koordinationsagent* bezeichnet. Jeder Koordinationsagent protokollierte den Status oder den Kontext der Anwendung und EDU. Jede EDU verbraucht mit zunehmender Anzahl an Verbindungen eine beträchtliche Menge an Speicherkapazität, und der Kontextwechsel zwischen den Agenten führt zu einem zusätzlichen Systemaufwand.

In der oben beschriebenen Architektur besteht eine Eins-zu-eins-Beziehung zwischen Verbindungen und EDUs. Der Verbindungskonzentrator lässt jedoch eine Viele-zu-eins-Beziehung zwischen Verbindungen und EDUs zu. Dies bedeutet, dass die Beziehung von Verbindungen (X) zu EDUs (Y) jetzt $X \geq Y$ ist.

Der Verbindungskonzentrator teilt den Agenten in zwei Entitäten auf: einen *logischen Agenten* und einen *Verarbeitungsagenten*. Logische Agenten stellen eine Anwendung dar, jedoch ohne Verweis auf eine bestimmte EDU. Der logische Agent enthält alle Informationen und Steuerblöcke, die eine Anwendung benötigt. Wenn n Anwendungen mit einem Server verbunden sind, sind auch n logische Agenten auf dem Server vorhanden. Verarbeitungsagenten sind physische EDUs, die Anforderungen von Anwendungen ausführen, jedoch keiner bestimmten Anwendung permanent zugeordnet sind. Zwecks Ausführung von Transaktionen werden Verarbeitungsagenten logischen Agenten zugeordnet. An der Transaktionsgrenze wird diese Zuordnung wieder beendet, und die Verarbeitungsagenten werden erneut in den verfügbaren Pool gestellt.

Eine als *Dispatcher* bezeichnete Entität ordnet Verarbeitungsagenten logischen Agenten zu. Einschränkungen bei der Anzahl der offenen Dateikennungen auf bestimmten Datenverarbeitungsplattformen können dazu führen, dass mehr als eine Scheduler-Instanz ausgeführt wird.

Einschränkungen beim Verbindungskonzentrator

Die Verwendung des DB2 Connect-Serverkonzentrators unterliegt einigen wesentlichen Einschränkungen. Bitte lesen Sie die folgenden Informationen zunächst vollständig durch, bevor Sie versuchen, den Verbindungskonzentrator auf Ihrem System zu verwenden.

Allgemeine Einschränkungen:

- Der Konzentrator stellt eingehende Verbindungen von lokalen und fernen Clients über das TCP/IP-Protokoll her. Nur eingehende Verbindungen, die TCP/IP oder Local (IPC) verwenden, können zusammengeslossene abgehende Verbindungen nutzen. Der Konzentrator akzeptiert Verbindungen über andere Kommunikationsprotokolle, wie beispielsweise benannte Pipes. Mit diesen Verbindungen können die XA-Konzentrationsfunktionen allerdings nicht verwendet werden.

- Bei der eng gekoppelten XA-Transaktionsunterstützung müssen alle Anwendungen, die an derselben XA-Transaktion teilnehmen, die Verbindung zum Host über dieselbe DB2 Connect-Serverinstanz herstellen.
- Nur Anwendungen, die WITH HOLD-Ressourcen (z. B. WITH HOLD-Cursor) an den Transaktionsgrenzen schließen, können vom Konzentrator profitieren. Transaktionen, die WITH HOLD-Cursor nicht schließen, werden zwar weiterhin durchgelassen, werden jedoch einem dedizierten Verarbeitungsagenten zugeordnet und können daher nicht die gesamte Funktionsgruppe des Konzentrators nutzen.
- Wenn Sie globale temporäre Tabellen deklarieren, müssen diese an einer Transaktions- oder Verzweigungsgrenze explizit geschlossen werden. Wenn Sie die Tabellen nicht schließen, wird die Verbindungskonzentration inaktiviert, die Anwendung kann jedoch weiter verwendet werden.
- Alle Anwendungen, die an derselben XA-Transaktion teilnehmen, müssen über dieselbe ID für codierten Zeichensatz (CCSID) verfügen und zur Herstellung der Verbindung dieselbe Benutzer-ID verwenden.
- Wenn eine abgehende Verbindung zur Unterstützung einer zweiphasigen Verbindung hergestellt wurde, kann der Agent dieser Verbindung lediglich zur Unterstützung von zweiphasigen Verbindungen verwendet werden. Ebenso können Agenten, die zur Unterstützung einer einphasigen Verbindung eingerichtet werden, lediglich einphasige Verbindungen unterstützen.
- Der Konzentrator unterstützt lediglich dynamisches SQL über CLI (Call Level Interface). Darüber hinaus dürfen CLI-Anwendungen nicht KEEP DYNAMIC verwenden, da der Konzentrator davon abhängig ist, dass Anweisungen für jeden Transaktionsgrenzwert erneut vorbereitet werden.
- Dynamische Vorbereitungsanforderungen von Anwendungen mit eingebettetem dynamischen SQL werden zurückgewiesen. Ihre Anwendungen sollten so geändert werden, dass sie entweder statisches SQL verwenden oder dass die Verwendung dynamischer SQL-Anweisungen über die Befehlszeilenschnittstelle (CLI) erfolgt.

Bei Verwendung von DB2 Version 9 oder Version 8 FixPak 13 (oder höher) wird zur Aktivierung der DB2 Connect-Konzentratorunterstützung System i Version 5 Release 4 (PTF SI23726) benötigt. Andernfalls wird nur die XA-Komponente des Verbindungskonzentrators unterstützt.

Aktivierung des Verbindungskonzentrators

Über den Konfigurationsparameter *max_coordagents* des Datenbankmanagers wird die maximale Anzahl der logischen Agenten festgelegt. Sie können die Konzentratorkonfiguration aktivieren, indem Sie den Wert für *max_connections* auf einen beliebigen Wert über dem Standardwert festlegen. Der Standardwert für *max_connections* entspricht dem Wert von *max_coordagents*. Da jede Anwendung über einen logischen Agenten verfügen wird, steuert *max_connections* eigentlich die Anzahl der Anwendungen, die mit der Datenbankinstanz verbunden werden können, während *max_coordagents* die Anzahl der eingehenden Verbindungen steuert, die gleichzeitig aktiv sein können. Der Parameter *max_connections* nimmt einen numerischen Bereich von *max_coordagents* bis 64.000 an. Die Standardanzahl an logischen Agenten entspricht dem Wert von *max_coordagents*.

Sowohl *max_connections* als auch *max_coordagents* können mit AUTOMATIC definiert werden. Ist *max_connections* mit AUTOMATIC definiert, kann die Anzahl der Verbindungen auf einen Wert über dem konfigurierten Basiswert gesetzt werden. Ist sowohl *max_connections* als auch *max_coordagents* mit AUTOMATIC definiert,

kann *max_connections* auf einen Wert über dem Basiswert erhöht werden. Der Parameter *max_coordagents* wird in diesem Fall automatisch so erhöht, dass das Verhältnis zwischen Verbindungen und Koordinatoragenten bei der Verbindungskonzentration gewahrt bleibt.

Für die Konfiguration von Agenten werden verschiedene Konfigurationsparameter verwendet. Hierbei handelt es sich um folgende Parameter:

max_coordagents

Maximale Anzahl an aktiven Koordinationsagenten

num_poolagents

Größe des Agentenpools. Der Agentenpool umfasst inaktive Agenten und Agenten im Bereitschaftsmodus. Um die Leistung zu optimieren, sollte *num_poolagents* gleich der durchschnittlichen Anzahl an Clients sein.

num_initagents

Anfängliche Anzahl an Verarbeitungsagenten im Pool. Hierbei handelt es sich um inaktive Agenten.

XA-Transaktionsunterstützung

Über die Architektur des Verbindungskonzentrators kann DB2 Connect eine eng gekoppelte XA-Transaktionsunterstützung für DB2 für OS/390 und z/OS sowie DB2 für System i zur Verfügung stellen. Wie bei allen anderen Transaktionen auch, ordnet der Konzentrador einem Verarbeitungsagenten eine bestimmte XA-Transaktion (einzelne Transaktions-ID, XID) zu. Wenn die XA-Transaktion jedoch durch *xa_end()* (Verzweigungsgrenze) beendet wird, erfolgt für den Verarbeitungsagenten keine Freigabe für den allgemeinen Pool. Stattdessen bleibt der Verarbeitungsagent dieser bestimmten XA-Transaktion zugeordnet. Wenn eine andere Anwendung derselben XA-Transaktion zugeordnet wird, wird der Verarbeitungsagent dieser Anwendung zugeordnet.

Durch einen Transaktionsgrenzenaufruf wird der Agent an den Pool zurückgegeben. Beispielsweise durch *xa_prepare()* mit Lesezugriff, *xa_rollback()*, *xa_recover()*, *xa_forget()*, *xa_commit()* oder einen beliebigen XA-Fehler, der einen Rollback verursacht, wird der Agent an den normalen Pool zurückgegeben. *xa_end()* selbst beendet lediglich die Transaktionsverzweigung, und dies reicht nicht für eine Beendigung der Zuordnung zur XID aus.

Beispiele für XA-Transaktionsunterstützung

1. Stellen Sie sich eine Umgebung vor, in der mindestens 4.000 gleichzeitig bestehende Verbindungen benötigt werden. Web-Server, die CGI-Anwendungen verwenden, oder Büroanwendungen mit vielen Desktop-Benutzern können diese Anforderung überschreiten. In diesen Fällen ist aus Effizienzgründen normalerweise erforderlich, dass DB2 Connect als eigenständiger Gateway fungiert. Dies bedeutet, dass die Datenbank und das DB2 Connect-System sich auf getrennten Maschinen befinden.

Das DB2 Connect-Server-System ist unter Umständen nicht in der Lage, 4.000 gleichzeitig offene Verbindungen zur Datenbankmaschine zu verwalten. In den meisten Fällen ist die Anzahl der Transaktionen, die zu einem bestimmten Zeitpunkt ausgeführt werden, weitaus geringer als die Anzahl der gleichzeitig bestehenden Verbindungen. Der Systemadministrator könnte die Effizienz des Systems maximieren, indem er die Datenbankkonfigurationsparameter wie folgt einstellt:

```
MAX_CONNECTIONS = 4,000
MAX_COORDAGENTS = 1,000
NUM_POOLAGENTS  = 1,000
```

Der Konzentrator hält bis zu 4.000 gleichzeitig ablaufende Sitzungen offen, obwohl der Gateway lediglich 1.000 Transaktionen gleichzeitig verwaltet.

2. Im oben beschriebenen Beispiel werden von Verarbeitungsagenten ständig Zuordnungen zu logischen Agenten erstellt und aufgehoben. Diejenigen Agenten, die aktiv sind, können eine Verbindung zur Datenbank verwalten, nehmen jedoch an keiner bestimmten Transaktion teil; also stehen sie allen logischen Agenten (Anwendungen) zur Verfügung, die eine Verbindung anfordern.

Bei den XA-Transaktionen sieht es etwas anders aus. Für dieses Beispiel wird angenommen, dass ein TP-Monitor mit einem DB2 Connect-Gateway und einer zSeries- oder System i-Datenbank verwendet wird. Wenn eine Anwendung eine Verbindung anfordert, übergibt der Konzentrator entweder einen inaktiven Agenten, um diese Anforderung zu bedienen, oder er erstellt einen neuen Verarbeitungsagenten. Nehmen wir an, eine Anwendung fordert eine XA-Transaktion an. Für diese Transaktion wird eine XID erstellt, und der Verarbeitungsagent wird der Transaktion zugeordnet.

Nachdem die Anforderung der Anwendung ausgeführt wurde, wird `xa_end()` ausgegeben, und die Zuordnung zum Verarbeitungsagenten wird aufgehoben. Der Verarbeitungsagent bleibt der XID der Transaktion zugeordnet. Er kann jetzt lediglich mit der ihm zugeordneten XID Anforderungen für Transaktionen ausführen.

Zu diesem Zeitpunkt kann eine weitere Anwendung eine Anforderung für eine Nicht-XA-Transaktion ausgeben. Selbst wenn keine anderen Verarbeitungsagenten verfügbar sind, wird der für die XID zugeordnete Agent der zweiten Anwendung nicht zur Verfügung gestellt. Er wird als aktiv angesehen. Für die zweite Anwendung wird ein neuer Verarbeitungsagent erstellt. Wenn die zweite Anwendung ihre Transaktion beendet hat, wird ihr Verarbeitungsagent für den verfügbaren Pool freigegeben.

Währenddessen können andere Anwendungen, die die Transaktion anfordern, die der XID des ersten Agenten zugeordnet ist, eine Verbindung zu diesem Agenten herstellen oder unterbrechen. Dieser Agent führt seine dedizierte XA-Transaktion für sie aus. Alle Anwendungen, die diese bestimmte Transaktion anfordern, werden an diesen Verarbeitungsagenten gesendet, sofern er frei ist.

Der Verarbeitungsagent wird erst dann für den allgemeinen Pool freigegeben, wenn eine Anwendung einen Transaktionsgrenzenaufruf (nicht `xa_end()`) ausgibt. Beispielsweise kann eine Anwendung die Transaktion mit `xa_commit()` beenden. Sobald dies geschieht, löscht der Verarbeitungsagent seine Zuordnung zur XID und wird in den verfügbaren Pool zurückgestellt. Ab diesem Zeitpunkt können alle Anwendungen, die eine Anforderung ausgeben, diesen Agenten entweder für eine weitere XA-Transaktion oder eine Nicht-XA-Transaktion verwenden.

Verbindungspooling und Verbindungskonzentrator

Auch wenn das Verbindungspooling und der Verbindungskonzentrator scheinbar Ähnlichkeiten aufweisen, gibt es doch Unterschiede bei der jeweiligen Implementierung und Verwendung. Das Verbindungspooling hilft dabei, den Systemaufwand bei Datenbankverbindungen zu reduzieren und das Verbindungsvolumen zu verarbeiten. Der Verbindungskonzentrator hilft dabei, die Skalierbarkeit der jeweiligen Lösung unter DB2 für OS/390 und z/OS sowie DB2 Connect zu verbessern, indem die Verwendung der entsprechenden Hostdatenbankservers optimiert wird.

Wird Verbindungspooling verwendet, steht die Verbindung erst dann wieder zur erneuten Verwendung zur Verfügung, nachdem die Anwendung, die Eigner der Verbindung ist, eine Trennungsanforderung ausgibt. Bei vielen zweischichtigen Client/Serveranwendungen trennen die Benutzer ihre Verbindung während des Arbeitstages jedoch nicht. Ebenso stellen die meisten Anwendungsserver in vielschichtigen Anwendungen beim Start des Servers Datenbankverbindungen her und geben diese Verbindungen erst wieder frei, nachdem der Anwendungsserver heruntergefahren wurde.

In diesen Umgebungen zeigt das Verbindungspooling kaum oder gar keine Wirkung. In Web- und Client/Server-Umgebungen, in denen Verbindungen häufiger hergestellt und wieder getrennt werden, führt das Verbindungspooling hingegen zu deutlichen Leistungsverbesserungen. Der Verbindungskonzentrator ordnet Hostdatenbankressourcen lediglich für die Dauer einer SQL-Transaktion zu und hält Benutzeranwendungen gleichzeitig im aktiven Status. Dies ermöglicht Konfigurationen, bei denen die Anzahl der DB2-Threads und die von ihnen verbrauchten Ressourcen viel geringer sein können als in Fällen, bei denen jede Anwendungsverbindung über einen eigenen Thread verfügt.

Wenn es auf einen störungsfreien Betrieb und Lastausgleich ankommt, ist der Verbindungskonzentrator auf jeden Fall die richtige Wahl, da er bei jeder neuen Transaktion die erneute Zuordnung der Arbeitsbelastung ermöglicht. Mit dem Verbindungspooling hingegen ist ein Lastausgleich nur in begrenztem Maße und nur während der Verbindungszeit möglich.

Das Verbindungspooling und der Verbindungskonzentrator sollten zusammen verwendet werden, auch wenn sie unterschiedliche Aufgabenbereiche abdecken.

Mit WebSphere MQ Transaction Manager und DB2 für OS/390 erforderlicher Verbindungskonzentrator

Bei der Ausführung von Anwendungen in einer IBM WebSphere MQ-Umgebung (bisher unter der Bezeichnung IBM MQSeries bekannt), kann WebSphere MQ als XA-kompatibler Transaktionsmanager fungieren und beliebige verteilte Transaktionen mit zweiphasigem Commit koordinieren. Wenn WebSphere MQ auf diese Weise als Transaktionsmanager eingesetzt wird und die Datenquellen der DB2-Produktfamilie angehören, gelten eine Reihe von Konfigurationsanforderungen.

Die meisten Konfigurationsanforderungen in einer solchen Transaktionsmanagerumgebung sind bereits an anderer Stelle dokumentiert. So muss beispielsweise für den DB2-Konfigurationsparameter TP_MON_NAME der Wert „MQ“ auf dem DB2-Laufzeitclient definiert werden.

Eine Konfigurationsanforderung ist jedoch noch nicht dokumentiert. Diese Anforderung ist für DB2 Connect bei der Herstellung einer Verbindung zu DB2 für OS/390-Serverdatenquellen spezifisch: Wenn WebSphere MQ für die Koordination verteilter Transaktionen verwendet wird, an denen DB2 für z/OS- und DB2 für i5/OS-Server beteiligt sind, muss die Verbindungskonzentratorfunktion von DB2 Connect am Gateway aktiviert werden. Der Verbindungskonzentrator wird aktiviert, wenn der Wert des Konfigurationsparameters **max_connections** größer ist als der Wert für den Konfigurationsparameter **max_coordagents**.

Wenn Sie den Verbindungskonzentrator nicht aktivieren, führt dies zu unerwartetem Verhalten bei Transaktionen.

Kapitel 18. DB2 Connect - SYSPLEX-Unterstützung

Bei einem SYSPLEX-System handelt es sich um einen Verbund von System z-Servern, die zusammen mithilfe der verfügbaren Hard- und Software den vorhandenen Workload verarbeiten. Das SYSPLEX-System koordiniert dieses Zusammenwirken durch die Erhöhung der Anzahl der Prozessoren, die zusammen arbeiten. Auf diese Weise lässt sich das Durchsatzvolumen steigern. Zusätzlich zu dieser Verbesserung der Verarbeitungskapazität bietet ein SYSPLEX-System durch die Möglichkeit zum Einsatz verschiedener Hard- und Softwareebenen und die Fähigkeit zum dynamischen Hinzufügen neuer Systeme eine verbesserte Flexibilität.

Sysplex ermöglicht es DB2 Connect, Verbindungen über verschiedene Mitglieder einer Gruppe mit gemeinsamer Datennutzung hinweg unterbrechungsfrei auszugleichen. Darüber hinaus kann DB2 Connect mithilfe von Sysplex beim Ausfall eines Gruppenmitglieds alternative Mitglieder einsetzen. Die Umleitungsfunktion für Sysplex ist Teil der DB2 Connect-Funktionalität. Standardmäßig ist die DB2 Connect-Unterstützung für Sysplex ebenso wie die Umleitungsfunktion für Sysplex aktiviert. Die Sysplex-Unterstützung für eine Hostdatenbank kann inaktiviert werden, indem der Parameter SYSPLEX aus dem DCS-Verzeichniseintrag entfernt wird; der DCS-Eintrag selbst darf jedoch nicht entfernt werden, auch wenn darin kein weiterer Parameter angegeben ist.

Bei Verwendung der automatischen Clientweiterleitung für Sysplex ist das Standardverhalten einer für Sysplex aktivierten Verbindung, den Verbindungsaufbau erneut zu versuchen, wenn ein Kommunikationsfehler auftritt. Sonderregisterwerte bis zur letzten erfolgreichen Transaktion, die keine Ressourcen belegt, werden wiederholt, wenn DB2 Connect mit einem DB2 für z/OS-Server verbunden ist.

Mit den Registrierdatenbankvariablen DB2_MAX_CLIENT_CONNRETRIES und DB2_CONNRETRIES_INTERVAL können Sie das genaue Wiederholungsverhalten bei der automatischen Clientweiterleitung konfigurieren bzw. bei Bedarf inaktivieren. Die Registrierdatenbankvariable für das Verbindungszeitlimit lautet DB2TCP_CLIENT_CONTIMEOUT.

Überlegungen zur SYSPLEX-Ausnutzung auf OS/390- und zSeries-Systemen

DB2 Connect bietet Lastausgleichs- und Fehlertoleranzfunktionen für die Weiterleitung von Verbindungsanforderungen an mehrere SYSPLEX-Systeme. Bei einer Verbindung zu einem DB2 für OS/390 und z/OS-Datenbankserver in einer Umgebung mit gemeinsamer Datenbenutzung verteilt DB2 Connect den Workload auf die verschiedenen DB2-Subsysteme, aus denen sich die Gruppe der Systeme mit gemeinsamer Datenbenutzung zusammensetzt. Die Verteilung erfolgt hierbei auf der Basis der Informationen zur Systemauslastung, die vom Workload Manager (WLM) bereitgestellt werden.

