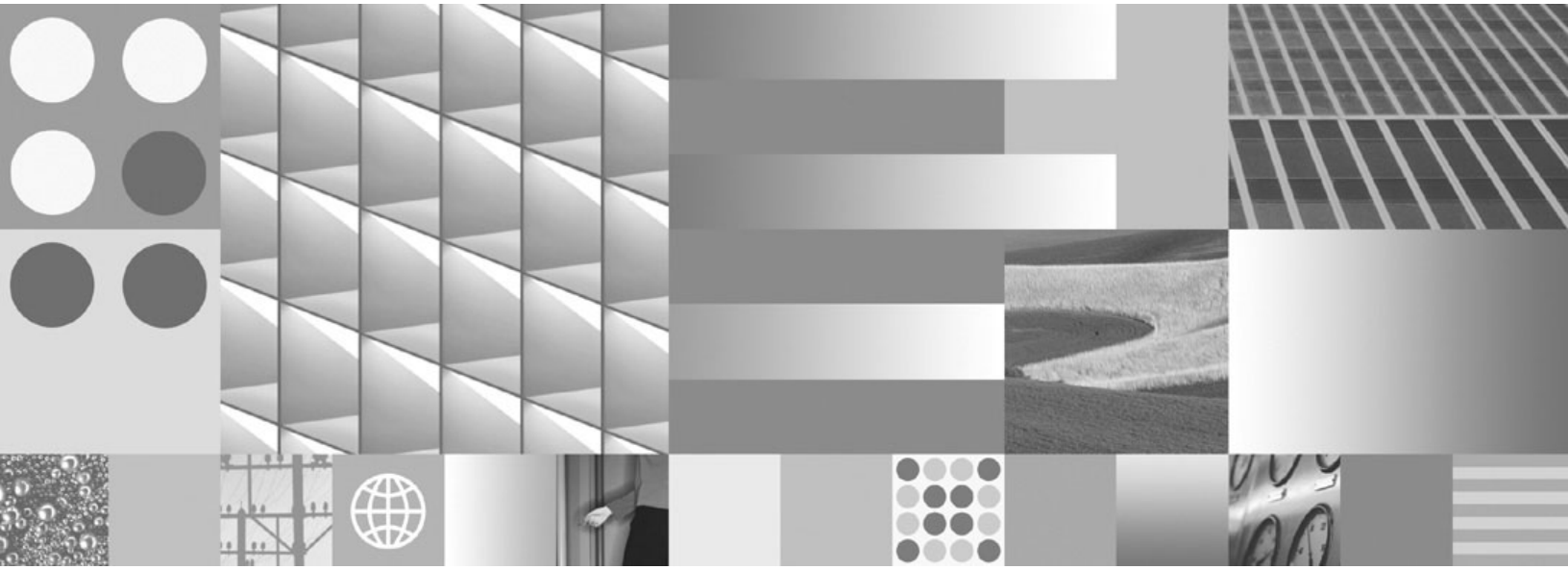
DB2 Version 9.5
für Linux, UNIX und Windows