DB2 Connect empfängt vom WLM eine nach Priorität geordnete Liste aller SYSPLEX-Teilnehmer. Jedes SYSPLEX-System gibt für die einzelnen Verbindungsadressen Informationen zur Prioritätsgewichtung zurück. Diese Liste wird anschließend von DB2 Connect zur Verarbeitung der eingehenden CONNECT-Anforderungen verwendet. Die Anforderungen werden auf die SYSPLEX-Teilnehmer mit der höchsten Priorität verteilt. Zum Lastausgleich wird die Liste mit den Informati-

onen zur Prioritätsgewichtung der verschiedenen SYSPLEX-Systeme während der Herstellung der einzelnen Verbindungen abgerufen. Wurde der DB2 Connect-Verbindungskonzentrator aktiviert, wird diese Liste auch zum Feststellen der Zieladresse für die zu versendenden Transaktionen benutzt.

Anmerkung: Die OS/390- und z/OS-DDF-Konfiguration muss nicht geändert werden, um SYSPLEX-Systeme unter DB2 Connect nutzen zu können.

DB2 Connect unterstützt die Fehlertoleranz außerdem dadurch, dass im Falle eines Verbindungsfehlers versucht wird, eine Verbindung zu einer anderen SYSPLEX-Maschine herzustellen. Ein Fehler wird erst dann an die Anwendung zurückgegeben, wenn alle bekannten Verbindungen fehlgeschlagen sind.

DB2 Connect-SYSPLEX wurde für die Verwendung mit Agentenpools entwickelt. Wenn SYSPLEX aktiviert ist, leitet DB2 Connect Verbindungen an ein anderes DDF-Member weiter, wenn die Verbindung zum ursprünglichen Member unterbrochen wird. Die Weiterleitung erfolgt gemäß einer SYSPLEX-Serverliste.

Durch das Hinzufügen des Konzentrators verfügt DB2 Connect nun über die Fähigkeit, die Auslastung an Transaktionsgrenzen auszugleichen. Der DB2 Connect-Konzentrator muss hierfür aktiviert werden.

DB2-SYSPLEX-Ausnutzung

In einem typischen Szenario kommuniziert ein DB2 Connect-Server (Server A) mit einem SYSPLEX-System, auf dem zwei DB2 für OS/390 und z/OS-Server (Server B und C) implementiert sind.

SYSPLEX-Server B	SYSPLEX-Server C
HOST_NAME=MVSHOST	HOST_NAME=MVSHOST1

Nehmen Sie beispielsweise an, eine Anwendung gibt in diesem Szenario Folgendes aus:

```
db2 connect to aliasb user xxxxxxx using xxxxxxxx
```

Die Verbindung zur Datenbank MVSHOST ist hergestellt. Da die SYSPLEX-Ausnutzung sowohl für den DB2 Connect-Server als auch den DCS-Verzeichniseintrag aktiviert wurde, identifiziert DB2 für OS/390 und z/OS für DB2 Connect die Netzwerkadressen für jeden der SYSPLEX-Teilnehmer (MVSHOST und MVSHOST1. Zum Zurückgeben dieser Informationen werden DRDA4-Protokolle und -Nachrichtenabläufe verwendet). Sobald eine einleitende Verbindung hergestellt ist, wird die zurückgegebene Liste von Adressen auf der DB2 Connect-Workstation zwischengespeichert. Sobald die anfängliche CONNECT-Anweisung für einen TCP/IP-Knoten ausgegeben wurde, werden die IP-Adressen zurückgegeben.

Prioritätsinformationen für den Lastausgleich und die Fehlertoleranz

Die Liste der Adressen, die von DB2 für OS/390 und z/OS bereitgestellt wird, enthält auch Prioritätsinformationen wie die Anzahl von Verbindungen für jede Netzwerkadresse. Die Liste wird von DB2 Connect bei jedem Herstellen einer neuen Verbindung aktualisiert. Diese zusätzlichen Informationen werden für den Lastausgleich und für die Fehlertoleranz benötigt.

Zwischengespeicherte Adressliste für DB2 Connect

Wenn die Datenbankverbindung zu ALIASB fehlschlägt, wird die Fehlermeldung SQL30081N ausgegeben, und die Verbindung wird gelöscht. Wenn für ALIASB eine weitere Verbindungsanforderung empfangen wird, werden von DB2 Connect die folgenden Schritte ausgeführt:

1. DB2 Connect versucht, eine Verbindung zum Server mit der höchsten Priorität herzustellen, der in der zwischengespeicherten Adressliste aufgeführt ist. Hierbei werden die Prioritätsinformationen herangezogen, die von DB2 für OS/390 und z/OS zurückgegeben wurden. Dieses Verfahren wird von DB2 Connect immer eingesetzt. Auf diese Weise wird der Lastausgleich erzielt.
2. Wenn dieser Verbindungsversuch fehlschlägt, werden die anderen Adressen in der Liste (in absteigender Prioritätsfolge) verwendet, die von DB2 für OS/390 und z/OS zurückgegeben wurde. Auf diese Weise nutzt DB2 Connect die SYSPLEX-Informationen zur Gewährleistung der Fehlertoleranz.
3. Wenn alle anderen Verbindungsversuche fehlschlagen, versucht DB2 Connect erneut die Herstellung einer Verbindung zu ALIASB mit der Adresse, die sich im Katalogknotenverzeichnis befindet.

Wenn Sie den Befehl `db2pd` mit dem Parameter `sysplex` verwenden (`db2pd -sysplex`) erhalten Sie Informationen über die Server, die einer Sysplex-Umgebung zugeordnet sind.

Konfigurationsvoraussetzungen für SYSPLEX-Systeme

Die SYSPLEX-Ausnutzung wird für eine bestimmte Datenbank nur dann verwendet, wenn für den DCS-Verzeichniseintrag dieser Datenbank im sechsten Positionsparameter `Sysplex` (Groß-/Kleinschreibung muss nicht beachtet werden) definiert ist.

Kapitel 19. DB2 Connect-Optimierung

Verschiedene Parameter in der Konfigurationsdatei des Datenbankmanagers können zum Optimieren von DB2 Connect verwendet werden.

RQRIOBLK

Der Parameter **RQRIOBLK** legt die maximale Größe der Netzwerk-E/A-Blöcke fest. Eine Vergrößerung des Blocks kann die Leistung großer Anforderungen steigern. Die Blockgröße hat normalerweise keine Auswirkungen auf die Antwortzeiten kleiner Anforderungen, z. B. Anforderungen für eine einzige Datenzeile.

Für eine größere Blockgröße ist in der Regel mehr Hauptspeicher auf dem DB2 Connect-Server erforderlich. Hierdurch wird die Menge der eingelagerten Seiten vergrößert, was in manchen Fällen zu einer erheblichen Zunahme der Paging-Rate auf kleinen Workstations führen kann.

Verwenden Sie den Standardwert für die DRDA-Blockgröße (32767), sofern er nicht zu starke Pagingaktivitäten bei der Ausführung der Anwendung verursacht. Andernfalls sollte die E/A-Blockgröße reduziert werden, bis kein Paging mehr stattfindet. Sobald Paging stattfindet, ist eine deutliche Verschlechterung der Leistung festzustellen. Mithilfe von Tools zur Leistungsüberwachung (wie beispielsweise 'vmstat' für Linux und UNIX-Betriebssysteme) können Sie feststellen, ob auf dem verwendeten System Paging stattfindet.

DIR_CACHE

Der Parameter **DIR_CACHE** legt fest, ob Verzeichnisinformationen in einem Cache zwischengespeichert werden. Wenn Zwischenspeicherung angegeben ist (DIR_CACHE=YES), werden Verzeichnisdateien gelesen und im Hauptspeicher zwischengespeichert, um den Systemaufwand zu verringern, der für das Erstellen der internen Verzeichnisstruktur und das Lesen der Verzeichnisdateien bei jedem Herstellen einer Verbindung erforderlich ist.

Ohne Zwischenspeicherung (DIR_CACHE=NO) wird bei jedem Herstellen einer Verbindung zu einer Datenbank das entsprechende Verzeichnis von einer Platte gelesen und anschließend die Suche ausgeführt. Nachdem die angeforderten Einträge gefunden wurden, werden alle für die Verzeichnissuche zugeordneten Speicherbereiche freigegeben.

Bei der Verwendung von Zwischenspeicherung wird während der Verarbeitung von db2start ein gemeinsamer Verzeichniscache erstellt und beim Beenden von DB2 freigegeben. Dieser Cache wird von allen DB2-Serverprozessen (db2agent) verwendet. Außerdem wird ein privater Verzeichnis-Cache für die Anwendung eingerichtet, wenn eine Anwendung ihre erste Verbindung zu einer Datenbank herstellt, und wieder freigegeben, wenn die Anwendung beendet wird.

Jeder Cache stellt ein Abbild des Systemdatenbankverzeichnisses, des DCS-Verzeichnisses für Datenbankverbindungen und des Knotenverzeichnisses zur Verfügung. Durch den Cache wird der Aufwand für die Verbindung verringert, indem die E/A-Operationen für Verzeichnisdateien vermieden und Suchoperationen in Verzeichnissen verringert werden.

Wenn ein im Cache zwischengespeichertes Verzeichnis aktualisiert wird, werden die Änderungen nicht sofort an die Caches weitergegeben. Wenn ein Verzeichniseintrag in einem Cache nicht gefunden wird, werden die ursprünglichen Verzeichnisdateien durchsucht.

Durch Zwischenspeicherung wird die Menge des privaten Speichers erhöht, der für die Ausführung einer Anwendung erforderlich ist. Ohne Zwischenspeicherung wird dieser Speicher nur benötigt, wenn eine Suchfunktion für ein Verzeichnis verarbeitet wird. Die Verwendung gemeinsam benutzten Speichers durch DB2 erhöht sich insgesamt leicht, weil Verzeichnisinformationen, auf die Datenbankagenten gemeinsam zugreifen, in den gemeinsam benutzten Speicher verlegt werden. Die Größe des für einen Cache benötigten Speichers hängt von der Anzahl von Einträgen ab, die in jedem Verzeichnis definiert sind.

NUMDB

Das Verhalten von DB2 Connect wurde vom Konfigurationsparameter **NUMDB** in vorherigen Versionen nicht beeinflusst, doch dies änderte sich ab Version 8. Dieser Parameter gibt die maximale Anzahl von Datenbanken an, zu denen der Client über den DB2 Connect-Server eine Verbindung herstellen kann. Genauer gesagt, die maximale Anzahl verschiedener Datenbank-Aliasnamen, die auf dem DB2 Connect-Server katalogisiert werden können.

Weitere DB2 Connect-Parameter

AGENTPRI und **MAXAGENTS** sind ab Version 9.5 veraltet.

Befehle zum Aktualisieren des Werts für **MAXAGENTS** sind weiterhin ausführbar, sodass vorhandene Anwendungen nicht beschädigt werden, die Werte werden jedoch ignoriert. Der Parametername wird in keiner Konfigurationsliste aufgeführt. Bisher wurde die Gesamtzahl der Agenten, die in einer bestimmten DB2-Partition erstellt werden konnten, durch den Konfigurationsparameter **MAXAGENTS** gesteuert. Nun kann die Konfiguration von Agenten automatisiert werden.

Standardmäßig wird für **NUM_POOLAGENTS** **AUTOMATIC** mit dem Standardwert 100 definiert. Darüber hinaus wird standardmäßig für **MAX_COORDAGENTS** **AUTOMATIC** mit dem Standardwert 200 definiert.

Um Abrechnungszeichenfolgen von Ihrer Clientanwendung an den DB2 Connect-Server zu senden, verwenden Sie Funktionen der API zum Einstellen der Abrechnungsdaten. Die Funktionen der API sind schneller und leistungsfähiger als der Einsatz der Umgebungsvariablen **DB2ACCOUNT**.

IBM Data Server Driver for JDBC and SQLJ

Merkmal `com.ibm.db2.jcc.DB2BaseDataSource.clientAccountingInformation`

DB2 .NET Data Provider

Merkmal `DB2Connection.ClientAccountingInformation`

CLI/ODBC

Konfigurationsschlüsselwort `ClientAcctStr` CLI/ODBC

Eingebettetes SQL (C, C++ und COBOL)

Funktion `sqlsact`

Wenn keine speziell angepasste **SQLCODE**-Zuordnungsdatei benötigt wird, kann die Leistung durch Verwendung der standardmäßigen **SQLCODE**-Zuordnung oder durch Ausschalten der **SQLCODE**-Zuordnung verbessert werden. Die Standardzu-

ordnungsdatei ist in die DB2 Connect-Bibliothek eingebettet; eine angepasste Zuordnungsdatei muss von der Platte gelesen werden, wodurch die Leistung beeinträchtigt wird.

Optimierung der Hostdatenbank

Die Systemleistung wird durch die Leistung des Host- oder System i-Datenbank-servers beeinflusst. Verschiedene Datenbankverwaltungssysteme haben verschiedene Leistungsmerkmale. SQL-Optimierungsprogramme verschiedener Systeme können sich z. B. bei derselben Anwendung unterschiedlich verhalten. Weitere Informationen können Sie der Leistungsbeschreibung in der Dokumentation des verwendeten Host- oder System i-Datenbankserversystems entnehmen.

Es kann möglicherweise eine Leistungsverbesserung erzielt werden, indem die Bindeoption für nicht festgeschriebene Lesevorgänge (UR, Uncommitted Read) oder für 'kein Commit' (NC, No Commit) verwendet wird, sodass keine Aufzeichnung stattfindet.

Anmerkung: Wird UR verwendet, können nicht aufgezeichnete Daten nur gelesen, jedoch nicht aktualisiert werden, und dies auch nur dann, wenn die Blockung auf ALL eingestellt ist.

Je nach dem verwendeten Anwendungsserver und der von ihm zur Verfügung gestellten Unterteilung für Sperren kann die für eine Abfrage oder Anwendung verwendete Isolationsstufe einen erheblichen Einfluss auf die Leistung ausüben. Die Datenbank sollte über eine geeignete Normalisierungsstufe, eine effiziente Verwendung von Indizes und eine sinnvolle Zuordnung von Datenbankbereich verfügen. Die Leistung kann auch durch die verwendeten Datentypen beeinflusst werden, wie in den nachfolgenden Abschnitten beschrieben wird.

Überlegungen zur Netzwerkoptimierung

Die beste Methode, die Gesamtleistung in einer verteilten Datenbankumgebung zu verbessern, ist das Eliminieren von Verzögerungen, die durch das Netzwerk verursacht werden. Netzwerkadministratoren gehen in der Regel davon aus, dass ein Netzwerk effektiver ist, wenn es so viele Daten wie möglich zwischen Übertragungen sammelt. Dies wirkt sich jedoch nachteilig auf Anwendungen für verteilte Datenbanken aus, weil dadurch in das Netzwerk Verzögerungen eingebaut werden. Der Endbenutzer sieht die Effektivität des Netzwerks nicht, sondern lediglich die Verzögerungen.

Für die meisten Netzwerkeinheiten gibt es Verzögerungsparameter, und die Mehrzahl der Parameter nimmt standardmäßig Werte an, die für verteilte Datenbanken sehr schlecht sind. Sie können die Leistung steigern, indem Sie diese Parameter suchen und, sofern möglich, auf null setzen. Stellen Sie zudem sicher, dass die Puffergröße für die Einheit groß genug ist, um erneute Übertragungen aufgrund verlorener Daten zu verhindern. UNIX-Systeme zum Beispiel weisen in der Regel den Wert 32 für die Warteschlangenlänge bei Übertragungs- und Empfangsvorgängen auf. Setzen Sie die Warteschlangenlänge zum Erzielen besserer Ergebnisse auf 150. Ein entsprechender Parameter in den DLC-Einstellungen ist die Empfangslänge, die auch auf 150 gesetzt werden sollte.

Der Parameter IOBUF ist an den meisten Standorten zu niedrig eingestellt. Er ist gewöhnlich auf 500 gesetzt. Erfahrungsgemäß eignet sich jedoch der Wert 3992 am besten, wenn Sie große Datenmengen, besonders bei Kanalverbindungen wie ESCON oder 3172, übertragen.

In einem LAN-System kann sich die Größe der DLC- oder LLC-Übertragungs- und Empfangsfenster drastisch auf die Leistung auswirken. Der Sendewert sollte auf sieben oder mehr gesetzt werden, und für die meisten Konfigurationen ist ein Empfangswert von vier oder weniger am besten geeignet.

Wenn Sie Ethernet ausführen, sollten Sie die TCP-Segmentgröße auf 1500 Byte setzen. In einem Token-Ring- oder FDDI-Netzwerk sollte dieser Wert 4400 Byte sein, und wenn Sie einen ESCON-Adapter mit TCP/IP verwenden, sollte die Segmentgröße immer 4096 sein.

Schließlich sollte die Größe der Sende- und Empfangspuffer für TCP bei TCP/IP-Netzwerken auf einen höheren Wert als 32768 gesetzt werden. Der Wert 65536 ist im Allgemeinen am besten.

Anmerkung: Das Herstellen einer Verbindung vom Gateway zum Server (abgehende Verbindung) ist wesentlich kostenintensiver als das Herstellen einer Verbindung von einem Client zum Gateway (eingehende Verbindung). In einer Umgebung, in der Tausende von Clients häufig über den Gateway eine Verbindung zum Server herstellen und trennen, wird ein hoher Prozentsatz der Verarbeitungszeit für das Herstellen von abgehenden Verbindungen benötigt. DB2 Connect stellt daher ein Verbindungspooling über TCP/IP bereit. Wenn ein Client das Trennen der Verbindung vom Server anfordert, löscht der Gateway die eingehende Verbindung mit dem Client, belässt die abgehende Verbindung zum Server jedoch in einem Pool. Wenn ein neuer Client am Gateway eine Verbindung anfordert, stellt der Gateway eine vorhandene Verbindung aus dem Pool bereit. Dadurch wird die Verbindungszeit insgesamt verringert, und es entfällt der hohe CPU-Verbindungsaufwand auf dem Server.

Eine Zusammenfassung der Methoden zur Verbesserung der Netzleistung finden Sie in Tabelle 17.

Tabelle 17. Methoden zur Verbesserung der Netzleistung

Kritischer Bereich	Beispiel	Einstellung	Hinweise
Absichtliche Verzögerungen	Verzögerungsparameter für Netzwerkeinheiten	Auf 0 setzen.	Die Standardwerte sind gewöhnlich höher.
Puffer	Parameter IOBUF	Auf bis zu 3992 setzen.	Besonders hilfreich für ESCON- oder andere Kanaladapter
Puffer	RUSIZE	Optimale Größe ist 4096.	Das Setzen von RUSIZE und RQRIOBLK auf dieselbe Größe ergibt möglicherweise die beste Leistung.
Puffer	Nachrichtendosierung	VPACING, PACING und Modusprofile sollten auf 63 gesetzt werden.	Verwenden Sie adaptive Nachrichtendosierung, wo möglich.
Adaptoreinstellungen	Warteschlangenlänge für Senden/Empfangen	Der empfohlene Wert ist 150.	Der Standardwert ist gewöhnlich 32.

Tabelle 17. Methoden zur Verbesserung der Netzleistung (Forts.)

Kritischer Bereich	Beispiel	Einstellung	Hinweise
TCP-Einstellungen	Segmentgrößen	1500 für Ethernet, 4400 für Token-Ring und FDDI.	Für TCP/IP verwendete ESCON-Adapter sollten immer auf 4096 gesetzt werden.
TCP-Einstellungen	Speicherbereichsgrößen für Senden/Empfangen	Sollte für beide 64 KB sein.	Der Standardwert für Windows beträgt lediglich 8192. Der Wert kann in der Windows-Registrierungsdatenbank eingestellt werden.

Konflikte bei Systemressourcen

Die Leistung kann sich verschlechtern, wenn viele Tasks im System versuchen, gleichzeitig auf bestimmte Systemressourcen zuzugreifen. Folgende Fragen müssen beantwortet werden:

- Ist die CPU überlastet? Gegebenenfalls sollte das System aufgerüstet, die Systemauslastung vermindert und/oder das System optimiert werden, um den Systemaufwand für die Verarbeitung zu verringern.
- Ist der Hauptspeicher überlastet? Gegebenenfalls sollte der Hauptspeicher aufgerüstet, die Systemauslastung verringert und/oder das System zur Verringerung der eingelagerten Seiten optimiert werden.
- Ist der Kommunikationsadapter bzw. die Kommunikationssteuereinheit zu stark ausgelastet? Gegebenenfalls sollte das Netzwerk erweitert oder sollten Token-Ring-Karten hinzugefügt werden.
- Ist eines der Subsysteme zu stark ausgelastet, und liegt dieses Subsystem im Datenpfad?
- Laufen im System unnötige Prozesse oder Tasks? Im Allgemeinen sollten nur Services konfiguriert oder gestartet werden, wenn sie regelmäßig verwendet werden, um eine Verschwendung von Systemressourcen zu vermeiden.
- Belegen einige wenige Prozesse oder Funktionen einen Großteil der Ressourcen? Können sie gestoppt werden? Kann ihre Priorität verringert werden? Können sie so geändert werden, dass sie weniger Ressourcen benötigen?

DB2 Connect-Fehlerbehebung

Wenn DB2 Connect-Benutzer lange Antwortzeiten bei großen Abfragen von Host- oder System i-Servern feststellen, sollten folgende Bereiche auf mögliche Ursachen für das Leistungsproblem untersucht werden:

1. Für Abfragen, die große Datenblöcke vom Host- oder System i-Server (gewöhnlich 32 KB Daten und mehr) zurückgeben, muss sichergestellt werden, dass der Konfigurationsparameter RQRIOLBK des Datenbankmanagers auf 32767 gesetzt ist. Dies kann folgendermaßen mit dem Befehlszeilenprozessor durchgeführt werden:


```
db2 update database manager configuration using RQRIOLBK 32767
```
2. Die maximale RU-Größe in der IBMRDB-Modusdefinition muss auf einen geeigneten Wert eingestellt sein. Empfehlenswert sind mindestens 4 KB für Ver-

bindungen mit Token-Ring-Hardware. Für Verbindungen mit Ethernet-Hardware beträgt die maximale Ethernet-Rahmengröße 1536 Byte. Dies kann ein einschränkender Faktor sein.

Optimierung von DB2 für OS/390 und z/OS

Sie können die Verarbeitung inaktiver Threads unter OS/390 und z/OS optimieren. In Version 5 sind maximal 25.000 gleichzeitig verbundene Clients zulässig. In allen Fällen beträgt die maximale Anzahl gleichzeitig aktiver Clients jedoch 1.999. Jeder Workstation-Client kann verbunden bleiben, wenn er inaktiv ist; sein Thread wird bei jedem Commit in eine Kette inaktiver Clients gestellt.

Die DSNZPARM-Parameter CMTSTAT, CONDBAT und MAXDBAT beeinflussen die Thread-Verarbeitung. Setzen Sie CMTSTAT auf INACTIVE, passen Sie CONDBAT an die maximale Anzahl verbundener DBATs bei guter Leistung an, und setzen Sie MAXDBAT auf die maximal zulässige Anzahl aktiver DBATs, um die beste Leistung zu erzielen.

Erhöhung der Datenübertragungsgeschwindigkeit unter DB2 Connect

DB2 für OS/390 und z/OS ist nicht nur in der Lage, Zeilen für eine Abfrageergebnismenge zu blocken, sondern kann auch mehrere solcher Abfrageblöcke als Antwort auf eine OPEN- oder FETCH-Abfrage an einen fernen Client wie beispielsweise DB2 Connect zurückgeben. Daher ist es nicht mehr erforderlich, dass der Client wiederholt Anfragen an den Server von DB2 für OS/390 und z/OS sendet, die jeweils einen Block von Zeilendaten enthalten. Stattdessen kann der Client jetzt als Option anfordern, dass der Server zusätzlich zu dem ohnehin gesendeten Abfrageblock noch weitere Abfrageblöcke zurücksendet. Diese weiteren Abfrageblöcke werden als zusätzliche Abfrageblöcke bezeichnet.

Diese neue Funktion ermöglicht es dem Client, die Anzahl der Umschaltungen von Netzwerkleitungen, die erheblich zu Lasten der Netzleistung gehen, zu minimieren. Die geringere Anzahl der vom Client an den Server gesendeten Anforderungen von Abfrageblöcken führt zu einer deutlichen Leistungssteigerung. Diese Leistungssteigerung ist darauf zurückzuführen, dass das Umschalten der Leitungen zwischen dem Sende- und Empfangsmodus viele Ressourcen verbraucht, was sich negativ auf die Leistung auswirkt. DB2 Connect kann diese Leistungsverbesserung jetzt nutzen, indem standardmäßig zusätzliche Abfrageblöcke von einem Server unter DB2 für OS/390 und z/OS angefordert werden.

Um die Rückgabe zusätzlicher Abfrageblöcke (mit einer Länge von jeweils bis zu 32 KB) für das bevorzugte Netzwerkprotokoll TCP/IP optimal nutzen zu können, wurden Erweiterungen für die Fensterskalierung gemäß Architektur unter RFC-1323 in DB2 Connect aktiviert. Diese Funktion ermöglicht es TCP/IP, die Größe der Sende- und Empfangsfenster dynamisch anzupassen, um die potenziell großen Datenmengen, die mit den zusätzlichen Abfrageblöcken zurückgegeben werden, effizient zu integrieren.

Zusätzlicher Abfrageblock

Die Unterstützung für zusätzliche Abfrageblöcke auf Servern mit DB2 Universal Database (UDB) für OS/390 und z/OS Version 7 oder höher wird über den Parameter EXTRA BLOCKS SRV im DB2-DDF-Installationsfenster konfiguriert. Die Konfiguration dieser Unterstützung erfolgt über eine Steuerung der maximalen Anzahl an zusätzlichen Abfrageblöcken, die DB2 bei einer Anforderungen an einen Client zurücksenden kann. Sie können diesen Parameter auf einen Wert zwischen 0 und 100 einstellen. Wird der Parameter auf 0 eingestellt, wird die Rückgabe zusätz-

licher Abfrageblöcke inaktiviert. Zur optimalen Nutzung dieser Funktion sollte stets der Standardwert 100 verwendet werden. Dadurch werden jegliche Eigenheiten im Netzwerk ausgeschlossen, die zu einer geringeren Leistung dieser Einstellung führen könnten.

Auf der Clientseite, auf der die Anwendung entweder direkt über eine benachbarte DB2 Connect-Installation oder über eine separate DB2 Connect-Serverinstallation auf DB2 für z/OS zugreift, bestehen verschiedene Methoden zum Aktivieren der entsprechenden DB2 Connect-Unterstützung pro Cursor oder Anweisung:

- Angabe der Größe von Abfragezeilengruppen für einen Cursor
- Verwendung der Klausel 'OPTIMIZE for N ROWS' in der einem Cursor zugeordneten Auswahlanweisung
- Verwendung der Klausel 'FETCH FIRST N ROWS ONLY' in der einem Cursor zugeordneten Auswahlanweisung

DB2 Connect kann die Unterstützung für zusätzliche Abfrageblöcke über verschiedene SQL-APIs aktivieren:

Eingebettetes SQL

- Der Benutzer kann für eine Abfrage die Unterstützung zusätzlicher Abfrageblöcke aufrufen, indem in der Auswahlanweisung selbst entweder die Klausel 'OPTIMIZE for N ROWS' oder die Klausel 'FETCH FIRST N ROWS ONLY' oder beides angegeben wird.
- Bei Angabe der Klausel 'OPTIMIZE for N ROWS' versucht DB2 für OS/390 und z/OS, die gewünschte Anzahl der an DB2 Connect zurückgebenden Zeilen zu blocken. Grundlage hierfür ist die Einstellung des Installationsparameters EXTRA BLOCKS SRV DDF. Die Anwendung kann auch mehr als N Zeilen abrufen, da DB2 für z/OS die Gesamtanzahl der Zeilen, die für die Abfrageergebnismenge schließlich zurückgegeben werden kann, nicht auf N begrenzt.
- Die Klausel 'FETCH FIRST N ROWS ONLY' hat eine ähnliche Wirkung, mit dem Unterschied, dass die Abfrageergebnismenge in diesem Fall von DB2 für OS/390 und z/OS auf N Zeilen begrenzt ist. Werden mehr als N Zeilen abgerufen, wird der SQL-Code +100 (Datenende) ausgegeben.

CLI/ODBC

- Der Benutzer kann für eine Abfrage über das Anweisungsattribut SQL_MAX_ROWS Unterstützung für zusätzliche Abfrageblöcke aufrufen.
- Für Server unter DB2 UDB für OS/390 und z/OS 7.1 oder höher wird stattdessen die Klausel 'FETCH FIRST N ROWS ONLY' verwendet.
 - Bei Version 7 wird die Abfrageergebnismenge von DB2 für OS/390 und z/OS auf N Zeilen begrenzt. Werden mehr als N Zeilen abgerufen, wird SQL_NO_DATA_FOUND ausgegeben.
 - Ab Version 8 stellt die Befehlszeilenschnittstelle (CLI) sicher, dass nur die ersten N Reihen über den Cursor Manager des Clients an die Anwendung zurückgegeben werden.

JDBC Der Benutzer kann für eine Abfrage über die Methode 'setMaxRows' Unterstützung für zusätzliche Abfrageblöcke aufrufen. Ähnlich wie bei der CLI/ODBC-Aktivierung fügt DB2 Connect dann für einen Server unter DB2 für OS/390 und z/OS 6.x die Klausel 'OPTIMIZE for N ROWS' hinzu. Für Server unter DB2 für z/OS 7.1 oder höher wird DB2 Connect ebenfalls die Klausel 'FETCH FIRST N ROWS ONLY' verwenden.

Fensterskalierung von RFC-1323

Die Funktion der Fensterskalierung wird auf allen Windows-, Linux- und UNIX-Plattformen unterstützt, die über Unterstützung der RFC-1323-Erweiterungen für TCP/IP verfügen. Sie können diese Funktion unter Verwendung der DB2-Registrierdatenbankvariablen DB2SORCVBUF unter DB2 für Windows, Linux oder UNIX aktivieren. Um die Fensterskalierung zu aktivieren, muss diese Registrierdatenbankvariable auf einen Wert über 64 KB gesetzt werden. Unter DB2 für Windows, Linux oder UNIX kann beispielsweise der Befehl `'db2set DB2SORCVBUF =65537'` abgesetzt werden.

Die Maximalgrößen der Sende- und Empfangspuffer hängen vom jeweils verwendeten Betriebssystem ab. Um zu gewährleisten, dass die konfigurierten Puffergrößen akzeptiert wurden, kann der Benutzer den Konfigurationsparameter DIAGLEVEL des Datenbankmanagers auf 4 (zur Information) einstellen und das Protokoll mit Benachrichtigungen für die Systemverwaltung auf Nachrichten überprüfen.

Damit die Funktion der Fensterskalierung wirksam wird, muss sie an beiden Enden der Verbindung, d. h. auf der Workstation und auf dem Host, aktiviert werden, entweder direkt über den TCP/IP-Stack des Betriebssystems oder indirekt über das DB2-Produkt. Bei DB2 für z/OS beispielsweise kann die Fensterskalierung derzeit lediglich über das Betriebssystem aktiviert werden, indem der Parameter TCPRCVBUFRSIZE auf einen beliebigen Wert über 64 KB eingestellt wird. Wenn Sie einen fernen IBM Data Server-Client verwenden, um über eine Server-Workstation von DB2 Connect auf eine Host- oder System i-DB2-Datenbank zuzugreifen, können Sie die Fensterskalierung auch auf dem Client aktivieren. Ebenso können Sie die Funktion der Fensterskalierung zwischen einem fernen IBM Data Server-Client und einem Workstation-DB2-Server aktivieren, wenn keine Host- oder System i-DB2-Datenbank beteiligt ist.

Zwar ist die Fensterskalierung dazu gedacht, die Netzleistung zu verbessern, doch ist unbedingt zu beachten, dass sich der erwartete Anstieg der Netzleistung nicht immer einstellt. Das Zusammenwirken verschiedener Faktoren, wie beispielsweise der verwendeten Rahmengröße für den Ethernet- oder Token-Ring-LAN-Adapter, der IP MTU-Größe und anderer Router-Einstellungen im Rahmen der DFV-Verbindung, kann sogar zu einer Beeinträchtigung der Leistung führen, wenn die Fensterskalierung aktiviert wird. Daher ist die Funktion der Fensterskalierung standardmäßig inaktiviert, d. h. die Größe des Sende- und Empfangspuffers ist jeweils auf 64 KB eingestellt.

Die Auswirkungen der aktivierten Fensterskalierung auf die Netzleistung ist im Einzelfall zu beurteilen, und gegebenenfalls müssen entsprechende Anpassungen am Netzwerk vorgenommen werden. Einführende Informationen dazu, wie das Netzwerk zur Verbesserung der Leistung optimiert werden kann, finden Sie unter <http://www.networking.ibm.com/>.

Hostdatenkonvertierung

Wenn Informationen zwischen unterschiedlichen Umgebungen übertragen werden z. B. Intel (Windows), IEEE (Linux und UNIX-Betriebssysteme), zSeries (VM, VSE, z/OS), System i (OS/400), müssen möglicherweise numerische Datentypen (z. B. Dezimalzahlen, Integer und Gleitkommazahlen) umgesetzt werden. Diese Umsetzung kann die Leistung beeinflussen.

Die Datenumsetzung bei Einzelbytezeichen belastet die CPU in der Regel weniger als die Umsetzung numerischer Daten (wenn eine Datenumsetzung erforderlich ist).

Der Aufwand für die Datenumsetzung von Daten des Typs DATE/TIME/TIMESTAMP ist fast so groß wie der für Einzelbytezeichen des Datentyps CHAR. Die Umsetzung von Gleitkommatypen (Datentyp FLOATING) ist am aufwendigsten. Diese Überlegungen sollte ein Anwendungsentwickler berücksichtigen, wenn er eine Anwendung entwirft, die auf DB2 Connect basiert.

Wenn eine Datenbanktabelle eine Spalte hat, für die als Datentyp FOR BIT DATA definiert ist, müssen die zwischen der Anwendung und der Datenbank übertragenen Zeichendaten nicht umgesetzt werden. Dieses Verfahren kann zum Archivieren von Daten auf dem Host- oder System i-Datenbankserver verwendet werden.

Datentypen für Zeichendaten

Zeichendaten können entweder den Datentyp CHAR oder VARCHAR haben. Welcher Datentyp effizienter ist, hängt von der durchschnittlichen Länge der Daten im Feld ab:

- Wenn die Länge der Daten erheblich variiert, ist VARCHAR effizienter, weil CHAR zum Auffüllen des Felds zusätzliche Leerzeichen hinzufügt. Diese Leerzeichen müssen wie alle anderen Zeichen über das Netzwerk übertragen werden.
- Wenn die Größe der Daten nicht stark variiert, ist CHAR effizienter, weil jedes Feld VARCHAR einige Byte an Längenangaben hat, die ebenfalls übertragen werden müssen.

Netzwerkhardware

Folgende Überlegungen gelten für die Hardware:

- Geschwindigkeit des Netzwerks oder der Übertragungsmedien

Die Leistung wird durch ein schnelleres Übertragungsmedium verbessert. Nachfolgend sind einige typische Geschwindigkeiten für die Rohdatenübertragung dargestellt:

Kanal zu Kanal (Lichtwellenleitertechnik)

4,0 MB/s

LAN mit 16 Mbps

2,0 MB/s

Kanal zu Kanal (regulär)

1,0 MB/s

LAN mit 4 Mbps

0,5 MB/s

T1-Hochgeschwindigkeitsträger (1,544 Mbps)

0,193 MB/s

Schnelle ferne Telefonleitung (56 Kbps)

0,007 MB/s

Modem mit 19,6 Kbps

0,002 MB/s

Modem mit 9600 bps

0,001 MB/s

Die Datenübertragungsgeschwindigkeit wird durch das langsamste Übertragungsmedium auf dem Weg zum Host- oder System i-Datenbankserver begrenzt.

- **Netzwerkadapter oder Kommunikationssteuereinheit**
Die Hauptspeicherbelegung durch den Netzwerkadapter und die Kommunikationssteuereinheit sollte sorgfältig geplant werden. Außerdem sollte ein Netzwerkspezialist konsultiert werden, um sicherzustellen, dass die Steuereinheit in der Lage ist, den durch DB2 Connect erzeugten zusätzlichen Datenverkehr zu bewältigen.
- **Netzwerktopologie**
Wenn Daten von LAN zu LAN und von einem Netzwerk zu einem anderen Netzwerk übertragen werden, muss die Übertragungszeit berücksichtigt werden. Brücken, Router und Gateways benötigen zusätzlich Zeit. Beispielsweise wird durch Vermindern der Anzahl überschrittener Brücken die Anzahl von Zwischenschritten verringert, die für jede Anforderung erforderlich ist.
Der physische Abstand zwischen zwei Knoten sollte ebenfalls in Betracht gezogen werden. Selbst wenn eine Nachricht über Satellit übertragen wird, ist die Übertragungsgeschwindigkeit durch die Lichtgeschwindigkeit ($3 \cdot 10^{**8}$ m/s) und die Umlaufdistanz zwischen dem Sender und dem Empfänger begrenzt.
- **Datenaustausch auf dem Netzwerk**
Wenn die Bandbreite des Netzwerks voll ausgenutzt wird, verschlechtern sich die Antwortzeit und die Datenübertragungsgeschwindigkeit für eine einzelne Anwendung.
Wenn sich Daten an einem bestimmten Teil des Netzwerks sammeln (z. B. bei einem alten NCP mit einer sehr geringen Puffergröße), kann eine Überlastung auftreten.
- **Zuverlässigkeit des Netzwerks**
Wenn die Fehlerrate des Netzwerks hoch ist, vermindert sich dessen Durchsatz. Die hierdurch bedingten Neuübertragungen von Daten resultieren in einer Leistungsverschlechterung.

Kapitel 20. Optimierung der CLI/ODBC-Anwendungsleistung

CLI/ODBC ist eine SQL-Anwendungsprogrammierschnittstelle, die von Ihren Datenbankanwendungen aufgerufen werden kann. CLI-Funktionen rufen die gespeicherten Prozeduren von DB2 auf, die ihrerseits auf die Systemkatalogtabellen zugreifen.

Einige Anwendungen verwenden ODBC-APIs zum Erfassen von Metadaten, die bei der weiteren Verarbeitung verwendet werden. Folgende zehn API-Aufrufe für Metadaten sind möglich:

- SQLTables
- SQLColumns
- SQLSpecialcolumns
- SQLStatistics
- SQLPrimarykeys
- SQLForeignkeys
- SQLTablePrivileges
- SQLColumnPrivileges
- SQLProcedures
- SQLProcedureColumns

Bestimmte CLI/ODBC-Anwendungen, die die oben aufgeführten Metadaten-APIs verwenden, fragen alle Objekte in einer Datenbank ab. Der Aufruf `SQLTables` fordert beispielsweise die Metadaten für alle Tabellen in der Datenbank an. Auf einem großen System kann eine solche Anfrage zu erhöhtem Datenverkehr führen, längere Zeit in Anspruch nehmen und einen beträchtlichen Teil der Serverressourcen benötigen.

Es gibt mehrere CLI/ODBC-Initialisierungsschlüsselwörter, mit denen die Datenmenge begrenzt werden kann, die während der Stufe des "Zusammentragens von Informationen" nach dem ersten Herstellen der Verbindung zur Datenbank durch die anfänglichen API-Aufrufe zurückgegeben wird. Diese Schlüsselwörter können durch folgende Methoden festgelegt werden:

1. Manuelles Editieren der Datei `db2cli.ini`
2. Ändern der ODBC/CLI-Einstellungen für die Datenbank mithilfe von 'Client-Konfiguration - Unterstützung' (auf den sie unterstützenden Plattformen)
3. Aktualisieren der Datenbank-CLI-Konfiguration mithilfe der DBA-Befehlszeilenschnittstelle

Die Schlüsselwörter lauten:

- DBName
- TableType
- SchemaList
- SysSchema
- GrantorList
- GranteeList

Teil 5. Fehlerbehebung

Kapitel 21. Fehlerbehebung

Die DB2 Connect-Umgebung umfasst mehrere Software-, Hardware- und Kommunikationsprodukte. Der beste Ansatz für die Fehlerbehebung ist das Ausschließen von Möglichkeiten und eine Annäherung an die Fehlerursache in Einzelschritten.

Stellen Sie zuerst die relevanten Informationen zusammen, und bestimmen Sie auf der Grundlage dieser Informationen das für Ihren Fall zutreffende Thema. Fahren Sie dann mit dem entsprechenden Abschnitt fort.

Zusammenstellen relevanter Informationen

Zur Fehlerbehebung gehört unter anderem das Einkreisen des Problembereichs und das Prüfen möglicher Ursachen. Zu Beginn der Analyse sollten die relevanten Informationen zusammengestellt werden, und es sollte festgestellt werden, welche Fakten bekannt sind, welche Fakten nicht bekannt sind und welche möglichen Problemursachen ausgeschlossen werden können. Es sollten mindestens folgende Fragen beantwortet werden:

- War die einleitende Verbindung erfolgreich?
- Funktioniert die Hardware einwandfrei?
- Funktionieren die Übertragungswege ordnungsgemäß?
- Wurden Änderungen am Kommunikationsnetzwerk vorgenommen, durch die frühere Verzeichniseinträge ungültig wurden?
- Wurde die Datenbank gestartet?
- Tritt der Kommunikationsfehler zwischen mehreren Clients und dem DB2 Connect-Server (Gateway) auf, zwischen dem DB2 Connect-Gateway und dem Host- oder System i-Datenbankserver oder zwischen DB2 Connect Personal Edition und dem Host- oder System i-Datenbankserver?
- Was lässt sich aus dem Inhalt der Nachricht und den in der Nachricht aufgeführten Token ablesen?
- Können Diagnose-Tools wie db2trc, db2pd, und db2support im derzeitigen Stadium hilfreich sein?
- Arbeiten andere Maschinen, die die gleichen Funktionen ausführen, einwandfrei?
- Wird im Fall einer fernen Funktion diese lokal erfolgreich ausgeführt?