Version 9 Release 5



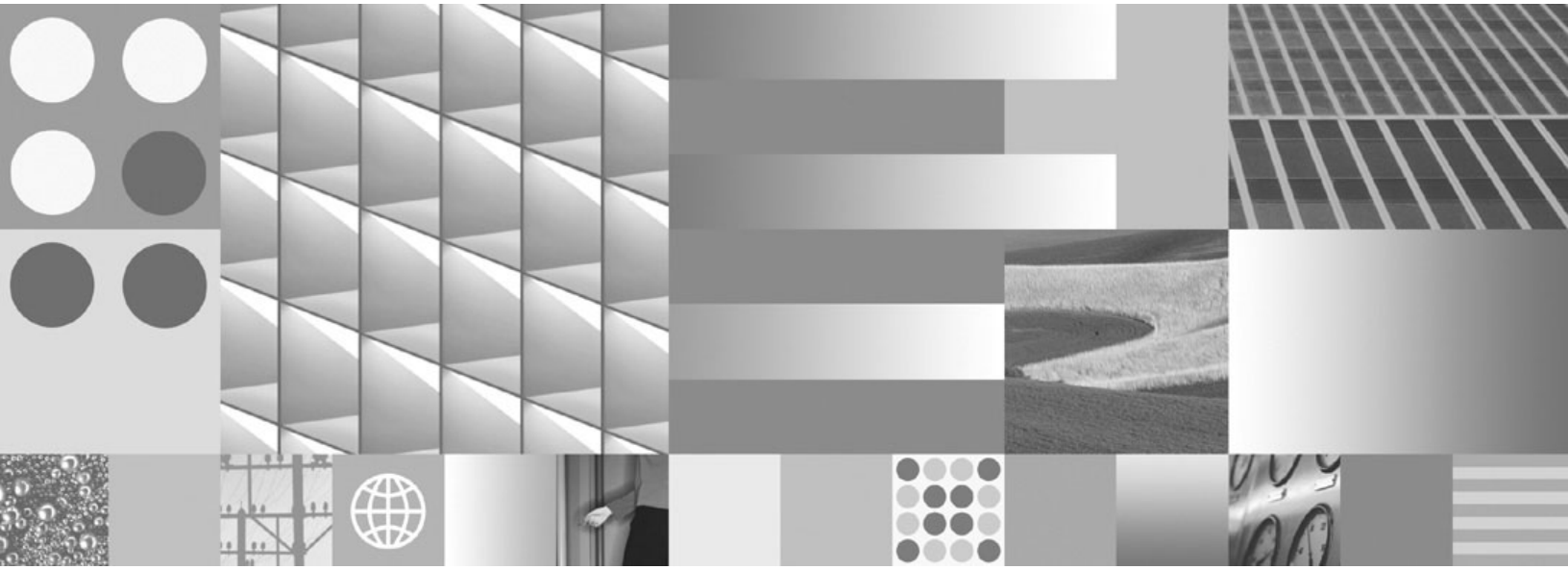
Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz

Aktualisierung: Dezember 2010

SC12-3919-03

DB2 Version 9.5
für Linux, UNIX und Windows

Version 9 Release 5



Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz

Aktualisierung: Dezember 2010

SC12-3919-03

Hinweis

Vor Verwendung dieser Informationen und des darin beschriebenen Produkts sollten die allgemeinen Informationen in Anhang B, „Bemerkungen", auf Seite 431 gelesen werden.

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Version 9.5 for Linux, UNIX and Windows, Data Recovery and High Availability Guide and Reference,
IBM Form SC23-5848-03,
herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 2001, 2010
© Copyright IBM Deutschland GmbH 2010

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
Dezember 2010

Inhaltsverzeichnis

Zu diesem Handbuch vii

Teil 1. Hohe Verfügbarkeit 1

Kapitel 1. Strategien für hohe Verfügbarkeit 3

Hohe Verfügbarkeit durch Redundanz 3
Hohe Verfügbarkeit durch Funktionsübernahme (Failover) 4
Hohe Verfügbarkeit durch Clustering 5
Datenbankprotokollierung 5
Umlaufprotokollierung 6
Archivprotokollierung 7
Protokollsteuerdateien 8

Kapitel 2. Hohe Verfügbarkeit mit DB2-Datenbankprodukten 9

Automatische Clientweiterleitung - Übersicht . . . 9
DB2-Fehlermonitorfunktion für Linux und UNIX . . 10
HADR (High Availability Disaster Recovery) . . . 11
DB2 High Availability Feature 12
Hohe Verfügbarkeit durch Protokollübertragung . . 13
Spiegeln von Protokollen 14
Hohe Verfügbarkeit durch Unterstützung der ausgesetzten E/A und der Onlineteilung einer Spiegeldatenbank. 15

Kapitel 3. Konfiguration für hohe Verfügbarkeit 17

Automatische Clientweiterleitung - Beschreibung und Einrichtung 17
Konfigurieren der automatischen Clientweiterleitung für die Distributortechnologie für die Clientverbindung 19
Angeben eines alternativen Servers für die automatische Clientweiterleitung. 21
Automatische Clientweiterleitung - Einschränkungen 21
DB2-Registrierungsdatei für Fehlermonitore . . . 24
DB2-Fehlermonitor mit dem Befehl db2fm konfigurieren 25
DB2-Fehlermonitor mit db2fmc und Systembefehlen konfigurieren 26
Initialisieren von High Availability Disaster Recovery (HADR) 27
Automatische Clientweiterleitung und HADR (High Availability Disaster Recovery) konfigurieren 29
Indexprotokollierung und High Availability Disaster Recovery (HADR) 30
Datenbankkonfiguration für HADR (High Availability Disaster Recovery) 31
Konfiguration der Protokollarchivierung für DB2 High Availability Disaster Recovery (HADR) . . 39
HADR-Leistung 40
Cluster-Manager und HADR (High Availability Disaster Recovery) 43
Initialisieren einer Bereitschaftsdatenbank . . . 44
Synchronisationsmodus von DB2 HADR (High Availability Disaster Recovery) konfigurieren . . 46
Unterstützung für High Availability Disaster Recovery (HADR) 49
Terminieren von Verwaltungs- und Wartungsaktivitäten für hohe Verfügbarkeit. 55
Erfassen von Informationen zur Richtlinie für automatische Verwaltung mit SYSPROC.AUTOMAINT_GET_POLICY oder SYSPROC.AUTOMAINT_GET_POLICYFILE 56
Konfigurieren einer Richtlinie für automatische Verwaltung mit SYSPROC.AUTOMAINT_SET_POLICY oder SYSPROC.AUTOMAINT_SET_POLICYFILE 56
Konfigurieren der Optionen zur Datenbankprotokollierung 58
Konfigurationsparameter für die Datenbankprotokollierung 59
Verringern der Protokollieraktivitäten mit dem Parameter NOT LOGGED INITIALLY 68
Blockieren von Transaktionen bei vollem Protokollverzeichnis 70
Protokolldateiverwaltung durch Protokollarchivierung. 70
Konfigurieren der Datenbankprotokollierung ohne Dateisystemcaching 73
Konfigurieren einer Clusterumgebung für hohe Verfügbarkeit 74
Cluster-Manager-Integration mit DB2 High Availability Feature 75
Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm 75
Automatisches Konfigurieren eines Clusters mit DB2 High Availability Feature 96
Konfigurieren einer Clusterumgebung mit db2haicu 97
DB2-Cluster-Manager-API 140
Unterstützte Cluster-Management-Software . . 140
Synchronisieren der Systemzeit in einer Umgebung mit partitionierten Datenbanken 161
Konvertieren von Client/Server-Zeitmarken . . 162

Kapitel 4. Verwaltung und Pflege einer Hochverfügbarkeitslösung 163

Protokolldateiverwaltung 163
Bedarfsgesteuerte Protokollarchivierung . . . 165
Protokollarchivierung mit db2tapemgr 165
Archivierung und Abruf von Protokolldateien mithilfe von Benutzerexitprogrammen automatisieren 167

Zuordnen und Entfernen von Protokolldateien 171
Einschließen von Protokolldateien in ein Backup-Image 173
Zufälligen Verlust von Protokolldateien verhindern 175
Beeinträchtigung der Verfügbarkeit durch Verwaltungs- und Wartungsaktivitäten minimieren . . . 176
Stoppen von DB2 High Availability Disaster Recovery (HADR). 176
Aktivierung und Inaktivierung von Datenbanken in einer DB2-HADR-Umgebung. 178
Ausführen von schrittweisen Aktualisierungen und Upgrades in einer DB2-HADR-Umgebung . 179
Verwenden einer geteilten Spiegeldatenbank zum Klonen einer Datenbank 184
Szenario: Ändern der Systemuhr 185
Synchronisation der Primär- und der Bereitschaftsdatenbank 186
Replizierte Operationen von DB2 High Availability Disaster Recovery (HADR) 187
Nicht replizierte Operationen von DB2 High Availability Disaster Recovery (HADR). . . . 188
Status von DB2-HADR-Bereitschaftsdatenbanken (High Availability Disaster Recovery) . . . 190
Bestimmen des Status von HADR-Bereitschaftsdatenbanken mit dem Befehl GET SNAPSHOT . 193
Verwaltung von DB2 High Availability Disaster Recovery (HADR). 194
DB2-HADR-Befehle 194