Nicht erfolgreiche einleitende Verbindung

Beantworten Sie folgende Fragen, und stellen Sie sicher, dass die Schritte zur Installation ordnungsgemäß durchgeführt wurden:

1. *Wurde der Installationsvorgang erfolgreich abgeschlossen?*
 - Waren alle vorausgesetzten Softwareprodukte verfügbar?
 - War genug Hauptspeicher und Plattenspeicher verfügbar?
 - Wurde die Unterstützung für ferne Clients installiert?
 - Wurde die Installation der Kommunikationssoftware ohne Fehlerbedingungen beendet?
2. *Für UNIX-gestützte Systeme: Wurde eine Instanz des Produkts erstellt?*

- Haben Sie als root-Benutzer einen Benutzer und eine Gruppe als Instanzeigner und als Gruppe SYSADM erstellt?
3. *Wurden (falls dies im vorliegenden Fall zutrifft) die Lizenzinformationen erfolgreich verarbeitet?*
 - Wurde bei UNIX-Systemen die Datei 'nodelock' editiert und das von IBM angegebene Kennwort eingegeben?
 4. *Waren der Host- oder System i-Datenbankserver und die Kommunikation der Workstation ordnungsgemäß konfiguriert?*
 - Es gibt drei Konfigurationen, die betrachtet werden müssen:
 - a. Die Konfiguration des Host- oder System i-Datenbankservers identifiziert den Anwendungsrequester gegenüber dem Server. Das Verwaltungssystem des Host- oder System i-Datenbankservers verfügt über Systemkatalogeinträge, die die Position, das Netzwerkprotokoll und die Sicherheit des Requesters definieren.
 - b. Die Konfiguration der DB2 Connect-Workstation definiert die Gruppe von Clients gegenüber dem Server und den Host- oder System i-Server gegenüber dem Client.
 - c. In der Konfiguration der Client-Workstation müssen der Name der Workstation und das Übertragungsprotokoll definiert sein.
 - Wenn keine einleitende Verbindung hergestellt wurde, muss im Rahmen der Problemanalyse überprüft werden, ob die Namen aller physischen Einheiten vollständig und korrekt sind. Bei TCP/IP-Verbindungen muss überprüft werden, ob die korrekte Portnummer und der korrekte Hostname angegeben wurden.
 - Sowohl der Datenbankadministrator des Host- oder System i-Servers als auch die Netzwerkadministratoren verfügen über Dienstprogramme, mit denen eine Problemdiagnose durchgeführt werden kann.
 5. *Verfügen Sie über die für das Datenbankverwaltungssystem des Host- oder System i-Servers erforderliche Berechtigungsstufe zur Verwendung der Host- oder System i-Serverdatenbank?*
 - Es müssen die Zugriffsberechtigung des Benutzers, die Regeln für Tabellenqualifikationsmerkmale und die erwarteten Ergebnisse beachtet werden.
 6. *Schlägt der Versuch fehl, mit dem Befehlszeilenprozessor (CLP) SQL-Anweisungen für einen Host- oder System i-Datenbankserver abzusetzen?*
 - Wurde der CLP anhand der entsprechenden Prozedur an den Host- oder System i-Datenbankserver gebunden?

Probleme nach dem Herstellen einer einleitenden Verbindung

Die folgenden Fragen sollen helfen, den Problembereich einzukreisen.

1. *Liegen besondere oder ungewöhnlichen Umstände beim Betrieb vor?*
 - Handelt es sich um eine neue Anwendung?
 - Werden neue Prozeduren verwendet?
 - Wurden in letzter Zeit Änderungen vorgenommen, die Auswirkungen auf das System haben könnten? Wurden z. B. Softwareprodukte oder Anwendungen geändert, seit die Anwendung oder das Szenario zum letzten Mal erfolgreich durchgeführt wurden?
 - Bei Anwendungsprogrammen: Welche Anwendungsprogrammierschnittstelle (API) wurde zur Erstellung des Programms verwendet?

- Wurden andere Anwendungen, die die Software oder Kommunikations-APIs verwenden, auf dem System des Benutzers ausgeführt?
 - Wurde in letzter Zeit ein Fixpack installiert? Trat das Problem auf, als ein Benutzer versuchte, eine Funktion zu verwenden, die seit ihrer Installation nicht im Betriebssystem verwendet (oder geladen) wurde, ermitteln Sie das neueste Fixpack von IBM, und laden Sie es, *nachdem* die Funktion installiert wurde.
2. *Trat dieser Fehler bereits früher auf?*
 - Gibt es dokumentierte Lösungen für frühere Fehlerbedingungen?
 - Wer waren die Betroffenen? Können diese Hinweise für mögliche Maßnahmen bieten?
 3. *Wurde versucht, über Befehle der DFV-Software Informationen zum Netzwerk abzurufen?*
 - Durch die Verwendung von TCP/IP-Befehlen und -Dämonen können möglicherweise wertvolle Informationen abgerufen werden.
 4. *Wurden Informationen im SQL-Kommunikationsbereich zurückgegeben, die nützlich sein könnten?*
 - Prozeduren zur Fehlerbehebung sollten Maßnahmen zur Überprüfung des Inhalts der Felder für SQLCODE- und SQLSTATE-Werte umfassen.
 - SQLSTATE-Werte ermöglichen Anwendungsprogrammierern das Testen auf Fehlerklassen, die der DB2-Familie von Datenbankprodukten gemeinsam sind. In einem Netzwerk mit verteilten relationalen Datenbanken kann dieses Feld eine gemeinsame Basis darstellen.
 5. *Wurde DB2START auf dem Server ausgeführt?* Stellen Sie zusätzlich sicher, dass die Umgebungsvariable DB2COMM korrekt eingestellt ist, sodass Clients Fernzugriff auf den Server haben.
 6. *Können andere Maschinen, die die gleichen Funktionen ausführen, die Verbindung zum Server erfolgreich herstellen?* Es könnte sein, dass die maximale Anzahl von Clients erreicht ist, die versuchen, eine Verbindung zum Server herzustellen. Wenn ein anderer Client die Verbindung zum Server trennt, kann der Client, der bisher die Verbindung nicht herstellen konnte, sie nun herstellen?
 7. *Hat die Maschine die richtige Adressierung?* Prüfen Sie, ob die Maschine im Netzwerk eindeutig ist.
 8. *Wurde dem Client die richtige Berechtigung für den Fernzugriff erteilt?* Die Verbindung zur Instanz kann zwar erfolgreich sein, allerdings wurde möglicherweise keine Berechtigung auf Datenbank- oder Tabellenebene erteilt.
 9. *Ist dies die erste Maschine, die eine Verbindung zu einer fernen Datenbank herstellt?* In verteilten Umgebungen können Router oder Brücken zwischen Netzwerken die Kommunikation zwischen dem Client und dem Server blockieren. Bei TCP/IP muss beispielsweise sichergestellt werden, dass der ferne Host auf ein Ping-Signal antwortet.

Diagnosetools

Wenn ein Problem auftritt, stehen folgende Hilfsmittel zur Verfügung:

- Alle Diagnosedaten einschließlich von Speicherauszugsdateien, Trapdateien, Fehlerprotokollen, Benachrichtigungsdateien und Alertprotokollen befinden sich in dem Pfad, der über den Konfigurationsparameter **diagpath** des Datenbankmanagers definiert ist. Dieser Parameter gibt den Pfad für das Verzeichnis mit den Diagnosedaten an.

Ist dieser Konfigurationsparameter mit Null definiert, werden die Diagnosedaten in eines der folgenden Verzeichnisse geschrieben:

- Linux- und UNIX-Umgebungen: `INSTHOME/sqllib/db2dump`, wobei `INSTHOME` das Ausgangsverzeichnis der Instanz angibt.
- Unterstützte Windows-Umgebungen:
 - Ist die Umgebungsvariable **DB2INSTPROF** nicht definiert, wird das Verzeichnis `x:\SQLLIB\DB2INSTANCE` verwendet, wobei `x:\SQLLIB` auf das Laufwerk und das über die Registrierdatenbankvariable **DB2PATH** angegebene Verzeichnis verweist und **DB2INSTANCE** den Namen der Instanz angibt.

Anmerkung: Das Verzeichnis muss nicht zwingend mit `SQLLIB` bezeichnet sein.

- Ist die Umgebungsvariable **DB2INSTPROF** definiert, wird das Verzeichnis `x:\DB2INSTPROF\DB2INSTANCE` verwendet, wobei **DB2INSTPROF** der Name des Instanzprofilverzeichnisses und **DB2INSTANCE** der Name der Instanz (standardmäßig **DB2INSTDEF** unter 32-Bit-Windows-Betriebssystemen) ist.
- Bei Windows-Betriebssystemen können Sie mithilfe der Ereignisanzeigefunktion (Event Viewer) das Protokoll für Verwaltungshinweise anzeigen.
- Als Diagnosetools sind u. a. die Tools **db2trc**, **db2pd** und **db2support** verfügbar.
- Bei Linux- und UNIX-Betriebssystemen der Befehl **ps**, der Informationen zum Status der aktiven Prozesse an die Standardausgabe weiterleitet.
- Bei UNIX-Betriebssystemen die Kerndatei, die im aktuellen Verzeichnis erstellt wird, wenn schwerwiegende Fehler auftreten. Sie enthält ein Hauptspeicherabbild des beendeten Prozesses und kann herangezogen werden, um zu ermitteln, welche Funktion den Fehler verursachte.

Kapitel 22. DB2-Traces in DB2 Connect

Die Traceerstellung für Aktionen und Operationen bei deren Ausführung innerhalb Ihrer Umgebung kann nützliche Informationen für die Fehlerbehebung beim Auftreten eines Problems liefern. Sie können einen Trace, der in einem DB2-Datenbankserverprodukt erstellt wurde, abrufen, formatieren und einen Speicherauszug dafür erstellen. Die Tracefunktion wird als Teil des DB2-Datenbankserverprodukts bereitgestellt.

Abrufen eines DB2-Trace mit 'db2trc'

Der Befehl **db2trc** steuert die mit DB2 zur Verfügung gestellte Tracefunktion. Mit der Tracefunktion können Informationen zu Operationen aufgezeichnet und in ein lesbares Format konvertiert werden.

Es ist zu beachten, dass während der Durchführung eines Trace zusätzlicher Systemaufwand anfällt. Demnach kann die Aktivierung der Tracefunktion die Systemleistung beeinträchtigen.

Im Allgemeinen verwenden die DB2-Unterstützungsfunktion und -Entwicklungsteams DB2-Traces zur Fehlerbehebung. Sie können einen Trace durchführen, um Informationen zu einem Problem zu erhalten, das untersucht wird. Allerdings ist der Nutzen eines solchen Trace ohne Kenntnisse des DB2-Quellcodes relativ begrenzt.

Dennoch sollten Sie mit der korrekten Aktivierung der Tracefunktion vertraut sein und wissen, wie Speicherauszüge für Tracedateien erstellt werden, falls Sie gebeten werden, diese abzurufen.

Anmerkung: Sie benötigen eine der Berechtigungen SYSADM, SYSCTRL oder SYSMAINT, um db2trc zu verwenden.

Führen Sie den Befehl db2trc ohne Parameter aus, um eine Übersicht über die verfügbaren Optionen zu erhalten:

```
C:\>db2trc
Syntax: db2trc (chg|clr|dmp|flw|fmt|inf|off|on) optionen
```

Weitere Informationen zu einem bestimmten Parameter des Befehls db2trc erhalten Sie, wenn Sie die Option -u verwenden. Führen Sie beispielsweise den folgenden Befehl aus, um weitere Informationen zur Aktivierung der Tracefunktion zu erhalten:

```
db2trc on -u
```

Hierdurch werden Informationen zu allen zusätzlichen Optionen (als 'facilities' bezeichnet) aufgerufen, die bei der Aktivierung der DB2-Tracefunktion angegeben werden können.

Beim Einschalten der Tracefunktion ist die wichtigste Option -L. Sie gibt die Größe des Speicherpuffers an, der zum Speichern der Trace-Informationen verwendet wird. Die Puffergröße kann in Byte oder MB angegeben werden. (Für die Angabe von Megabyte hängen Sie an den Wert 'M' oder 'm' an.) Die Tracepuffergröße muss eine Potenz von 2 MB sein. Wenn Sie eine Größe angeben, die diese Voraussetzungen nicht erfüllt, wird die Puffergröße automatisch auf die nächste Potenz von 2

abgerundet. Wenn der Puffer zu klein ist, können Informationen verloren gehen. Standardmäßig werden nur die neuesten Trace-Informationen gespeichert, wenn der Puffer voll ist. Ist der Puffer zu groß, können beim Senden der Datei an das DB2-Unterstützungsteam Probleme auftreten.

Wenn Sie einen Trace für eine Operation erstellen, die relativ kurz ist (wie beispielsweise eine Datenbankverbindung), reicht eine Größe von ca. 8 MB normalerweise aus:

```
C:\> db2trc on -l 8M Trace is turned on
```

Wenn Sie jedoch einen Trace für eine größere Operation erstellen oder wenn viele Aktionen gleichzeitig stattfinden, ist möglicherweise ein größerer Tracepuffer erforderlich.

Auf den meisten Plattformen kann die Tracefunktion zu jedem beliebigen Zeitpunkt aktiviert werden und funktioniert wie oben beschrieben. Bitte beachten Sie jedoch die folgenden Situationen:

1. Auf Systemen mit mehreren Datenbankpartitionen müssen Sie einen Trace für jede physische (im Gegensatz zur logischen) Datenbankpartition ausführen.
2. Wenn auf HP-UX-, Linux- und Solaris-Plattformen die Tracefunktion inaktiviert wird, nachdem die Instanz gestartet wurde, wird unabhängig von der angegebenen Größe beim nächsten Start der Tracefunktion ein sehr kleiner Puffer verwendet. Ein Beispiel: Gestern haben Sie die Tracefunktion mit `db2trc on -l 8m` eingeschaltet, einen Trace erfasst und anschließend die Tracefunktion ausgeschaltet (`db2trc off`). Heute möchten Sie einen Trace mit einem Hauptspeicherpuffer von 32 MB durchführen (`db2trc on -l 32m`), ohne die Instanz herunterzufahren und neu zu starten. In diesem Fall wird der Trace nur mit einem kleinen Puffer durchgeführt werden. Um einen Trace auf diesen Plattformen effektiv durchzuführen, müssen Sie die Tracefunktion mit der benötigten Puffergröße einschalten, bevor Sie die Instanz starten und den Inhalt des Puffers gegebenenfalls zu einem späteren Zeitpunkt „löschen“.

Erstellen eines Speicherauszugs einer DB2-Tracedatei

Wenn die Tracefunktion mit der Option 'on' aktiviert wurde, wird für die gesamte nachfolgende Arbeit, die von dieser Instanz ausgeführt wird, ein Trace erstellt.

Während die Tracefunktion aktiv ist, können Sie mit der Option 'clr' den Tracepuffer bereinigen. Alle im Tracepuffer vorhandenen Informationen werden entfernt.

```
C:\>db2trc clr  
Trace has been cleared
```

Wenn die Operation, für die ein Trace erstellt wird, abgeschlossen ist, können Sie die Option 'dmp' gefolgt von einem Tracedateinamen verwenden, um für den Speicherpuffer einen Speicherauszug auf der Platte zu erstellen. Beispiel:

```
C:\>db2trc dmp trace.dmp  
Trace has been dumped to file
```

Nachdem für den Tracepuffer ein Speicherauszug auf der Platte erstellt wurde, ist die Tracefunktion weiterhin aktiv. Verwenden Sie die Option 'off', um die Tracefunktion zu inaktivieren:

```
C:\>db2trc off  
Trace is turned off
```

Formatieren einer DB2-Tracedatei

Die mit dem Befehl `db2trc dmp` erstellte Speicherauszugsdatei liegt in Binärformat vor und ist nicht lesbar.

Prüfen sie, ob eine Tracedatei gelesen werden kann, indem Sie die binäre Tracedatei formatieren, damit die Fluss-Steuerung angezeigt wird, und senden Sie die formatierte Ausgabe an eine Nulleinheit. Das folgende Beispiel zeigt den Befehl für die Ausführung dieser Task:

```
db2trc flw example.trc nul
```

Hierbei ist `example.trc` eine Binärdatei, die mit der Option `dmp` erstellt wurde.

Die Ausgabe dieses Befehls gibt Ihnen explizit an, wenn ein Problem beim Lesen der Datei besteht oder ob beim Trace ein Umlauf stattgefunden hat.

Zu diesem Zeitpunkt kann die Speicherauszugsdatei an die DB2-Unterstützungsfunktion gesendet werden. Dort wird sie auf der Basis der verwendeten DB2-Servicestufe formatiert. In manchen Fällen werden Sie jedoch dazu aufgefordert, die Speicherauszugsdatei in ASCII-Format zu konvertieren, bevor Sie sie senden. Hierzu verwenden Sie die Optionen `flw` und `fmt`. Sie müssen den Namen der binären Speicherauszugsdatei und den Namen der zu erstellenden ASCII-Datei angeben:

```
C:\>db2trc flw trace.dmp trace.flw
C:\Temp>db2trc flw trace.dmp trace.flw
Total number of trace records      : 18854
Trace truncated                   : NO
Trace wrapped                     : NO
Number of trace records formatted : 1513 (pid: 2196 tid 2148 node: -1)
Number of trace records formatted : 100 (pid: 1568 tid 1304 node: 0)
...
```

```
C:\>db2trc fmt trace.dmp trace.fmt
C:\Temp>db2trc fmt trace.dmp trace.fmt
Trace truncated                   : NO
Trace wrapped                     : NO
Total number of trace records      : 18854
Number of trace records formatted : 18854
```

Wenn diese Ausgabe für "Trace wrapped" die Angabe "YES" enthält, bedeutet dies, dass der Tracepuffer nicht ausreichte, um alle während des Tracezeitraums erfassten Daten aufzuzeichnen. Abhängig von der Situation kann ein solcher Trace akzeptabel sein. Wenn Sie die neuesten Informationen benötigen (dies sind die Informationen, die standardmäßig aufbewahrt werden, es sei denn, die Option `-i` wird angegeben), ist der Inhalt der Tracedatei möglicherweise ausreichend. Wenn Sie jedoch Informationen zu den Ereignissen zu Beginn des Tracezeitraums oder Informationen zu allen Ereignissen benötigen, sollten Sie die Operation mit einem größeren Tracepuffer wiederholen.

Bei der Formatierung einer Binärdatei in eine lesbare Textdatei stehen Ihnen Optionen zur Verfügung. Sie können zum Beispiel `db2trc fmt -xml trace.dmp trace.fmt` verwenden, um die Binärdaten zu konvertieren und das Ergebnis in einem Format auszugeben, das durch `xml` geparkt werden kann. Weitere Optionen werden in der ausführlichen Beschreibung der Tracebefehle (`db2trc`) dargestellt.

Darüber hinaus ist zu beachten, dass DB2 unter Linux- und UNIX-Betriebssystemen automatisch einen Speicherauszug des Tracepuffers auf der Platte erstellt, wenn es die Instanz auf Grund eines schwer wiegenden Fehlers beendet. Wenn also bei der abnormalen Beendigung einer Instanz die Tracefunktion aktiviert ist, wird eine Datei im Diagnoseverzeichnis erstellt, die den Namen db2trdmp.### hat, wobei ### die Nummer der Datenbankpartition angibt. Auf Windows-Plattformen findet dies nicht statt. In diesen Fällen muss ein Speicherauszug für den Trace manuell erstellt werden.

Abschließend folgt als Zusammenfassung ein Beispiel der üblichen Folge von db2trc-Befehlen:

```
db2trc on -l 8M
db2trc clr
<Befehle zur Fehlerreproduktion ausführen>
db2trc dump db2trc.dmp
db2trc off
db2trc flw db2trc.dmp <dateiname>.flw
db2trc fmt db2trc.dmp <dateiname>.fmt
db2trc fmt -c db2trc.dmp <dateiname>.fmtc
```

Kapitel 23. DRDA-Tracedateien

Vor der Analyse von DRDA-Traces sollten Sie sich darüber im Klaren sein, dass es sich bei DRDA um einen offenen Standard für die Definition von Daten- und Kommunikationsstrukturen handelt. So umfasst DRDA beispielsweise eine Reihe von Regeln zur Strukturierung von Daten für die Übertragung sowie zur Durchführung dieser Datenübertragung. Diese Regeln sind in den folgenden Referenzhandbüchern definiert:

- DRDA V3 Vol. 1: Distributed Relational Database Architecture
- DRDA V3 Vol. 2: Formatted Data Object Content Architecture
- DRDA V3 Vol. 3: Distributed Data Management Architecture

PDF-Versionen dieser Handbücher sind unter www.opengroup.org verfügbar.

Das Dienstprogramm **db2drdat** zeichnet den Datenaustausch zwischen einem DRDA-Anwendungsrequester (AR) und einem DB2 DRDA-Anwendungsserver (AS) auf (zum Beispiel zwischen DB2 Connect und einem Host oder einem System i-Datenbankserver).

Dienstprogramm für Trace

Das Dienstprogramm **db2drdat** zeichnet die Daten auf, die zwischen dem DB2 Connect-Server (für den IBM Data Server-Client) und dem Host- oder System i-Datenbankserver ausgetauscht werden.

Die Kenntnis dieses Datenstroms ist für den Datenbankadministrator und den Anwendungsentwickler oft sehr hilfreich, da anhand dieses Wissens die Ursachen bestimmter Fehler gefunden werden können. Nehmen Sie die folgende Situation als Beispiel: Eine Datenbankanweisung **CONNECT T0** für einen Host- oder System i-Datenbankserver wird abgesetzt, der Befehl schlägt jedoch fehl, und Sie erhalten einen Rückkehrcode, der auf einen Fehler hinweist. Wenn genau bekannt ist, welche Informationen an das Verwaltungssystem des Host- oder System i-Datenbankservers übertragen wurden, kann die Fehlerursache auch dann ermittelt werden, wenn die Informationen des Rückkehrcodes allgemein sind. Häufig schlägt ein Befehl aufgrund eines einfachen Benutzerfehlers fehl.