Kapitel 5. Feststellen und Handhaben von Systemausfällen in einer Hochverfügbarkeitslösung 199
Protokoll mit Benachrichtigungen für die Systemverwaltung 200
Feststellen einer ungeplanten Betriebsunterbrechung 200
Überwachen von HADR (High Availability Disaster Recovery) 201
Handhabung einer ungeplanten Betriebsunterbrechung 201
Automatische Clientweiterleitung - Beispiele 202
Ausführen einer HADR-Funktionsübernahmeoperation. 204
Tauschen von Datenbankrollen bei High Availability Disaster Recovery (HADR) 207
Reintegrieren einer Datenbank nach einer Übernahmeoperation. 209

Teil 2. Datenrecovery 211

Kapitel 6. Entwickeln einer Backup- und Recoverystrategie 213
Häufigkeit von Backups 216
Aspekte des Speicherbedarfs für Recovery. . . . 218
Backupkomprimierung 219
Zusammenhalten zusammengehöriger Daten . . . 219
Backup- und Restoreoperationen zwischen unterschiedlichen Betriebssystemen und Hardwareplattformen 220

Kapitel 7. Datei des Recoveryprotokolls 223
Eintragsstatus der Datei des Recoveryprotokolls 224
Einträge in der Datei des Recoveryprotokolls mithilfe der Verwaltungssicht DB_HISTORY anzeigen . 228
Bereinigen der Datei des Recoveryprotokolls . . . 230
Bereinigen der Datei des Recoveryprotokolls automatisieren 231
Einträge in der Datei des Recoveryprotokolls vor der Bereinigung schützen 233

Kapitel 8. Verwalten von Recoveryobjekten. 235
Protokollfolgenummern 235
Obergrenzen für Protokollfolgenummern . . . 235
Löschen von Datenbankrecoveryobjekten mit dem Befehl PRUNE HISTORY oder der API db2Prune . 242
Automatisieren der Verwaltung von Datenbankrecoveryobjekten 243
Schutz von Recoveryobjekten vor dem Löschen 244
Verwalten von Momentaufnahmebackup-Objekten 245

Kapitel 9. Überwachen des Fortschritts von Backup, Restore- und Recoveryoperationen 247
Statuswerte für Tabellenbereiche 248

Kapitel 10. Backup - Übersicht 249
Verwenden von Backup 252
Durchführen eines Momentaufnahmebackups 254
Verwenden einer geteilten Spiegeldatenbank als Backup-Image 255
Backup auf Band 256
Backup in benannten Pipes. 258
Backup von partitionierten Datenbanken 259
Backup partitionierter Tabellen mit IBM Tivoli Space Manager Hierarchical Storage Management 260
Aktivieren des automatischen Backups 261
Automatisches Datenbank-Backup 261
Optimieren der Leistung des Backup-Dienstprogramms 262
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von Backup 264
Kompatibilität von Online-Backup und anderen Dienstprogrammen 264
Backup - Beispiele. 266

Kapitel 11. Recovery - Übersicht . . . 267
Recovery von Daten 267
Datenrecovery mit 'db2adutl' 268
Recovery einer gelöschten Tabelle 278
Recovery nach Systemabsturz 281
Recovery beschädigter Tabellenbereiche . . . 282

Recovery von Tabellenbereichen in wiederherstellbaren Datenbanken 283
Recovery von Tabellenbereichen in nicht wiederherstellbaren Datenbanken 283
Begrenzen der Auswirkungen von Datenträgerausfällen. 284
Begrenzen der Auswirkungen von Transaktionsfehlern 287
Recovery nach Transaktionsfehlern in einer Umgebung mit partitionierten Datenbanken . . . 287
Recovery nach dem Ausfall eines Datenbankpartitionsservers 291
Wiederherstellen von unbestätigten Transaktionen auf Mainframe- oder Mid-Range-Servern. . 291
Recovery nach einem Katastrophenfall 293
Versionsrecovery 294
Aktualisierende Recovery 295
Inkrementelles Backup und inkrementelle Recovery 298
Restore von inkrementellen Backup-Images . . 300
Automatischer inkrementeller Restore - Einschränkungen 302
Optimieren der Recoveryleistung. 303
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung des Recoverydienstprogramms. . 305

Kapitel 12. Restore - Übersicht 307
Verwenden von Restore 307
Restore für ein Momentaufnahmebackup-Image 309
Restore in einer vorhandenen Datenbank . . . 310
Restore in einer neuen Datenbank 311
Verwenden des inkrementellen Restores in einer Test- und Produktionsumgebung 312
Ausführen einer umgeleiteten Restoreoperation . . 314
Erneutes Definieren von Tabellenbereichscontainern durch Restore einer Datenbank mithilfe eines automatisch generierten Scripts 315
Ausführen eines umgeleiteten Restores mithilfe eines automatisch generierten Scripts 317
Erneutes Erstellen einer Datenbank 318
Erneutes Erstellen und Tabellenbereichscontainer 324
Erneutes Erstellen und Tabellenbereiche für temporäre Tabellen. 324
Auswählen eines Zielimages für die erneute Erstellung einer Datenbank 325
Erneutes Erstellen ausgewählter Tabellenbereiche 328
Erneutes Erstellen mit Images inkrementeller Backups 330
Erneutes Erstellen von partitionierten Datenbanken. 330
Einschränkungen für das erneute Erstellen von Datenbanken 332
Optimieren der Leistung des Restoredienstprogramms 332
Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von Restore 333
Restore - Beispiele. 333
Sitzungen für umgeleiteten Restore - CLP-Beispiele 333
Sitzungen zur erneuten Erstellung - Beispiele für CLP 335

Kapitel 13. Aktualisierende Recovery - Übersicht 347
Verwenden der aktualisierenden Recovery. 349
Aktualisierende Recovery von Änderungen in einem Tabellenbereich 350
Erforderliche Berechtigungen für aktualisierende Recovery 355
Sitzungen zur aktualisierenden Recovery - Beispiele für CLP 356

Kapitel 14. Datenrecovery mit IBM Tivoli Storage Manager (TSM) 361
Konfigurieren eines Tivoli Storage Manager-Clients 361
Aspekte der Verwendung von Tivoli Storage Manager 364

Kapitel 15. DB2 Advanced Copy Services (ACS) 365
Aktivieren von DB2 ACS 365
Installieren von DB2 ACS 366
Aktivieren von DB2 ACS 367
Konfigurieren von DB2 Advanced Copy Services (ACS). 368
DB2 ACS-Konfigurationsscript setup.sh. 370
DB2 ACS-API 371
DB2 ACS-API-Funktionen 371
Datenstrukturen der DB2 ACS-API 398
DB2 ACS - Best Practices 412
DB2 ACS - Einschränkungen 412
Rückkehrcodes der DB2 ACS-API 413
DB2 Advanced Copy Services (ACS) - Unterstützte Betriebssysteme und Hardware 415

Teil 3. Anhänge und Schlussteil 417

Anhang A. Übersicht über die technischen Informationen zu DB2 419
Bibliothek mit technischen Informationen zu DB2 im Hardcopy- oder PDF-Format 420
Bestellen gedruckter DB2-Bücher 422
Aufrufen der Hilfe für den SQL-Status über den Befehlszeilenprozessor 423
Zugriff auf verschiedene Versionen der DB2-Informationszentrale 424
Anzeigen von Themen in der gewünschten Sprache in der DB2-Informationszentrale 424
Aktualisieren der auf Ihrem Computer oder Intranet-Server installierten DB2-Informationszentrale . 425
DB2-Lernprogramme 427
Informationen zur Fehlerbehebung in DB2 . . . 428
Bedingungen 429

Anhang B. Bemerkungen 431

Index 435

Zu diesem Handbuch

In der vorliegenden Veröffentlichung, 'Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz', wird beschrieben, wie Sie hohe Verfügbarkeit für Ihre DB2 Database für Linux®, UNIX® und Windows®-Datenbanklösungen gewährleisten und Datenverlust vermeiden.

'Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz' besteht aus zwei Teilen:

- Teil 1 enthält Informationen zur Hochverfügbarkeit und beschreibt Strategien sowie DB2-Datenbankkomponenten und -funktionen, die Sie bei der Aufrechterhaltung der hohen Verfügbarkeit Ihrer Datenbanklösungen unterstützen.
- Teil 2 enthält Informationen zur Datenrecovery und beschreibt, wie Sie die DB2-Backup- und -Restorefunktionen einsetzen können, um Datenverlust zu vermeiden.