In der Ausgabe von **db2drdat** werden die zwischen der DB2 Connect-Workstation und dem Verwaltungssystem des Host- oder System i-Datenbankservers ausgetauschten Datenströme aufgelistet. An den Host- oder System i-Datenbankserver übertragene Daten werden unter **SEND BUFFER** (Sendepuffer), vom Host- oder System i-Datenbankserver empfangene Daten unter **RECEIVE BUFFER** (Empfangspuffer) aufgeführt.

Wenn ein Empfangspuffer Informationen zum SQL-Kommunikationsbereich enthält, folgt auf diese eine formatierte Interpretation dieser Daten unter der Bezeichnung **SQLCA**. Das **SQLCODE**-Feld eines SQL-Kommunikationsbereichs ist der *nicht zugeordnete* Wert, so wie er vom Host- oder System i-Datenbankserver zurückgegeben wurde. Die Sendepuffer und Empfangspuffer sind von den ältesten zu den neuesten innerhalb der Datei sortiert. Jeder Puffer verfügt über folgende Angaben:

- Die Prozess-ID

- Eine Bezeichnung SEND BUFFER, RECEIVE BUFFER oder SQLCA. Der erste DDM-Befehl oder das erste DDM-Objekt in einem Puffer wird als DSS TYPE bezeichnet.

Die weiteren Daten in Sende- und Empfangspuffern werden in den folgenden fünf Spalten dargestellt:

- Die Byteanzahl
- Spalte 2 und 3 stellen den zwischen den beiden Systemen ausgetauschten DRDA-Datenstrom in ASCII oder EBCDIC dar.
- Eine ASCII-Darstellung der Spalten 2 und 3
- Eine EBCDIC-Darstellung der Spalten 2 und 3

Traceausgabe

Das Dienstprogramm db2drdat schreibt die folgenden Informationen in die *Tracedatei*:

- -r
 - Art der/des DRDA-Antwort/Objekts
 - Empfangspuffer
- -s
 - Art der DRDA-Anforderung
 - Sendepuffer
- -c
 - SQLCA
- TCP/IP-Fehlerinformationen
 - Rückkehrcode der Empfangsfunktion
 - Bewertung
 - Verwendetes Protokoll
 - Verwendete API
 - Funktion
 - Fehlernummer

Anmerkung:

1. Der Wert null für den Endecode zeigt an, dass der Befehl erfolgreich ausgeführt wurde. Ein Wert ungleich null zeigt an, dass der Befehl nicht erfolgreich ausgeführt wurde.
2. Die zurückgegebenen Felder hängen von der verwendeten API ab.
3. Welche Felder zurückgegeben werden, hängt von der Plattform ab, auf der DB2 Connect ausgeführt wird. Es können daher für dieselbe API unterschiedliche Felder zurückgegeben werden.
4. Wenn der Befehl db2drdat die Ausgabe an eine bereits existierende Datei leitet, wird die alte Datei gelöscht, sofern die Berechtigungen für die Datei dies zulassen.

Analyse der Traceausgabedatei

Die folgenden Informationen werden bei einem db2drdat-Trace erfasst:

- Die Prozess-ID (PID) der Client-Anwendung
- Der RDB_NAME, der im DCS-Verzeichnis katalogisiert ist
- Die ID(s) für den codierten Zeichensatz von DB2 Connect
- Die IDs für den codierten Zeichensatz des Host- oder System i-Datenbankservers
- Das Verwaltungssystem des Host- oder System i-Datenbankservers, mit dem das DB2 Connect-System kommuniziert

Der erste Puffer enthält die Befehle EXCSAT (Exchange Server Attributes) und ACCRDB (Access RDB), die an das Verwaltungssystem des Host- oder System i-Datenbankservers gesendet werden. Diese Befehle werden als Ergebnis des Datenbankbefehls CONNECT TO gesendet. Der nächste Puffer enthält die Antwort, die DB2 Connect vom Verwaltungssystem des Host- oder System i-Datenbankservers empfing. Sie enthält EXCSATRD (Exchange Server Attributes Reply Data) und eine ACCRDBRM (Access RDB Reply Message).

EXCSAT

Der Befehl EXCSAT enthält den Workstationnamen des Clients, der vom Objekt SRVNAM (Servername) angegeben wird; dieser Name entspricht gemäß der DDM-Spezifikation dem Codepunkt X'116D'. Der Befehl EXCSAT befindet sich im ersten Puffer. Im Befehl EXCSAT werden die Werte X'9481A292' (die in CCSID 500 codiert sind) in *mask* umgesetzt, sobald X'116D' entfernt ist.

Der Befehl EXCSAT enthält außerdem das Objekt EXTNAM (externer Name), das oft in Diagnoseinformationen im Host- oder System i-Datenbankverwaltungssystem zu finden ist. Es besteht aus einer 20 Byte langen Anwendungs-ID, gefolgt von einer 8 Byte langen Prozess-ID (oder einer 4 Byte langen Prozess-ID und einer 4 Byte langen Thread-ID). Es wird durch den Codepunkt X'115E' dargestellt und hat in diesem Beispiel den Wert db2bp, der durch Leerzeichen aufgefüllt ist und an den sich 000C50CC anschließt. Auf einem Linux- oder UNIX-IBM Data Server-Client kann dieser Wert mit dem Befehl ps korreliert werden, der Prozess-Statusinformationen zu aktiven Prozessen an die Standardausgabe übergibt.

ACCRDB

Der Befehl ACCRDB enthält den RDB_NAME im Objekt RDBNAM (Codepunkt X'2110'). Der Befehl ACCRDB folgt auf den Befehl EXCSAT im ersten Puffer. Im Befehl ACCRDB werden die Werte X'E2E3D3C5C3F1' in STLEC1 umgesetzt, sobald X'2110' entfernt ist. Dies entspricht dem Feld für den Zieldatenbanknamen im DCS-Verzeichnis.

Die Abrechnungszeichenfolge hat den Codepunkt X'2104'.

Der codierte Zeichensatz für die DB2 Connect-Workstation kann ermittelt werden, indem das CCSID-Objekt CCSIDSBC (CCSID für Einzelbytezeichen) mit Codepunkt X'119C' im ACCRDB angegeben wird. In diesem Beispiel ist der Wert für CCSIDSBC X'0333', d. h. 819.

Die zusätzlichen Objekte CCSIDDBC (CCSID für Doppelbytezeichen) und CCSIDMBC (CCSID für Mischbytezeichen) mit dem Codepunkt X'119D' und X'119E' sind im Befehl ACCRDB ebenfalls vorhanden. In diesem Beispiel ist der Wert für CCSIDDBC X'04B0', d. h. 1200. Der Wert für CCSIDMBC ist X'0333', d. h. 819.

EXCSATRD und ACCRDBRM

CCSID-Werte werden auch vom Host- oder System i-Datenbankserver in ACCRDBRM (Access RDB Reply Message) im zweiten Puffer zurückgegeben. Dieser Puffer enthält die EXCSATRD-Daten, gefolgt von den ACCRDBRM-Daten. Die Beispielausgabedatei enthält zwei CCSIDs für das Host- oder System i-Datenbankserverssystem. Die Werte sind 1208 (für Einzelbyte- und Mischbytezeichen) bzw. 1200 (für Doppelbytezeichen).

Wenn DB2 Connect die Codepage, die vom Host- oder System i-Datenbankserver zurückgegeben wird, nicht erkennt, wird SQLCODE -332 mit der Quellen- und Zielcodepage an den Benutzer ausgegeben. Wenn der Host- oder System i-Datenbankserver den von DB2 Connect gesendeten codierten Zeichensatz nicht erkennt, gibt er VALNSPRM (Parameterwert nicht unterstützt, DDM-Codepunkt X'1252') zurück; diese Angaben werden für den Benutzer in SQLCODE -332 umgesetzt.

Der Befehl ACCRDBRM enthält auch den Parameter PRDID (produktspezifische Kennung, Codepunkt X'112E'). Der Wert ist X'C4E2D5F0F8F0F1F5', d. h. DSN08015 in EBCDIC. Gemäß den Standards entspricht DSN der Angabe DB2 Universal Database für z/OS und OS/390. Die Versionsnummer ist ebenfalls angegeben. ARI ist DB2 Server für VSE & VM, SQL ist DB2 Database oder DB2 Connect, und QSQ ist DB2 für i5/OS.

Beispiele für die Traceausgabedatei

Die folgenden Abbildungen zeigen ein Beispiel einer Ausgabe, das einige DRDA-Datenströme darstellt, die zwischen DB2 Connect-Workstations und einem Host- oder System i-Datenbankserver ausgetauscht werden. Vom Standpunkt des Benutzers wurde über den Befehlszeilenprozessor (CLP) ein Befehl CONNECT T0 für eine Datenbank abgesetzt.

Abb. 13 auf Seite 151 verwendet DB2 Connect Enterprise Edition Version 9.1 und DB2 Universal Database (UDB) für z/OS Version 8 über eine TCP/IP-Verbindung.

1 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 0 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 233

2 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 19532 probe 1177
 bytes 250

SEND BUFFER(AR):

EXCSAT RQSDSS		(ASCII)	(EBCDIC)																	
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	0123456789ABCDEF			
0000	00C3D041000100BD	1041007F115E8482	...	A	...	A	...	^	..	.	C	}	;	db				
0010	F282974040404040	4040404040404040	...	@	@	@	@	@	@	@		2bp	
0020	4040F0F0F0C3F5F0	C3C3F0F0F0000000	@@															000C50CC000...	
0030	0000000000000000	0000000000000000	
0040	0000000000000000	000000000060F0F0-00	
0050	F0F1A2A495404040	4040404040404040	@	@	@	@	@	@	@		01sun	
0060	4040404040404040	4040404040404040	@	@	@	@	@	@	@	@		
0070	C4C5C3E5F8404040	F0A2A49540404040	@	@	@	@	@	@	@		DECV8 0sun	
0080	4040404040404040	4000181404140300	@	@	@	@	@	@	@	@
0090	0724070008147400	05240F0008144000	..	\$...	t	...	\$...	@
00A0	08000E1147D8C4C2	F261C1C9E7F6F400	...	G	...	aQDB2/AIX64.
00B0	08116D9481A29200	0C115AE2D8D3F0F9	..	m	Z_mask...]SQL09
00C0	F0F0F0		...																	000

ACCSEC RQSDSS		(ASCII)	(EBCDIC)																	
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	0123456789ABCDEF			
0000	0026D00100020020	106D000611A20003	..	&	m}.....s..
0010	00162110E2E3D3C5	C3F1404040404040	..	!	@	@	@	@	@	@STLEC1	
0020	40404040404040		@	@	@	@	@	@	@	@		

3 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110546200 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 105

4 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110549755 probe 1178
 bytes 122

RECEIVE BUFFER(AR):

EXCSATRD OBJDSS		(ASCII)	(EBCDIC)																	
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	0123456789ABCDEF			
0000	0059D04300010053	1443000F115EE5F8	..	Y	...	S	...	C	...	^	..	}	;	V8				
0010	F1C14BE2E3D3C5C3	F100181404140300	..	K															1A.STLEC1.....
0020	0724070007147400	05240F0007144000	..	\$...	t	...	\$...	@
0030	0700081147D8C4C2	F20014116DE2E3D3	...	G	mQDB2..._STL
0040	C5C3F14040404040	4040404040000C11	...	@	@	@	@	@	@	@		EC1 ...	
0050	5AC4E2D5F0F8F0F1	F5	Z]DSN08015

ACCSECRD OBJDSS		(ASCII)	(EBCDIC)																	
0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	0123456789ABCDEF	0123456789ABCDEF			
0000	0010D0030002000A	14AC000611A20003}.....s..

5 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110656806 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 233

Abbildung 13. Beispiel einer Traceausgabe (TCP/IP-Verbindung)

6 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110659711 probe 1177
 bytes 250

SEND BUFFER(AR):

	SECCHK RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	003CD04100010036	106E000611A20003	..<.A...6.n.....
0010	00162110E2E3D3C5	C3F1404040404040	..!.....@@@
0020	40404040404000C	11A1D9858799F485	@@@.....
0030	A599000A11A09585	A6A39695vr....newton

	ACCRDB RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	00ADD001000200A7	20010006210F2407!.\$.
0010	00172135C7F9F1C1	F0C4F3C14BD7C1F8	..!5.....K...
0020	F806030221064600	162110E2E3D3C5C3!F.!.....
0030	F140404040404040	4040404040000C111
0040	2EE2D8D3F0F9F0F0	F000D002FD8E3C4/...SQL0900....QTD
0050	E2D8D3C1E2C30016	00350006119C03335.....3 SQLASC.....
0060	0006119D04B00006	119E0333003C21043.

7 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259908001 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 176

8 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259911584 probe 1178
 bytes 193

RECEIVE BUFFER(AR):

	SECCHKRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0015D0420001000F	1219000611490000	...B.....I..
0010	000511A400	u.

	ACCRDBRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	009BD00200020095	2201000611490000"....I..
0010	000D002FD8E3C4E2	D8D3F3F7F0000C11	.../.....
0020	2EC4E2D5F0F8F0F1	F5001600350006115....
0030	9C04B80006119E04	B80006119D04B000
0040	0C11A0D5C5E6E3D6	D540400006212524@...!%\$
0050	34001E244E000624	4C00010014244D00	4..\$N..\$L...\$M.
0060	06244FFFFF000A11	E8091E768301BE00	.\$0.....v....
0070	2221030000000005	68B3B8C7F9F1C1F0	"!.....h.....
0080	C4F3C1D7C1F8F840	4040400603022106@@@...!
0090	46000A11E8091E76	831389	F.....v....

9 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364420503 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 10

Abbildung 14. Beispiel einer Trace-Ausgabe (TCP/IP-Verbindung) - Fortsetzung

10 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364440751 probe 1177
 bytes 27

SEND BUFFER(AR):

	RDBCMM RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000AD00100010004 200E}.....

11 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475009631 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 54

12 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475014579 probe 1178
 bytes 71

RECEIVE BUFFER(AR):

	ENDUOWRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD05200010025 220C000611490004	..+.R...%"....I..	..}.....
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@#@#@#@STLEC1
0020	4040404040400005 211501	@#@#@@...!..

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000BD00300010005 2408FF\$..	..}.....

13 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721710319 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 126

14 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721727276 probe 1177
 bytes 143

SEND BUFFER(AR):

	EXCSQLIMM RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0053D0510001004D 200A00442113E2E3	..S.Q...M ..D!...	..}....(.....ST
0010	D3C5C3F140404040 4040404040404040@#@#@#@#@#@	LEC1
0020	D5E4D3D3C9C44040 4040404040404040@#@#@#@#@	NULLID
0030	4040E2D8D3C3F2C6 F0C1404040404040	@@.....@#@#@#@	SQLC2F0A
0040	4040404041414141 41484C5600CB0005	@#@@AAAAAHLV....<.....
0050	2105F1	!..	..1

	SQLSTT OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD00300010025 2414000000001B64	..+....%\$......d	..}.....
0010	656C657465206672 6F6D206464637375	elete from ddcsu	..%.....?_.....
0020	73312E6D79746162 6C65FF	s1.mytable.	..._`./.%..

15 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832901261 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 102

Abbildung 15. Beispiel einer Trace-Ausgabe (TCP/IP-Verbindung) - Fortsetzung

16 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832906528 probe 1178
 bytes 119

RECEIVE BUFFER(AR):

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0066D00300010060 240800FFFFFF3434	.f.....`\$.44	..}....-.....
0010	3237303444534E58 4F544C2000FFFFFFE	2704DSNXOTL+!.<.....
0020	0C00000000000000 00FFFFFFF000000
0030	00000000000572020 2057202020202020W W
0040	001053544C454331 2020202020202020	..STLEC1<.....
0050	2020000F44444353 5553312E4D595441	..DDCSUS1.MYTA(...
0060	424C450000FF	BLE...<.....

17 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833156953 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 10

18 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833159843 probe 1177
 bytes 27

SEND BUFFER(AR):

	RDBRLLBCK RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000AD00100010004 200F}.....

19 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943302832 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 54

20 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943306288 probe 1178
 bytes 71

RECEIVE BUFFER(AR):

	ENDUOWRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD05200010025 220C000611490004	..+R...%"....I..	..}.....
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@#@#@@STLEC1
0020	4040404040400005 211502	@#@#@@...!..

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000BD00300010005 2408FF\$..	..}.....

Abbildung 16. Beispiel einer Trace-Ausgabe (TCP/IP-Verbindung) - Fortsetzung

Informationen zu nachfolgenden Puffern für DRDA-Traces

Auch die nachfolgenden Sendepuffer und Empfangspuffer können auf zusätzliche Informationen hin analysiert werden. Die nächste Anforderung enthält einen Befehl COMMIT. Der Befehl commit weist das Verwaltungssystem des Host- oder System i-Datenbankservers an, die aktuelle UOW festzuschreiben. Der vierte Puffer wird vom Datenbankverwaltungssystem des Host- oder System i-Datenbankservers als Ergebnis eines Commits oder eines Rollbacks empfangen. Er enthält die ENDUOWRM-Nachricht (End Unit of Work Reply Message), die anzeigt, dass die aktuelle UOW beendet wurde.

In diesem Beispiel enthält der Trace-Eintrag 12 einen leeren SQLCA, angegeben durch den DDM-Codepunkt X'2408' gefolgt von X'FF'. Ein leerer SQL-Kommunikationsbereich (X'2408FF') zeigt die erfolgreiche Ausführung an (SQLCODE 0).

Abb. 13 auf Seite 151 zeigt ein Beispiel eines Empfangspuffers mit einem Fehler-SQLCA bei Trace-Eintrag 16.

Teil 6. Nachrichten

Kapitel 24. DB2 Connect - Häufige Probleme

In diesem Abschnitt werden die häufigsten Symptome von Verbindungsproblemen bei der Verwendung von DB2 Connect aufgelistet. Für jedes Problem werden Ihnen folgende Informationen zur Verfügung gestellt:

- Eine Kombination aus Nachrichtennummer und Rückkehrcode (oder protokollspezifischem Rückkehrcode) für die Nachricht. Jede Kombination aus Nachricht und Rückkehrcode hat eine separate Überschrift, und die Überschriften sind der Nachrichtennummer und dann dem Rückkehrcode nach geordnet.
- Es wird ein Symptom angegeben, in der Regel in Form einer Beispielnachricht.
- Es wird eine Lösung vorgeschlagen, die die wahrscheinliche Ursache des Fehlers angibt. In einigen Fällen werden eventuell mehrere Lösungen vorgeschlagen.

SQL0965 oder SQL0969

Symptom

Die Nachrichten SQL0965 und SQL0969 können mit einer Reihe unterschiedlicher Rückkehrcodes von DB2 für i5/OS, DB2 für z/OS und DB2 für VM & VSE ausgegeben werden.

Wird eine dieser Nachrichten angezeigt, müssen Sie den ursprünglichen SQL-Code in der Dokumentation für das Datenbankserverprodukt nachschlagen, das die Nachricht ausgegeben hat.

Lösung

Der von der Host- oder i5/OS-Datenbank empfangene SQL-Code kann nicht umgesetzt werden. Korrigieren Sie das Problem basierend auf dem Fehlercode, und wiederholen Sie den fehlgeschlagenen Befehl.

SQL5043N

Symptom

Die Unterstützung für eines oder mehrere Übertragungsprotokolle konnte nicht gestartet werden. Die Kernfunktionalität des Datenbankmanagers wurde jedoch erfolgreich gestartet.

Vielleicht wurde das TCP/IP-Protokoll auf dem DB2 Connect-Server nicht gestartet. Möglicherweise hat zuvor eine erfolgreiche Client-Verbindung bestanden.

Bei Verwendung von `diaglevel = 4` enthält `db2diag.log` eventuell einen Eintrag wie zum Beispiel den folgenden:

```
2001-05-30-14.09.55.321092 Instance:svtdbm5 Node:000
PID:10296(db2tcpm) Appid:none
common_communication sqlcctcpconnmgr_child Probe:46
DIA3205E Die Socket-Adresse "30090", die in der
TCP/IP-Servicedatei definiert und für die TCP/IP-Server-
Unterstützung erforderlich ist, wird von einem anderen
Prozess verwendet.
```

Lösung

Diese Warnung ist ein Symptom dafür, dass DB2 Connect als Server für ferne Clients Schwierigkeiten beim Verarbeiten von einem oder mehreren Clientübertragungsprotokollen hat. Dabei kann es sich um TCP/IP-Protokolle oder andere Protokolle handeln. In der Nachricht wird in der Regel

angegeben, dass eines der für DB2 Connect definierten Übertragungsprotokolle nicht ordnungsgemäß konfiguriert ist.

Eine mögliche Ursache ist häufig, dass die Profilvariable DB2COMM nicht oder falsch definiert ist. Im Allgemeinen ist das Problem das Ergebnis einer Abweichung zwischen der Variablen DB2COMM und den in der Datenbankmanagerkonfiguration definierten Namen (zum Beispiel svcname oder nname).

Ein mögliches Szenario ist, dass zuvor erfolgreich eine Verbindung hergestellt wurde und dass dann die Fehlernachricht SQL5043 angezeigt wird, obwohl die Konfiguration nicht geändert wurde. Dazu kann es bei Verwendung des TCP/IP-Protokolls kommen, wenn das ferne System die Verbindung aus einem bestimmten Grund abnormal beendet. Tritt dieser Fall auf, kann auf dem Client weiterhin eine Verbindung vorhanden sein, und es ist eventuell möglich, die Verbindung durch Absetzen der unten stehenden Befehle ohne weiteres Eingreifen wiederherzustellen.