Teil 1. Hohe Verfügbarkeit

Die Verfügbarkeit einer Datenbanklösung ist ein Maß dafür, wie erfolgreich die Benutzeranwendungen die erforderlichen Datenbanktasks ausführen. Können Benutzeranwendungen keine Verbindung zur Datenbank herstellen oder schlagen ihre Transaktionen aufgrund von Fehlern oder Zeitlimitüberschreitungen durch zu hohe Arbeitslast des Systems fehl, ist die Datenbanklösung nicht sehr verfügbar. Können die Benutzeranwendungen aber erfolgreich Verbindungen zur Datenbank herstellen und ihre Aufgaben abarbeiten, weist die Datenbanklösung eine hohe Verfügbarkeit auf.

Der Entwurf einer hoch verfügbaren Datenbanklösung oder die Steigerung der Verfügbarkeit einer bereits vorhandenen Lösung setzt ein Verständnis der Erfordernisse jener Anwendungen voraus, die auf die Datenbank zugreifen. Wenn Sie den größtmöglichen Vorteil aus den Kosten für zusätzlichen Speicherplatz, schnellere Prozessoren oder weitere Softwarelizenzen erzielen möchten, empfiehlt es sich, Ihre Datenbanklösung insbesondere für die wichtigsten Anwendungen in Ihrem Geschäft so verfügbar wie möglich in den Zeiträumen zu gestalten, in denen die Anwendungen am stärksten darauf angewiesen sind.

Ungeplante Betriebsunterbrechungen

Zu den unerwarteten Systemausfällen, die die Verfügbarkeit Ihrer Datenbanklösung für den Benutzer beeinträchtigen könnten, gehören z. B. Stromausfälle, Netzunterbrechungen, Hardwarefehler, Fehler im Betriebssystem oder anderer Software sowie vollständige Systemausfälle im Katastrophenfall. Wenn ein solcher Ausfall zu einem Zeitpunkt auftritt, zu dem der Benutzer erwartet, mit der Datenbank arbeiten zu können, hat eine hoch verfügbare Datenbanklösung die folgenden Aufgaben:

- Abschirmen der Benutzeranwendungen vor dem Ausfall, sodass diese den Ausfall nicht bemerken. Beispiel: DB2 Data Server kann Datenbankclientverbindungen an alternative Datenbankserver weiterleiten, falls ein Datenbankserver ausfällt.
- Reagieren auf den Ausfall, um seine Auswirkungen einzudämmen. Beispiel: Falls auf einer Maschine eines Clusters ein Fehler auftritt, kann der Cluster-Manager diese Maschine aus dem Cluster entfernen, sodass keine weiteren Transaktionen zur Verarbeitung an die fehlerhafte Maschine weitergeleitet werden.
- Recovery nach dem Ausfall durchführen und das System wieder in den normalen Betriebszustand zurückführen. Beispiel: Eine Bereitschaftsdatenbank übernimmt ersatzweise die Datenbankoperationen einer fehlgeschlagenen Primärdatenbank, sodass die fehlgeschlagene Datenbank erneut starten, eine Recovery durchführen und wieder die Primärdatenbankfunktion übernehmen kann.

Diese drei Tasks dürfen mit nur minimalem Effekt auf die Verfügbarkeit der Lösung für die Benutzeranwendungen erfüllt werden.

Geplante Betriebsunterbrechung

Analog müssen in einer hoch verfügbaren Datenbanklösung die Auswirkungen von Verwaltungsaktivitäten auf die Verfügbarkeit der Datenbank für die Benutzeranwendungen minimiert werden.

Beispiel: Falls die Datenbanklösung für eine typische Ladenfront mit Geschäftszeiten zwischen 9.00 und 17.00 Uhr eingesetzt wird, können die Verwaltungsaktivitäten offline und außerhalb der Geschäftszeiten stattfinden, ohne die Verfügbarkeit der Datenbank für die Benutzeranwendungen zu beeinträchtigen. Falls die Datenbanklösung im Online-Bankinggeschäft eingesetzt wird und den Kunden 24 Stunden am Tag für den Zugriff über das Internet zur Verfügung stehen soll, müssen die Verwaltungsaktivitäten online ausgeführt oder für Aktivitätsperioden mit geringe Systemauslastung terminiert werden, damit sie eine möglichst geringe Auswirkung auf die Verfügbarkeit der Datenbank für die Kunden haben.