Sehr wahrscheinlich hat einer der Clients, der mit dem DB2 Connect-Server verbunden ist, noch eine Kennung am TCP/IP-Port. Geben Sie auf jeder Client-Maschine, die mit dem DB2 Connect-Server verbunden ist, die folgenden Befehle ein:

```
db2 terminate
db2stop
```

SQL30020

Symptom

SQL30020N Die Ausführung schlug aufgrund eines Verteilungsprotokollfehlers (Distributed Protocol Error) fehl. Dieser Fehler beeinflusst die erfolgreiche Ausführung der nachfolgenden Befehle und SQL-Anweisungen.

Lösungen

Wenden Sie sich bei diesem Fehler an den Service. Führen Sie den Befehl db2support aus, bevor Sie sich an den Service wenden.

SQL30060

Symptom

SQL30060N "<berechtigungs-id>" verfügt nicht über die Berechtigung, die Operation "<operationn>" auszuführen.

Lösung

Beim Herstellen der Verbindung zu DB2 für OS/390 und z/OS wurden die Kommunikationsdatenbanktabellen nicht ordnungsgemäß aktualisiert.

SQL30061

Symptom

Verbindung zu falschem Datenbankserverstandort auf dem Host- oder System i-System; keine Zieldatenbank gefunden.

Lösung

Im DCS-Verzeichniseintrag wurde eventuell der falsche Name der Serverdatenbank angegeben. In diesem Fall wird SQLCODE -30061 an die Anwendung zurückgegeben.

Überprüfen Sie den DB2-Knoten, die Datenbank und die DCS-Verzeichniseinträge. Das Feld für den Zieldatenbanknamen im DCS-Verzeichniseintrag muss mit dem Namen der Datenbank auf der Basis der Plattform überein-

stimmen. Bei einer Datenbank unter DB2 Universal Database für z/OS und OS/390 muss der zu verwendende Name beispielsweise mit dem Namen übereinstimmen, der im Feld "LOCATION=*standortname*" des BSDS (Boot Strap Data Set) verwendet wird und der auch in der Nachricht DSNL004I (LOCATION=*standort*) angezeigt wird, wenn DDF (Distributed Data Facility) gestartet wird.

Die korrekten Befehle für einen TCP/IP-Knoten lauten wie folgt:

```
db2 catalog tcpip node <knotenname> remote <hostname_oder_-adresse>
server <port_nr_oder_servicename>
db2 catalog dcs database <lokaler_name> as <tatsächlicher_datenbankname>
db2 catalog database <lokaler_name> as <alias> at node <knotenname>
authentication server
```

Setzen Sie anschließend den folgenden Befehl ab, um die Verbindung zur Datenbank herzustellen:

```
db2 connect to <alias> user <benutzername> using <kennwort>
```

SQL30081N mit Rückkehrcode 79

Symptom

```
SQL30081N Übertragungsfehler.
  Verwendetes Übertragungsprotokoll: "TCP/IP".
  Verwendete Übertragungs-API: "SOCKETS".
  Position, an der der Fehler festgestellt wurde: "".
  Übertragungsfunktion, die den Fehler feststellte: "connect".
  Protokollspezifische(r) Fehlercode(s): "79", "*", "*".
  SQLSTATE=08001
```

Lösung(en)

Dieser Fehler kann auftreten, wenn ein ferner Client keine Verbindung zu einem DB2 Connect-Server herstellen kann. Er kann auch auftreten, wenn eine Verbindung vom DB2 Connect-Server zu einem Host- oder System i-Datenbankserver hergestellt wird.

1. Möglicherweise ist die Profilvariable DB2COMM auf dem DB2 Connect-Server falsch eingestellt. Überprüfen Sie diese Variable. Zum Beispiel muss der Befehl `db2set db2comm=tcpip in sqllib/db2profile` vorhanden sein, wenn DB2 Enterprise Server Edition unter AIX ausgeführt wird.
2. Möglicherweise liegt eine Abweichung zwischen dem TCP/IP-Service-namen und den Angaben der Portnummern auf dem IBM Data Server-Client und dem DB2 Connect-Server vor. Prüfen Sie die Einträge in den TCP/IP-Dateien `services` auf beiden Maschinen.
3. Stellen Sie sicher, dass DB2 auf dem DB2 Connect-Server gestartet wurde. Setzen Sie `diaglevel` der Datenbankmanagerkonfiguration mit dem folgenden Befehl auf 4:

```
db2 update dbm cfg using diaglevel 4
```

Überprüfen Sie nach dem Stoppen und Neustart von DB2 in der Datei `db2diag.log`, ob die DB2-TCP/IP-Kommunikation gestartet wurde. Es wird eine Ausgabe angezeigt, die der folgenden ähnelt:

```
2001-02-03-12.41.04.861119 Instance:svtdbm2 Node:00
PID:86496(db2sysc) Appid:none
common_communication sqlcctcp_start_listen Probe:80
DIA3000I Die Protokollunterstützung für "TCPIP" wurde erfolgreich
gestartet.
```

SQL30081N mit protokollspezifischem Fehlercode 10032

Symptom

SQL30081N Übertragungsfehler.
Verwendetes Übertragungsprotokoll: "TCP/IP".
Verwendete Übertragungs-API: "SOCKETS".
Position, an der der Fehler festgestellt wurde: "9.21.85.159".
Übertragungsfunktion, die den Fehler feststellte: "send".
Protokollspezifische(r) Fehlercode(s): "10032", "*", "*".
SQLSTATE=08001

Lösung

Diese Fehlernachricht wird eventuell empfangen, wenn versucht wird, die Verbindung zu einer Maschine zu trennen, auf der die TCP/IP-Kommunikation bereits fehlgeschlagen ist. Korrigieren Sie das Problem mit dem TCP/IP-Subsystem.

Starten Sie dazu auf den meisten Maschinen einfach das TCP/IP-Protokoll erneut. Gelegentlich ist der Neustart der gesamten Maschine erforderlich.

SQL30082 RC=24 während CONNECT

Symptom

SQLCODE -30082 Angegebene Benutzer-ID oder Kennwort ist falsch.

Lösung

Stellen Sie sicher, dass in der Anweisung CONNECT das richtige Kennwort angegeben ist, falls es erforderlich ist. Das Kennwort kann nicht an die Zielservedatenbank gesendet werden, weil es nicht verfügbar ist. Ein Kennwort muss vom IBM Data Server-Client an die Zielservedatenbank gesendet werden. Auf bestimmten Plattformen, zum Beispiel AIX, kann das Kennwort nur abgerufen werden, wenn es in der Anweisung CONNECT bereitgestellt wird.

Teil 7. Anhänge und Schlussteil

Anhang A. Übersicht über die technischen Informationen zu DB2

Die technischen Informationen zu DB2 stehen über die folgenden Tools und Methoden zur Verfügung:

- DB2-Informationszentrale
 - Themen (zu Tasks, Konzepten und Referenzinformationen)
 - Hilfe für DB2-Tools
 - Beispielprogramme
 - Lernprogramme
- DB2-Bücher
 - PDF-Dateien (für den Download verfügbar)
 - PDF-Dateien (auf der DB2-PDF-DVD)
 - Gedruckte Bücher
- Befehlszeilenhilfe
 - Hilfe für Befehle
 - Hilfe für Nachrichten

Anmerkung: Die Themen der DB2-Informationszentrale werden häufiger aktualisiert als die PDF- und Hardcopybücher. Um stets die neuesten Informationen zur Verfügung zu haben, sollten Sie die Dokumentationsaktualisierungen installieren, sobald diese verfügbar sind, oder die DB2-Informationszentrale unter ibm.com aufrufen.

Darüber hinaus können Sie auf zusätzliche technische Informationen zu DB2, wie beispielsweise technische Hinweise (Technotes), White Papers und IBM Redbooks, online über ibm.com zugreifen. Rufen Sie die Website 'DB2 Information Management - Software - Library' unter <http://www.ibm.com/software/data/sw-library/> auf.

Feedback zur Dokumentation

Senden Sie uns Ihr Feedback zur DB2-Dokumentation! Wenn Sie Anregungen zur Verbesserung der DB2-Dokumentation haben, senden Sie eine E-Mail an db2docs@ca.ibm.com. Das DB2-Dokumentationsteam bearbeitet das gesamte Feedback, kann jedoch nicht im Einzelnen auf Ihre E-Mails antworten. Nennen Sie uns, wenn möglich, konkrete Beispiele, sodass wir die Problemstellung besser beurteilen können. Wenn Sie uns Feedback zu einem bestimmten Thema oder einer bestimmten Hilfedatei senden, geben Sie den entsprechenden Titel sowie die URL an.

Verwenden Sie diese E-Mail-Adresse nicht, wenn Sie sich an die DB2-Kundenunterstützung wenden möchten. Wenn ein technisches Problem bei DB2 vorliegt, das Sie mithilfe der Dokumentation nicht beheben können, fordern Sie beim zuständigen IBM Service-Center Unterstützung an.

Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format

Die folgenden Tabellen enthalten eine Beschreibung der DB2-Bibliothek, die im IBM Publications Center unter www.ibm.com/shop/publications/order zur Verfügung steht. Über die folgende Adresse können Sie englische Handbücher im PDF-Format sowie übersetzte Versionen zu DB2 Version 9.5 herunterladen: www.ibm.com/support/docview.wss?rs=71&uid=swg2700947.

In den Tabellen sind die Bücher, die in gedruckter Form zur Verfügung stehen, gekennzeichnet; möglicherweise sind diese in Ihrem Land oder Ihrer Region jedoch nicht verfügbar.

Die Formnummer wird bei jeder Aktualisierung eines Handbuchs erhöht. Anhand der nachfolgenden Liste können Sie sicherstellen, dass Sie die jeweils neueste Version des Handbuchs lesen.

Anmerkung: Die DB2-Informationszentrale wird häufiger aktualisiert als die PDF- und Hardcopybücher.

Tabelle 18. Technische Informationen zu DB2

Name	IBM Form	In gedruckter Form verfügbar
<i>Administrative API Reference</i>	SC23-5842-01	Ja
<i>Administrative Routines and Views</i>	SC23-5843-01	Nein
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC23-5844-01	Ja
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC23-5845-01	Ja
<i>Command Reference</i>	SC23-5846-01	Ja
<i>Dienstprogramme für das Versetzen von Daten Handbuch und Referenz</i>	SC12-3917-01	Ja
<i>Datenrecovery und hohe Verfügbarkeit Handbuch und Referenz</i>	SC12-3919-01	Ja
<i>Datenserver, Datenbanken und Datenbankobjekte</i>	SC12-3912-01	Ja
<i>Datenbanksicherheit</i>	SC12-3914-01	Ja
<i>Developing ADO.NET and OLE DB Applications</i>	SC23-5851-01	Ja
<i>Developing Embedded SQL Applications</i>	SC23-5852-01	Ja
<i>Developing Java Applications</i>	SC23-5853-01	Ja
<i>Developing Perl and PHP Applications</i>	SC23-5854-01	Nein
<i>Developing User-defined Routines (SQL and External)</i>	SC23-5855-01	Ja
<i>Getting Started with Database Application Development</i>	GC23-5856-01	Ja

Tabelle 18. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar
<i>Installation und Verwaltung von DB2 unter Linux und Windows - Erste Schritte</i>	GC12-3922-01	Ja
<i>Internationalisierung</i>	SC12-3916-01	Ja
<i>Fehlernachrichten, Band 1</i>	GI11-3098-00	Nein
<i>Fehlernachrichten, Band 2</i>	GI11-3099-00	Nein
<i>Migration</i>	GC12-3921-01	Ja
<i>Net Search Extender Verwaltung und Benutzerhandbuch</i>	SC12-3979-01	Ja
<i>Partitionierung und Clustering</i>	SC12-3915-01	Ja
<i>Query Patroller Verwaltung und Benutzerhandbuch</i>	SC12-3977-00	Ja
<i>IBM Data Server-Clients - Einstieg</i>	GC12-3924-01	Nein
<i>DB2-Server - Einstieg</i>	GC12-3923-01	Ja
<i>Spatial Extender und Geodetic Data Management Feature Benutzer- und Referenzhandbuch</i>	SC12-3978-01	Ja
<i>SQL Reference, Volume 1</i>	SC23-5861-01	Ja
<i>SQL Reference, Volume 2</i>	SC23-5862-01	Ja
<i>Systemmonitor Handbuch und Referenz</i>	SC12-3918-01	Ja
<i>Fehlerbehebung</i>	GI11-3097-01	Nein
<i>Optimieren der Datenbankanleistung</i>	SC12-3913-01	Ja
<i>Lernprogramm für Visual Explain</i>	SC12-3932-00	Nein
<i>Neue Funktionen</i>	SC12-3928-01	Ja
<i>Workload-Manager Handbuch und Referenz</i>	SC12-3929-01	Ja
<i>pureXML - Handbuch</i>	SC12-3930-01	Ja
<i>XQuery - Referenz</i>	SC12-3931-01	Nein

Tabelle 19. Technische Informationen zu DB2 Connect

Name	IBM Form	In gedruckter Form verfügbar
<i>DB2 Connect Personal Edition - Einstieg</i>	GC12-3926-01	Ja
<i>DB2 Connect-Server - Einstieg</i>	GC12-3927-01	Ja
<i>DB2 Connect Benutzerhandbuch</i>	SC12-3925-01	Ja

Tabelle 20. Technische Informationen zu Information Integration

Name	IBM Form	In gedruckter Form verfügbar
<i>Information Integration: Föderierte Systeme - Verwaltung</i>	SC12-3759-01	Ja
<i>Information Integration: ASNCLP Program Reference for Replication and Event Publishing</i>	SC19-1018-02	Ja
<i>Information Integration: Konfiguration föderierter Datenquellen</i>	SC12-3777-01	Nein
<i>Information Integration: SQL Replication - Handbuch und Referenz</i>	SC12-3782-01	Ja
<i>Information Integration: Replikation und Event-Publishing - Einführung</i>	GC12-3779-01	Ja

Bestellen gedruckter DB2-Bücher

Gedruckte DB2-Bücher können Sie in den meisten Ländern oder Regionen online bestellen. Das Bestellen gedruckter DB2-Bücher ist stets über den zuständigen IBM Ansprechpartner möglich. Beachten Sie hierbei bitte, dass einige Softcopybücher auf der DVD mit der *DB2-PDF-Dokumentation* nicht in gedruckter Form verfügbar sind. So sind beispielsweise die beiden Bände des Handbuchs *DB2 Fehlernachrichten* nicht in gedruckter Form erhältlich.

Gedruckte Versionen vieler DB2-Bücher, die auf der DVD mit der DB2-PDF-Dokumentation verfügbar sind, können gegen eine Gebühr bei IBM bestellt werden. Abhängig vom jeweiligen Land bzw. der jeweiligen Region können Sie Bücher möglicherweise online über das IBM Publications Center bestellen. Ist im jeweiligen Land bzw. der jeweiligen Region keine Onlinebestellung möglich, können Sie gedruckte DB2-Bücher stets über den zuständigen IBM Ansprechpartner bestellen. Nicht alle Bücher, die auf der DVD mit der DB2-PDF-Dokumentation verfügbar sind, können in gedruckter Form bestellt werden.

Anmerkung: Über <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5> haben Sie Zugriff auf die DB2-Informationszentrale, wo Sie die neueste und umfassendste DB2-Dokumentation finden.

Gehen Sie wie folgt vor, um gedruckte DB2-Bücher zu bestellen:

- Informationen dazu, ob in Ihrem Land oder Ihrer Region die Bestellung von gedruckten DB2-Büchern möglich ist, finden Sie auf der Website mit dem IBM Publications Center unter <http://www.ibm.com/shop/publications/order>. Wählen Sie ein Land, eine Region oder eine Sprache aus, um die Bestellinformationen für Veröffentlichungen aufzurufen, und führen Sie dann die entsprechenden Schritte des Bestellverfahrens für Ihr Land bzw. Ihre Region aus.
- Gehen Sie wie folgt vor, um gedruckte DB2-Bücher beim zuständigen IBM Ansprechpartner zu bestellen:
 1. Kontaktinformationen zum zuständigen Ansprechpartner finden Sie auf einer der folgenden Websites:
 - IBM Verzeichnis weltweiter Kontakte unter www.ibm.com/planetwide.

- Website mit IBM Veröffentlichungen unter <http://www.ibm.com/shop/publications/order>. Wählen Sie das gewünschte Land, die gewünschte Region oder die gewünschte Sprache aus, um auf die entsprechende Homepage mit Veröffentlichungen Ihres Landes bzw. Ihrer Region zuzugreifen. Folgen Sie auf dieser Seite dem Link für Informationen zu dieser Site ("About this Site").
- 2. Geben Sie bei Ihrem Anruf an, dass Sie eine DB2-Veröffentlichung bestellen möchten.
- 3. Teilen Sie dem zuständigen Ansprechpartner die Titel und Formularnummern der Bücher mit, die Sie bestellen möchten. Titel und Formularnummern finden Sie unter „Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format“ auf Seite 166.

Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor

DB2 gibt für Bedingungen, die aufgrund einer SQL-Anweisung generiert werden können, einen SQLSTATE-Wert zurück. Die SQLSTATE-Hilfe erläutert die Bedeutung der SQL-Statuswerte und der SQL-Statusklassencodes.

Zum Aufrufen der Hilfe für SQL-Statuswerte müssen Sie den Befehlszeilenprozessor öffnen und Folgendes eingeben:

? *sqlstate* oder ? *klassencode*

Hierbei steht *sqlstate* für einen gültigen fünfstelligen SQL-Statuswert und *klassencode* für die ersten beiden Ziffern dieses Statuswertes.

So kann beispielsweise durch die Eingabe von ? 08003 Hilfe für den SQL-Statuswert 08003 angezeigt werden, durch die Eingabe von ? 08 Hilfe für den Klassencode 08.

Zugriff auf verschiedene Versionen der DB2-Informationszentrale

Für Themen aus DB2 Version 9.5 lautet die URL der DB2-Informationszentrale <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>.

Für Themen aus DB2 Version 9 lautet die URL der DB2-Informationszentrale <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>.

Für Themen aus DB2 Version 8 lautet die URL der Informationszentrale (Version 8, 'Information - Unterstützung') <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>.

Anzeigen von Themen in der gewünschten Sprache in der DB2-Informationszentrale

In der DB2-Informationszentrale werden Themen, wenn möglich, in der Sprache angezeigt, die in den Vorgaben Ihres Browsers angegeben ist. Falls ein Thema nicht in die gewünschte Sprache übersetzt wurde, wird es in der DB2-Informationszentrale in Englisch angezeigt.

- Um Themen in der gewünschten Sprache im Browser 'Internet Explorer' anzuzeigen, gehen Sie wie folgt vor:
 1. Klicken Sie im Internet Explorer **Extras** —> **Internetoptionen...** —> **Sprachen...** an. Das Fenster **Spracheinstellung** wird geöffnet.

2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
 - Klicken Sie den Knopf **Hinzufügen...** an, um eine neue Sprache zur Liste hinzuzufügen.

Anmerkung: Das Hinzufügen einer Sprache bedeutet nicht zwangsläufig, dass der Computer über die erforderlichen Schriftarten verfügt, um die Themen in der gewünschten Sprache anzuzeigen.

- Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Nach oben** aus, bis die Sprache an erster Stelle in der Liste steht.
3. Löschen Sie den Inhalt des Browser-Cache, und aktualisieren Sie anschließend die Seite, um die DB2-Informationszentrale in der gewünschten Sprache anzuzeigen.
- Um Themen in der gewünschten Sprache in einem Firefox- oder Mozilla-Browser anzuzeigen, gehen Sie wie folgt vor:
 1. Wählen Sie den Knopf im Bereich **Languages** des Dialogfensters **Tools** —> **Options** —> **Advanced** aus. Die Anzeige für die Auswahl der Sprache wird im Fenster mit den Einstellungen aufgerufen.
 2. Stellen Sie sicher, dass die gewünschte Sprache als erster Eintrag in der Liste angegeben ist.
 - Wenn Sie eine neue Sprache zur Liste hinzufügen möchten, klicken Sie den Knopf **Add...** an, um eine Sprache im entsprechenden Fenster auszuwählen.
 - Um eine Sprache an den Anfang der Liste zu verschieben, wählen Sie zunächst die gewünschte Sprache und anschließend den Knopf **Move Up** aus, bis die Sprache an erster Stelle in der Liste steht.
 3. Löschen Sie den Inhalt des Browser-Cache, und aktualisieren Sie anschließend die Seite, um die DB2-Informationszentrale in der gewünschten Sprache anzuzeigen.

Bei einigen Kombinationen aus Browser und Betriebssystem müssen Sie möglicherweise auch die Ländereinstellungen des Betriebssystems in die gewünschte Locale und Sprache ändern.

Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationszentrale

Wenn Sie die DB2-Informationszentrale lokal installiert haben, können Sie Dokumentationsaktualisierungen von IBM abrufen und installieren.

Zur Aktualisierung der lokal installierten DB2-Informationszentrale sind die folgenden Schritte erforderlich:

1. Stoppen Sie die DB2-Informationszentrale auf Ihrem Computer, und starten Sie die Informationszentrale im Standalone-Modus erneut. Die Ausführung der Informationszentrale im Standalone-Modus verhindert, dass andere Benutzer in Ihrem Netz auf die Informationszentrale zugreifen, und ermöglicht das Anwenden von Aktualisierungen. DB2-Informationszentralen, deren Installation nicht als Administrator oder Root ausgeführt wurde, werden stets im Standalone-Modus ausgeführt.

2. Verwenden Sie die Aktualisierungsfunktion, um zu prüfen, welche Aktualisierungen verfügbar sind. Falls Aktualisierungen verfügbar sind, die Sie installieren möchten, können Sie die Aktualisierungsfunktion verwenden, um diese abzurufen und zu installieren.

Anmerkung: Wenn es in der verwendeten Umgebung erforderlich ist, die Aktualisierungen für die DB2-Informationszentrale auf einer Maschine zu installieren, die nicht über eine Verbindung zum Internet verfügt, müssen Sie die Aktualisierungssite auf ein lokales Dateisystem spiegeln und dabei eine Maschine verwenden, die mit dem Internet verbunden ist und auf der die DB2-Informationszentrale installiert ist. Wenn viele Benutzer Ihres Netzes die Dokumentationsaktualisierungen installieren sollen, können Sie die Zeit, die jeder einzelne Benutzer für die Aktualisierungen benötigt, reduzieren, indem Sie die Aktualisierungssite lokal spiegeln und ein Proxy dafür erstellen. Ist dies der Fall, verwenden Sie die Aktualisierungsfunktion, um die Pakete abzurufen. Die Aktualisierungsfunktion ist jedoch nur im Standalone-Modus verfügbar.

3. Stoppen Sie die im Standalone-Modus gestartete Informationszentrale, und starten Sie die DB2-Informationszentrale auf Ihrem Computer erneut.

Anmerkung: Unter Windows Vista müssen Sie zur Ausführung der nachfolgend aufgeführten Befehle über Administratorberechtigung verfügen. Zum Starten einer Eingabeaufforderung oder eines Grafiktools mit vollen Administratorberechtigungen klicken Sie mit der rechten Maustaste auf die Verknüpfung, und wählen Sie **Als Administrator ausführen** aus.

Gehen Sie wie folgt vor, um die auf Ihrem Computer bzw. Intranet-Server installierte DB2-Informationszentrale zu aktualisieren:

1. Stoppen Sie die DB2-Informationszentrale.
 - Unter Windows klicken Sie **Start** → **Einstellungen** → **Systemsteuerung** → **Verwaltung** → **Dienste** an. Klicken Sie mit der rechten Maustaste die **DB2-Informationszentrale** an, und wählen Sie **Stoppen** aus.
 - Unter Linux: Geben Sie den folgenden Befehl ein:
`/etc/init.d/db2icdv95 stop`
2. Starten Sie die Informationszentrale im Standalone-Modus.
 - Unter Windows:
 - a. Öffnen Sie ein Befehlsfenster.
 - b. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die DB2-Informationszentrale im Verzeichnis <Programme>\IBM\DB2 Information Center\Version 9.5 installiert, wobei <Programme> das Verzeichnis der Programmdateien (Program Files) angibt.
 - c. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis doc\bin.
 - d. Führen Sie die Datei help_start.bat aus:
`help_start.bat`
 - Unter Linux:
 - a. Navigieren Sie zu dem Pfad, in dem die Informationszentrale installiert ist. Standardmäßig ist die DB2-Informationszentrale im Verzeichnis /opt/ibm/db2ic/V9.5 installiert.
 - b. Navigieren Sie vom Installationsverzeichnis in das Verzeichnis doc/bin.
 - c. Führen Sie das Script help_start aus:
`help_start`

Der standardmäßig auf dem System verwendete Web-Browser wird aufgerufen und zeigt die Standalone-Informationszentrale an.

3. Klicken Sie den Aktualisierungsknopf (🔄) an. Klicken Sie im rechten Fenster der Informationszentrale den Knopf für die Suche nach Aktualisierungen an. Eine Liste der Aktualisierungen für die vorhandene Dokumentation wird angezeigt.
4. Wählen Sie zum Initiieren des Installationsprozesses die gewünschten Aktualisierungen aus, und klicken Sie anschließend den Knopf für die Installation der Aktualisierungen an.
5. Klicken Sie nach Abschluss des Installationsprozesses **Fertig stellen** an.
6. Stoppen Sie die im Standalone-Modus gestartete Informationszentrale:
 - Unter Windows: Navigieren Sie in das Verzeichnis `doc\bin` des Installationsverzeichnisses, und führen Sie die Datei `help_end.bat` aus:

```
help_end.bat
```
 - Unter Linux: Navigieren Sie in das Verzeichnis `doc/bin` des Installationsverzeichnisses, und führen Sie das Script `help_end` aus:

```
help_end
```
7. Starten Sie die DB2-Informationszentrale erneut.
 - Unter Windows klicken Sie **Start** → **Einstellungen** → **Systemsteuerung** → **Verwaltung** → **Dienste** an. Klicken Sie mit der rechten Maustaste die **DB2-Informationszentrale** an, und wählen Sie **Start** aus.
 - Unter Linux: Geben Sie den folgenden Befehl ein:

```
/etc/init.d/db2icdv95 start
```

In der aktualisierten DB2-Informationszentrale werden die neuen und aktualisierten Themen angezeigt.

DB2-Lernprogramme

Die DB2-Lernprogramme unterstützen Sie dabei, sich mit den unterschiedlichen Aspekten der DB2-Produkte vertraut zu machen. Die Lerneinheiten bieten eine in einzelne Schritte unterteilte Anleitung.

Vorbereitungen

Die XHTML-Version des Lernprogramms kann über die Informationszentrale unter <http://publib.boulder.ibm.com/infocenter/db2help/> angezeigt werden.

In einigen der Lerneinheiten werden Beispieldaten und Codebeispiele verwendet. Informationen zu bestimmten Voraussetzungen für die Ausführung der Tasks finden Sie in der Beschreibung des Lernprogramms.

DB2-Lernprogramme

Klicken Sie zum Anzeigen des Lernprogramms den Titel an.

„pureXML“ in *pureXML - Handbuch*

Einrichten einer DB2-Datenbank, um XML-Daten zu speichern und Basisoperationen mit dem nativen XML-Datenspeicher auszuführen.

„Visual Explain“ in *Lernprogramm für Visual Explain*

Analysieren, Optimieren und Anpassen von SQL-Anweisungen zur Leistungsverbesserung mithilfe von Visual Explain.

Informationen zur Fehlerbehebung in DB2

Eine breite Palette verschiedener Informationen zur Fehlerbestimmung und Fehlerbehebung steht zur Verfügung, um Sie bei der Verwendung von DB2-Produkten zu unterstützen.

DB2-Dokumentation

Informationen zur Fehlerbehebung stehen im Handbuch DB2-Fehlerbehebung oder im Abschnitt zur Unterstützung und Fehlerbehebung der DB2-Informationszentrale zur Verfügung. Dort finden Sie Informationen dazu, wie Sie Probleme mithilfe der DB2-Diagnosetools und -Dienstprogramme eingrenzen und identifizieren können, Lösungen für einige der häufigsten Probleme sowie weitere Hinweise zur Behebung von Fehlern und Problemen, die bei der Verwendung der DB2-Produkte auftreten können.

DB2-Website mit technischer Unterstützung

Auf der DB2-Website mit technischer Unterstützung finden Sie Informationen zu Problemen und den möglichen Ursachen und Fehlerbehebungsmaßnahmen. Die Website mit technischer Unterstützung enthält Links zu den neuesten DB2-Veröffentlichungen, technischen Hinweisen (TechNotes), APARs (Authorized Program Analysis Reports) und Fehlerkorrekturen, Fixpacks sowie weiteren Ressourcen. Sie können diese Wissensbasis nach möglichen Lösungen für aufgetretene Probleme durchsuchen.

Rufen Sie die DB2-Website mit technischer Unterstützung unter <http://www.ibm.com/software/data/db2/udb/support.html> auf.

Bedingungen

Die Berechtigungen zur Nutzung dieser Veröffentlichungen werden Ihnen auf der Basis der folgenden Bedingungen gewährt.

Persönliche Nutzung: Sie dürfen diese Veröffentlichungen für Ihre persönliche, nicht kommerzielle Nutzung unter der Voraussetzung vervielfältigen, dass alle Eigentumsvermerke erhalten bleiben. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM nicht weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Kommerzielle Nutzung: Sie dürfen diese Veröffentlichungen nur innerhalb Ihres Unternehmens und unter der Voraussetzung, dass alle Eigentumsvermerke erhalten bleiben, vervielfältigen, weitergeben und anzeigen. Sie dürfen diese Veröffentlichungen oder Teile der Veröffentlichungen ohne ausdrückliche Genehmigung von IBM außerhalb Ihres Unternehmens nicht vervielfältigen, weitergeben, anzeigen oder abgeleitete Werke davon erstellen.

Abgesehen von den hier gewährten Berechtigungen erhalten Sie keine weiteren Berechtigungen, Lizenzen oder Rechte (veröffentlicht oder stillschweigend) in Bezug auf die Veröffentlichungen oder darin enthaltene Informationen, Daten, Software oder geistiges Eigentum.

IBM behält sich das Recht vor, die in diesem Dokument gewährten Berechtigungen nach eigenem Ermessen zurückzuziehen, wenn sich die Nutzung der Veröffentlichungen für IBM als nachteilig erweist oder wenn die obigen Nutzungsbestimmungen nicht genau befolgt werden.

Sie dürfen diese Informationen nur in Übereinstimmung mit allen anwendbaren Gesetzen und Vorschriften, einschließlich aller US-amerikanischen Exportgesetze und Verordnungen, herunterladen und exportieren.

IBM übernimmt keine Gewährleistung für den Inhalt dieser Informationen. Diese Veröffentlichungen werden auf der Grundlage des gegenwärtigen Zustands (auf "as-is"-Basis) und ohne eine ausdrückliche oder stillschweigende Gewährleistung für die Handelsüblichkeit, die Verwendungsfähigkeit oder die Freiheit der Rechte Dritter zur Verfügung gestellt.

Anhang B. Bemerkungen

Die vorliegenden Informationen wurden für Produkte und Services entwickelt, die auf dem deutschen Markt angeboten werden.

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 zuständigen IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Services von IBM verwendet werden können. Anstelle der Produkte, Programme oder Services können auch andere ihnen äquivalente Produkte, Programme oder Services verwendet werden, solange diese keine gewerblichen oder andere 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. Lizenzanforderungen sind schriftlich an folgende Adresse zu richten (Anfragen an diese Adresse müssen auf Englisch formuliert werden):

IBM Director of Licensing
IBM Europe, Middle East & Africa
Tour Descartes
2, avenue Gambetta
92066 Paris La Defense
France

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 oder in Technical News Letters (TNLs) bekannt gegeben. IBM kann ohne weitere Mitteilung jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Dieses Dokument enthält möglicherweise Links oder Verweise auf Websites und Ressourcen anderer Anbieter. Es bestehen keine Zusicherungen, Gewährleistungen oder Verpflichtungen von IBM hinsichtlich der Websites oder Ressourcen anderer Anbieter, auf die im vorliegenden Dokument verwiesen wird, Zugriff besteht oder Links vorhanden sind. Ein Link auf eine Website eines anderen Anbieters bedeutet nicht, dass IBM den Inhalt und die Verwendung dieser Website billigt oder deren Eigentümer anerkennt. Darüber hinaus ist IBM nicht an Transaktionen beteiligt und übernimmt keine Verantwortung für Transaktionen zwischen Ihnen und anderen Anbietern, auch wenn die Informationen (oder Links) zu diesen Anbietern auf einer IBM Website zur Verfügung stehen. IBM ist nicht für die Verfügbarkeit solcher externen Sites oder Ressourcen verantwortlich und übernimmt keine Verantwortung oder Haftung für Inhalte, Services, Produkte oder sonstiges Material, die bzw. das auf diesen oder über diese Sites oder Ressourcen verfügbar sind. Die Software anderer Anbieter unterliegt den Lizenzbedingungen der jeweiligen Software.

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
8200 Warden Avenue
Markham, Ontario
L6G 1C7
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 Dokument aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt auf der Basis der IBM Rahmenvereinbarung sowie der Allgemeinen Geschäftsbedingungen von IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer kontrollierten 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.

Alle Informationen zu Produkten anderer Anbieter stammen von den Anbietern der aufgeführten Produkte, deren veröffentlichten Ankündigungen oder anderen allgemein verfügbaren Quellen. IBM hat diese Produkte nicht getestet und kann daher keine Aussagen zu Leistung, Kompatibilität oder anderen Merkmalen machen. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten von IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele von 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.

COPYRIGHTLIZENZ:

Diese Veröffentlichung enthält Musteranwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Musterprogramme 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 Musterprogramme geschrieben werden. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet. Daher kann IBM die Zuverlässigkeit, Wartungsfreundlichkeit oder Funktion dieser Programme weder zusagen noch gewährleisten.

Kopien oder Teile der Musterprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Musterprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. *„Jahr/Jahre angeben“*. Alle Rechte vorbehalten.

Marken

Folgende Namen sind Marken oder eingetragene Marken der International Business Machines Corporation in den USA und/oder anderen Ländern.

pureXML	1-2-3
Distributed Relational Database Architecture (DRDA)	DB2
AIX	i5/OS
POWER	Encina
WebSphere	OS/390
DB2 Connect	Passport Advantage
DB2 Universal Database	TXSeries
Redbooks	z/OS
System i	developerWorks
SecureWay	CICS
IBM	zSeries
Lotus	DRDA
OS/400	Approach
eServer	ibm.com
pSeries	iSeries

Folgende Namen sind Marken oder eingetragene Marken anderer Unternehmen.

- Linux ist eine eingetragene Marke von Linus Torvalds in den USA und/oder anderen Ländern.
- Java und alle Java-basierten Marken sind Marken von Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- UNIX ist eine eingetragene Marke von The Open Group in den USA und/oder anderen Ländern.
- Intel Xeon und Pentium sowie Intel sind Marken der Intel Corporation oder deren Tochtergesellschaften in den USA und/oder anderen Ländern.
- Microsoft und Windows sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicennamen können Marken anderer Hersteller sein.

Index

A

Abfrageblöcke
 Datenübertragungsgeschwindigkeit in DB2 Connect erhöhen 130
ACCRDB, Befehl 149
ACCRDBRM, Befehl 149
ACCSEC, Befehl 149
agentpri, Konfigurationsparameter des Datenbankmanagers 125
Aktualisieren
 DB2 für z/OS-Diagnosemonitor 84
Aktualisierungen
 Datenbankverzeichnisse 33
 DB2-Informationszentrale 170
Aktualisierungen auf mehreren Systemen
 aktivieren 59
 DUOW (Distributed Unit of Work, verteilte Arbeitseinheit) 59
 Steuerzentrale 60
 Synchronisationspunktmanager 61
 Test 60
Alertobjekte
 anzeigen 89
Alertzusammenfassungen
 anzeigen 88
Anpassung
 Verzeichnisse, Arbeitsblätter 40
Antwortzeit 101
Anweisung SELECT
 FOR FETCH ONLY 107
 im Anwendungsentwurf 107
Anweisungen
 COMMIT 107
 DESCRIBE 107
 EXECUTE IMMEDIATE 107
 FOR FETCH ONLY 107
 GRANT 53
 PREPARE 107
 REVOKE 53
 ROLLBACK
 Anwendungsentwurf 107
 SELECT 107
Anwendungen
 Bindung 55
 Compound-SQL-Anweisung 107
 Entwurf 107
 gespeicherte Prozeduren 107
 Leistung 107
 Web
 mit DB2 Connect 19
Anwendungsentwicklung 107
 DB2 AD-Client 15
 ODBC 15
Anwendungsname, Monitorelement 78
Anwendungsrequester (AR)
 DRDA, Definition 10
 Parameter 40
Anwendungsserver (AS)
 Clients mit allen Funktionen 23
 DB2 Connect-Unterstützung 23
 DRDA, Definition 10

Anwendungsserver (AS) (*Forts.*)
 Implementierung 23
 Konfiguration 23
 Übersicht 23
 zweischichtige und dreischichtige Modelle 23
Arbeitsblätter
 Anpassung von Verzeichnissen 40
AS, Zieldatenbankname 35
Assistenten
 Aktualisierung auf mehreren Systemen 60
Ausführen von Vergleichstests
 Leistung 101
Authentifizierung 40
 Gültigkeitsprüfung 50
 Typen
 CLIENT 50, 52
 DATA_ENCRYPT 50
 KERBEROS 50
 SERVER 50
 SERVER_ENCRYPT 50
 Standardwert 50
 Übersicht 50
Authentifizierungswert 33
Automatische Clientweiterleitung
 Beschreibung 95
 Konfiguration für DB2 Connect 95
 Verbindungsfehler 97

B

Bedingungen
 Verwendung der Veröffentlichungen 173
Befehle
 ACCRDB 149
 ACCRDBRM 149
 ACCSEC 149
 COMMIT 149
 db2drdat 147
 db2trc 143, 145
 EXCSAT 149
 EXCSATRD 149
 GET SNAPSHOT 76
 SECCHK 149
Befehlszeilenprozessor (CLP)
 Leistung 107
 SQL-Anweisungen 7
Beispiele
 Verbindungskonzentratoren 113
 XA-Konzentratoren 113
Bemerkungen 175
Berechtigungen
 Bindung 55
Berechtigungs-ID 78
Bestellen von DB2-Büchern 168
Bidirektionale CCSID-Unterstützung
 BIDI, Parameter 35
BINDADD, Zugriffsberechtigung
 Bindeberechtigung 55
Bindeliste 55
Bindung
 Anwendungen und Dienstprogramme 55

- Bindung (*Forts.*)
 - Berechtigung
 - Parametermarken mit Offset 55
 - Pakete 55
 - Paketnamen 55
- Blockgröße 125
- Bootstrap Data Set (BSDS), Parameter
 - z/OS und OS/390 34
- Bücher
 - gedruckt
 - bestellen 168

C

- CCSID (Coded Character Set Identifier)
 - bidirektionale Unterstützung
 - Beschreibung 35
- CGI-Programmierung (GCI = Common Gateway Interface)
 - Einschränkungen 19
 - Vorteile 19
- CHAR, Datentyp
 - Beschreibung 133
- Character Data Representation Architecture (CDRA) 10
- CLI (Call Level Interface)
 - Anwendungen
 - CURRENTPACKAGESET 52
 - gesicherte Verbindungen 45
 - Übersicht 135
- CLIENT, Authentifizierungstyp
 - DB2 Connect, Überlegungen 50
- Client-DB-Aliasname 78
- Client-NNAME 78
- Clientanwendungen
 - ID 78
 - Verbindungen wiederherstellen 95
- Clientfolgenummer 78
- Clientprodukt-ID 78
- Codepage-ID 78
- COMMIT, Anweisung
 - statisch gebunden 107
- COMMIT, Befehl
 - Traceausgabepuffer 149
- Common Gateway Interface (CGI), Programmierung
 - Einschränkungen 19
 - Vorteile 19
- Compound-SQL-Anweisung
 - nicht ganzheitlich 107
- CPU
 - Nutzungstools 101
- CREATE IN COLLECTION NULLID, Berechtigung 55
- CURRENTPACKAGESET, CLI/ODBC-Schlüsselwort 52

D

- D (disconnect), Parameter 35
- DATA_ENCRYPT, Authentifizierungstyp 50
- Database Connection Services (DCS), Verzeichnis
 - Werte 35
- Daten
 - Blockung 107
 - Datenflüsse
 - DB2 Connect 10, 101
 - Quellen
 - verteilte Anforderung 12
 - übertragen
 - Geschwindigkeit 133

- Daten (*Forts.*)
 - übertragen (*Forts.*)
 - Leistung 133
 - zwischen Hosts und Workstations 65
 - Übertragung
 - Geschwindigkeit 101
- Datenbankanforderungen
 - Gruppierung für die Leistung 107
- Datenbanken
 - Aliasnamen
 - Arbeitsblatt zur Verzeichnisanpassung 40
 - Systemdatenbankverzeichnis 33
 - Anforderungen gruppieren 107
 - Hostdatenbanken 6
 - Leistungsanalysetools 101
 - Namen
 - Arbeitsblatt zur Verzeichnisanpassung 40
 - DCS-Verzeichnis 35
 - RDBNAM, Objekt 149
 - Systemdatenbankverzeichnis 33
 - Optimierung 127
- Datenbanksystemmonitor
 - ferne Clients 75
 - Übersicht 7
- Datenbankverzeichnisse
 - aktualisieren 33
 - DCS (Database Connection Services) 33
 - Knoten 33
 - mehrfache Einträge 41
- Datenblockung 107
- Datenkonvertierung
 - Host 133
- Datentypen
 - CHAR 133
 - gepackt dezimal 133
 - gezont dezimal 133
 - Gleitkomma
 - Hostdatenkonvertierung 133
 - INTEGER
 - Hostdatenkonvertierung 133
 - Umsetzung
 - Auswirkung auf die Leistung 133
 - VARCHAR
 - Übersicht 133
 - Zeichen 133
- Datumsangaben
 - Zeitzoneunterstützung 35
- DB2 Connect
 - Enterprise Edition
 - APIs 22
 - JDBC 22
 - Konnektivitätsserver 18
 - SQLJ 22
 - Transaktionsverarbeitungsmonitore 27
 - Web-Server 23
 - Webanwendungen 19
 - XA-kompatible Transaktionsmanager 62
 - Erweiterungen 3
 - Hostunterstützung 15
 - Konnektivitätsserverszenarios 15
 - Produkte 3
 - Sicherheit 45
 - SYSPLEX-Unterstützung 121
 - System i-Unterstützung 15
 - Übersicht 3
 - Verbindungskonzentratoren 119
 - Versetzen von Daten 65

- DB2 für z/OS-Diagnosemonitor
 - Anzeigen, Übergeben, Speichern empfohlener Aktionen 85
 - Anzeigen von Alertobjekten 89
 - Anzeigen von Alertzusammenfassungen 88
 - starten, stoppen, aktualisieren 84
 - Übersicht 83
- DB2-Informationszentrale
 - Aktualisierung 170
 - in verschiedenen Sprachen anzeigen 169
 - Sprachen 169
 - Versionen 169
- DB2 Universal Database für OS/390 und z/OS
 - Bootstrap Data Set 34
 - DOMAIN 34
 - DYNAMICRULES (BIND), Option 52
 - Knotenverzeichniswerte 34
 - Parameter BSDS 34
 - RESPORT 34
 - Sicherheit 52
 - TCPPORT 34
- db2drdat, Befehl
 - Ausgabedatei 147
- db2trc, Befehl
 - Speicherauszug für Traceausgabe erstellen 144
 - Traceausgabe formatieren 145
 - Übersicht 143
- DCS-Verzeichnis (DCS = Database Connection Services)
 - Einträge aktualisieren 33
 - siehe Database Connection Services (DCS), Verzeichnis 35
- dcs1ari.map, Datei 69
- dcs1dsn.map, Datei 69
- dcs1qsq.map, Datei 69
- ddcs400.lst, Datei 55
- ddcsmvs.lst, Datei 55
- ddcstrc, Dienstprogramm 148
- ddcsvm.lst, Datei 55
- ddcsvse.lst, Datei 55
- DDM (Distributed Data Management)
 - db2drdat, Ausgabe 147
 - Distributed Relational Database Architecture (DRDA) 10
- DESCRIBE, Anweisung
 - Compound-SQL-Anweisungen 107
 - Leistung mit der Anweisung PREPARE 107
- Diagnosealert 83
- Diagnoseinformationen
 - Übersicht 142
- Diagnosemonitor
 - DB2 für z/OS 83
- Dienstprogramm EXPORT
 - Übertragung von Daten zwischen Hosts und Workstations 65
- Dienstprogramm IMPORT
 - Daten zwischen Host und Workstation übertragen 65
- Dienstprogramm zur Feststellung des Prozessstatus 142, 149
- Dienstprogramme
 - Bindung 55
 - Datenbanksystemmonitor 7
 - db2drdat 147
 - ddcspkgn 55
 - Prozessstatus 149
 - ps (Prozessstatus) 142, 149
 - Trace 147
 - Verwaltung, DB2 Connect 7
- DIRCACHE, Parameter 125
- Distributed Relational Database Architecture (DRDA)
 - Datenzugriff 9

- Distributed Relational Database Architecture (DRDA) (Forts.)
 - Übersicht 9, 10
- Dokumentation
 - gedruckt 166
 - Nutzungsbedingungen 173
 - PDF 166
 - Übersicht 165
- DUOW (Distributed Unit of Work, verteilte Arbeitseinheit)
 - Aktualisierungen auf mehreren Systemen 59
 - Übersicht 9
 - unterstützte Server 59
 - zweiphasiges Commit 59
- Durchsatz
 - Transaktionen 101
- Dynamisches SQL
 - Auswirkungen auf die Verarbeitung 6
 - CURRENTPACKAGESET, CLI/ODBC-Konfigurationsparameter 52
 - Leistung
 - Verfahren 107

E

- Einschränkungen
 - Verbindungskonzentrator 113
- Empfangspuffer 147
- Empfohlene Aktionen
 - anzeigen, übergeben, speichern 85
- Engpässe bei der Leistung
 - Leistung 101
 - Transaktionen 101
- Entscheidungshilfesystem (DSS) 147
- Et-Zeichen (doppeltes &)
 - SQLCODE-Zuordnungsdatei 69
- Exchange Server Attributes, Befehl 149
- EXCSAT, Befehl 149
- EXCSATRD, Befehl 149
- EXECUTE IMMEDIATE, Anweisung
 - Anwendungsentwurf 107
- EXTNAM, Objekt 149

F

- Fehler
 - Fehlerbehebung 139
- Fehlerbehebung
 - DB2 Connect 159
 - Leistung 129
 - Lernprogramme 173
 - Onlineinformationen 173
 - Tracefunktionen 143
 - DRDA 150, 155
 - Übersicht 139
 - verbinden 139, 140
 - Zusammenstellen von Informationen 139
- Fehlerbestimmung
 - Diagnosetools
 - Übersicht 142
 - Lernprogramme 173
 - nach der Verbindungsherstellung 140
 - Verbindung 139
 - verfügbare Informationen 173
- Fehlernachrichten
 - DB2 Connect 159
- Fensterskalierung
 - RFC-1323-Erweiterungen 132

- Föderierte Datenbanken
 - verteilte Anforderung 12
- Folgennummer für abgehende Daten
 - LIST DCS APPLICATIONS, DB2-Befehl 78
- FOR FETCH ONLY, Klausel
 - Anweisung SELECT 107
- FORCE, Befehl
 - Agenten-ID 78
- Formatted Data Object Content Architecture (FDOCA) 10

G

- Ganzheitliche Compound-SQL-Anweisung
 - von DB2 Connect nicht unterstützt 107
- Gepackt dezimal, Datentyp 133
- Gesicherte Verbindungen 45
 - Benutzer mit CLI/ODBC wechseln 48
 - mit CLI/ODBC 46
- Gesicherter Kontext
 - DB2 Connect-Unterstützung 45
 - mit CLI/ODBC 46
- Gespeicherte Prozeduren
 - Übersicht 23
- GET SNAPSHOT, Befehl 76
- gezont dezimal
 - Datentypen 133
- Gleitkomma
 - Datentyp 133
- GRANT, Anweisung
 - Sicherheit 53

H

- Hardware
 - Netzwerkleistung 133
- Hilfe
 - Konfiguration der Sprache 169
 - SQL-Anweisungen 169
- Hohe Verfügbarkeit
 - DB2 Connect 91
- Hostanwendungs-ID 78
- Hostdatenbanken
 - Konnektivität
 - hohe Verfügbarkeit 93
 - Lastausgleich 93
 - Zugriff mit DB2 Connect Personal Edition 15

I

- IBM WebSphere 21
- INTEGER, Datentyp 133
- INTERRUPT_ENABLED (disconnect), Parameter 35
- iSeries
 - DRDA 10

J

- Java
 - Anwendungsserver
 - APIs 22
 - DB2 Connect 22
 - JDBC 22
 - SQLJ 22

K

- Kennwörter
 - ändern
 - OS/390 und z/OS 52
- Kerberos-Authentifizierungsprotokoll
 - DB2 Connect 50
 - OS/390 und z/OS 51
- Kerndateien
 - Fehlerbestimmung 142
- Knoten
 - Name
 - Arbeitsblatt zur Verzeichnisanpassung 40
 - Knotenverzeichniswerte 34
 - Systemdatenbankwerte 33
 - Verzeichnisse
 - aktualisieren 33
 - Werte 34
- Konfigurationsparameter
 - agentpri 125
 - DIRCACHE 125
 - max_coordagents 111, 113
 - MAXDARI 125
 - num_initagents 111, 113
 - num_poolagents 111, 113
 - numdb 125
 - rqrioblk 125
 - TCP_KEEPALIVE 97
- Konfigurieren
 - Hostverbindungen 15
 - Überlegungen zur Kennwortänderung 52
- Konfigurieren, Aktualisierungen auf mehreren Systemen (Assistent) 60
- Konkurrenzsituation
 - Systemressourcen 129
- Konnektivität
 - Server, DB2 Connect Enterprise Edition 18

L

- Leistung
 - Anwendungen 107
 - Befehlszeilenprozessor (CLP), Auswirkung 107
 - DB2 Connect
 - Datenübertragungsgeschwindigkeit erhöhen 130
 - Fehlerbehebung 129
 - Optimierung 99
 - Übersicht 101
 - Netzwerk
 - Hardware 133
 - ODBC-Zugriff 105
 - OS/390 130
 - Systemressourcen 129
 - Verbindungskonzentrator 118
 - Verbindungspooling 118
 - z/OS 130
- Lernprogramme
 - Fehlerbehebung 173
 - Fehlerbestimmung 173
 - Visual Explain 172
- LIST DCS APPLICATIONS, Befehl 78
- LOCALDATE, Parameter 35

M

- max_coordagents, Konfigurationsparameter des Datenbankmanagers 111, 113

- maxagents, Konfigurationsparameter des Datenbankmanagers 125
- MAXDARI, Konfigurationsparameter 125
- Microsoft Windows
 - Anwendungen 15
- Monitor
 - Leistung 75
 - Verbindungen 75

N

- Netzwerk
 - Datenübertragungsgeschwindigkeit 133
 - Leistungsanalysetools 101
 - Optimierung 127
- Nicht ganzheitliche Compound-SQL-Anweisung
 - Anwendungsentwurf 107
- NOMAP, Parameter
 - DCS-Verzeichnisparameter 69
 - Inaktivieren der SQL-Zuordnung 69
 - SQLCODE-Zuordnung 35
- NONE, Sicherheitseinstufung 53
- NULLID
 - OS/400 55
- num_initagents, Konfigurationsparameter des Datenbankmanagers 111, 113
- num_poolagents, Konfigurationsparameter des Datenbankmanagers 111, 113
- NUMDB
 - Parameter 125

O

- ODBC (Open Database Connectivity)
 - Anwendungen
 - CURRENTPACKAGESET 52
 - Schnittstelle 15
 - Übersicht 135
 - Zugriffsoptimierung 105
- Optimierung
 - DB2 Connect 99
 - DB2 für OS/390 und z/OS 130
 - Leistung
 - Datenbank 127
 - Netzwerk 127
 - Parameter
 - AGENTPRI 125
 - DIRCACHE 125
 - MAXAGENTS 125
 - MAXDARI 125
 - NUMDB 125
 - RQRIOBLK 125
- OS/390
 - DRDA 10

P

- Paging-Blockgröße 125
- Pakete
 - Hostdatenbankserver 55
 - System i-Datenbankserver 55
- Parameter
 - MAXAGENTS 125
 - PRDID 149
 - SYSPLEX 35
 - Verzeichnisse 40

- Parameter (*Forts.*)
 - Zeichenfolgen 41
- Parameterzeichenfolgen
 - doppelte Kommata 35
 - Kommata 35
- Prädikate
 - Logikausführung 107
- PRDID, Parameter 149
- PREPARE, Anweisung
 - Anwendungsentwurf 107
 - Auswirkung auf die Leistung 107
- PROGRAM, Sicherheitseinstufung 53
- Protokolle
 - Richtlinienbewertung 83
- ps, Befehl
 - EXTNAM, Objekt 149
 - Übersicht 142

R

- Relational Connect
 - Produktbeschreibung 8
- Release-Erweiterungen 3
- Resource Access Control Facility (RACF)
 - Sicherheit 53
- REVOKE, Anweisung
 - Sicherheit 53
- Richtlinienbewertungsprotokoll
 - DB2 für z/OS-Diagnosemonitor 83
- ROLLBACK, Anweisung
 - statisch gebunden 107
- RQRIOBLK, Parameter
 - Optimierung 125
- RUOW (Remote Unit of Work, ferne Arbeitseinheit)
 - Beispiel 11
 - Merkmale 11
 - Übersicht 11

S

- SAME, Sicherheitseinstufung 53
- SECCHK, Befehl 149
- SELECT, Anweisung
 - aktualisierbar 107
- Sendepuffer
 - Datentrace 147
- Server
 - Anwendung
 - DB2 Connect EE 23
- SERVER, Authentifizierungstyp 50
- SERVER_ENCRYPT, Authentifizierungstyp 50
- SET CURRENT PACKAGESET, Anweisung 52
- SHOW DETAIL, Monitoroption 78
- Sicherheit
 - DB2 Connect
 - Unterstützung 53
 - Einführung 45
 - erweiterte Codes
 - OS/390 und z/OS 52
 - GRANT, Anweisung 53
 - Hinweise 52
 - Kerberos 51
 - Knotenverzeichniswerte 34
 - REVOKE, Anweisung 53
 - TCP/IP 53
 - Tipps 52

- Sicherheit (Forts.)
 - Typen 40
 - SOCKS
 - Knoten
 - erforderliche Umgebungsvariablen 34
 - Speicherausgang für Traceausgabe in Datei
 - Übersicht 144
 - SQL (Structured Query Language)
 - dynamisch 107
 - statisch 107
 - SQL-Anweisungen
 - DB2 Connect 6
 - Hilfe anzeigen 169
 - SQL_ATTR_
 - TRUSTED_CONTEXT_PASSWORD
 - Verwendung 48
 - TRUSTED_CONTEXT_USERID
 - Verwendung 48
 - USE_TRUSTED_CONTEXT
 - Verwendung 46
 - SQL/DS
 - DRDA 10
 - SQL0965, Fehlercode 159
 - SQL0969, Fehlercode 159
 - SQL1338, Fehlercode 34, 159
 - SQL30020, Fehlercode 159
 - SQL30060, Fehlercode 159
 - SQL30061, Fehlercode 159
 - SQL30073, Fehlercode 159
 - SQL30081N, Fehlercode 159
 - SQL30082, Fehlercode 159
 - SQL5043N, Fehlercode 159
 - SQLCA (SQL-Kommunikationsbereich)
 - Datenpuffer 147
 - SQLCODE-Feld 147
 - SQLCODE
 - Feld im SQL-Kommunikationsbereich 147
 - Zuordnung 69
 - SQLCODE-Zuordnungsdatei 69
 - SQLDA (SQL-Deskriptorbereich)
 - Zuordnungsgröße 107
 - SQLSTATE
 - Klassencodes 69
 - SRVNAM, Objekt 149
 - START MVS, Systembefehl 83
 - Starten
 - DB2 für z/OS-Diagnosemonitor 84
 - Statisches SQL
 - Auswirkungen auf die Verarbeitung 6
 - Leistung 107
 - Steuerzentrale
 - Aktualisierungen auf mehreren Systemen 60
 - STOP MVS, Systembefehl 83
 - Stoppen
 - DB2 für z/OS-Diagnosemonitor 84
 - Symbolischer Bestimmungsname (Symbolic Destination Name) 40
 - Groß-/Kleinschreibung 34
 - Synchronisationspunktmanager (SPM)
 - Standardparameter 62
 - Szenarios 61
 - Sysplex
 - DB2 Connect-Unterstützung 121
 - Fehlertoleranz 122
 - Konfigurationsvoraussetzungen 123
 - Lastausgleich 122
 - Parameter 35
 - Sysplex (Forts.)
 - Prioritätsinformationen 122
 - Überlegungen zu zSeries 121
 - verwenden 122
 - System i
 - DRDA 10
 - Systemdatenbankverzeichnis
 - aktualisieren 33
 - Werte 33
 - Systemressourcen
 - Konkurrenzsituation 129
 - Systemstatus
 - GET SNAPSHOT, Befehl 76
 - Szenarios
 - TCP/IP-Sicherheit 53
- ## T
- TCP/IP
 - ACCSEC, Befehl 149
 - DOMAIN 34
 - Hostnamen 40
 - Konfiguration
 - Hostverbindungen 17
 - Namen von fernen Hosts 34, 40
 - Portnummern 40
 - REPORT 34
 - RFC-1323-Erweiterungen
 - Fensterskalierung 132
 - Rpesynchronisationsport 34
 - SECCHK, Befehl 149
 - Servicenamen 34
 - Sicherheit
 - Szenarios 53
 - überprüft 52
 - TCPPORT 34
 - TCP_KEEPALIVE
 - Konfigurationsparameter des Betriebssystems 97
 - Test
 - Aktualisierungen auf mehreren Systemen 60
 - Token
 - SQLCODEs 69
 - Tools
 - CPU-Auslastung 101
 - Leistung 101
 - Speicherbelegung 101
 - Tools für die Speicherbelegung 101
 - Tracedienstprogramm (db2drdat) 147
 - Tracefunktion
 - DB2-Traces 143, 144, 145
 - DRDA-Traces 150, 155
 - Übersicht 143
 - Traces
 - Ausgabedatei 147, 148
 - Ausgabedateibeispiele 150
 - Daten zwischen DB2 Connect und dem Server 147
 - DRDA
 - interpretieren 147
 - Pufferinformationen für DRDA-Traces 155
 - Transaktionen
 - Aktualisierungen auf mehreren Systemen 9, 59
 - DB2 Connect Enterprise Edition 27
 - Durchsatz 101
 - Transaktionsverarbeitungsmonitore 27
 - Unterstützung 63
 - UOW (Unit of Work, Arbeitseinheit) 9

- Transaktionen (*Forts.*)
 - verteilte
 - unterstützte Server 59
 - verteilte XA-Anwendungen 63
 - zweiphasiges Commit 9
- Transaktionsverarbeitungsmonitore
 - Aktualisierungen auf mehreren Systemen 59
 - Anwendungskennndaten 27
 - Beispiele 27
 - OLTP 27
 - Transaktionen 27
 - Tuxedo 27
- Tuxedo
 - DB2 Connect Enterprise Edition 27

U

- Überwachung
 - Verbindungen
 - DB2 Connect-Server 75
 - Windows Performance Monitor 75
- Umsetzungen
 - Hostdaten 133
- UOW beendet, Nachricht (ENDUOWRM) 149
- UOWs (Units of Work, Arbeitseinheiten)
 - Definition 9
 - ferne 11
 - verteilte 59

V

- VARCHAR, Datentyp
 - Beschreibung 133
- Verbindungen
 - DB2 Connect Enterprise Edition 18
 - direkt zum Host 15
 - direkte, zur Host- oder System i-Datenbank 17
 - erneut herstellen
 - DB2 Connect Enterprise Edition 18
 - direkt zum Host 15
 - Konzentratoren, siehe Verbindungskonzentratoren 113
 - Pooling
 - Übersicht 111
 - Verbindungskonzentratoren 113
 - Vorteile 113
- Verbindungen wiederherstellen
 - Clientanwendungen 95
- Verbindungsfehler
 - automatische Clientweiterleitung 97
- Verbindungskonzentrator 111
- Verbindungskonzentratoren
 - Beispiele 113
 - DB2 Connect 119
 - Dispatcher 113
 - Einschränkungen 113
 - im Vergleich zum Verbindungspooling 118
 - Implementierung 113
 - Konfigurationsparameter 113
 - logische Agenten 113
 - MAX_COORDAGENTS, Konfigurationsparameter 113
 - NUM_INITAGENTS, Konfigurationsparameter 113
 - NUM_POOLAGENTS, Konfigurationsparameter 113
 - Pooling 113
 - Systemaufwand 113
 - Übersicht 111
 - Verarbeitungsagenten 113

- Verbindungskonzentratoren (*Forts.*)
 - XA-Transaktionsunterstützung 113
- Verbindungspooling 111
 - im Vergleich zum Verbindungskonzentrator 118
 - Übersicht 111
- Verbindungsverwaltung 111
- Versetzen von Daten
 - mit DB2 Connect 65
- Verteilte Anforderungen
 - DB2 Connect-Unterstützung 12
 - föderierte Datenbanken 12
 - Kompensation 12
 - Standorttransparenz 12
 - Übersicht 12
- Vertrauensstellungen
 - gesicherte Kontexte und gesicherte Verbindungen 45
- Verwaltungsdienstprogramm
 - DB2 Connect 7
- Verweise
 - mehrere Datenbankeinträge definieren 41
- Verzeichniscacheunterstützung, Konfigurationsparameter
 - DB2 Connect-Optimierung 125
- Verzeichnisse
 - Arbeitsblätter zur Anpassung 40
 - Systemdatenbank
 - aktualisieren 33
 - Werte 33
- Virtual Telecommunications Access Method (VTAM) 53
- Visual Explain
 - Lernprogramm 172
- VM
 - DRDA
 - und DB2 Connect 10
- VSE
 - DRDA 10

W

- Web-Server
 - DB2 Connect 23
- Webanwendungen
 - DB2 Connect 19
 - gespeicherte Prozeduren 23
- WebSphere
 - Übersicht 21
- WebSphere Federation Server
 - Übersicht 8
- WebSphere MQ
 - DB2 Connect 119
- Werte des DCS-Verzeichnisses 35
- Windows-Betriebssysteme
 - Performance Monitor
 - Überwachung von DB2-Anwendungen 75

X

- X/Open-Modell der verteilten Transaktionsverarbeitung (DTP) 27
- XA
 - gesicherte Verbindungen 45
 - Konzentrator, Beispiele 113
 - Ressourcenmanager 27
- XA-Transaktionsmanager
 - Beschreibung 27
 - Verbindungskonzentratoren 113

Z

z/OS

DRDA 10

Zeichendatentyp 133

Zeitzone

Übersicht 35

Zieldatenbanken

Name 35, 40

Zu diesem Handbuch v

Zuordnung

SQLCODEs 69

NOMAP, Parameter 69

Zuordnung von SQLCODEs

Anpassung 69

NOMAP, Parameter 69

Zusätzliche Abfrageblöcke

EXTRA BLOCKS SRV, Parameter 130

Übersicht 130

Zweiphasiges Commit

aktivieren 59

Resynchronisationsport für TCP/IP-Verbindungen 34

Zwischengespeicherte Adressliste 122



SC12-3925-01