Die folgenden beiden Faktoren sind gegeneinander abzuwägen, wenn Sie Geschäftsentscheidungen und Entwurfsmöglichkeiten im Hinblick auf die Verfügbarkeit Ihrer Datenbanklösung in Betracht ziehen:

- Die Kosten, die Ihnen in Ihren Geschäftsaktivitäten entstehen, wenn die Datenbank den Kunden nicht zur Verfügung steht.
- Der Aufwand, der durch die Implementierung eines bestimmten Verfügbarkeitsgrades entsteht.

Beispiel: Ein internetbasiertes Geschäft erwirtschaftet pro Stunde, in der seine Datenbanklösung Kunden bedient, den Ertrag X. Eine Hochverfügbarkeitsstrategie, mit der jährlich zehn Stunden Ausfallzeit gespart werden, erbringt dem Geschäft pro Jahr den zusätzlichen Ertrag von zehn mal X. Wenn die Kosten der Implementierung dieser Hochverfügbarkeitsstrategie geringer sind, als der erwartete Zusatzertrag, ist die Implementierung lohnenswert.

Kapitel 1. Strategien für hohe Verfügbarkeit

Einem Benutzer ist es egal, warum seine Datenbankanforderung fehlgeschlagen ist. Ob bei einer Transaktion das Zeitlimit wegen schlechten Leistungsverhaltens überschritten wurde, eine Komponente innerhalb der Lösung fehlgeschlagen ist oder ein Administrator die Datenbank in den Offlinestatus wegen Instandhaltungsarbeiten versetzt hat, das Ergebnis ist für den Benutzer das gleiche: die Datenbank steht nicht für die Verarbeitung von Anforderungen zur Verfügung.

Die folgenden Strategien tragen zur Verbesserung der Verfügbarkeit Ihrer Datenbanklösung bei:

Redundanz
Sie verfügen über Zweitkopien aller Komponenten in Ihrer Datenbanklösung, die im Falle einer Störung oder eines Fehlers Workload übernehmen können.

Systemüberwachung
Es werden Statistikdaten über die Komponenten in Ihrer Datenbanklösung erfasst, um den Lastausgleich zu erleichtern bzw. um festzustellen, ob Komponenten eventuell ausgefallen sind.

Lastausgleich
Ein Teil der Verarbeitungsprozesse einer überlasteten Komponente in Ihrer Datenbanklösung wird an eine andere Komponente in Ihrer Datenbanklösung übertragen, die zum aktuellen Zeitpunkt weniger Arbeitslast aufweist.

Funktionsübernahme
Alle Verarbeitungsprozesse einer fehlgeschlagenen Komponente in Ihrer Datenbanklösung werden auf eine sekundäre Komponente übertragen.

Leistung maximieren
Die Möglichkeit, dass Transaktionen sehr viel Zeit zum Beenden benötigen oder das Zeitlimit überschreiten, wird verringert.

Beeinträchtigungen durch Verwaltungsaktivitäten minimieren
Terminierung automatisierter und manueller Verwaltungsaktivitäten, um die Benutzeranwendungen so wenig wie möglich zu beeinträchtigen.

Hohe Verfügbarkeit durch Redundanz

Wenn irgendein Element oder eine Komponente Ihrer Datenbanklösung ausfällt, sei es die Stromversorgung oder das Netzverbindungskabel, das Betriebssystem oder die Anwendungssoftware, ist das Endergebnis dasselbe: Ihre Datenbanklösung steht den Benutzeranwendungen nicht zur Verfügung. Eine wichtige Strategie zur Aufrechterhaltung der hohen Verfügbarkeit besteht darin, über redundante Systeme zu verfügen, sodass für den Fall, dass eine Komponente ausfällt, eine sekundäre oder eine Backup-Kopie der ausgefallenen Komponente die Operationen übernehmen kann, wodurch die Datenbank den Benutzeranwendungen weiterhin zur Verfügung stehen kann.

Redundanz ist im Systemaufbau in folgenden Bereichen üblich:

- Nicht unterbrochene oder Notstromversorgung

- Verwendung mehrerer Netzverbindungsleitungen zwischen den einzelnen Komponenten
- Bonding oder Lastausgleich für Netzkarten
- Verwenden mehrerer Festplattenlaufwerke in einem redundanten Array
- Verwenden von CPU-Clustern

Wenn eine dieser Komponenten des Systems nicht redundant ist, könnte sich die betreffende Komponente als SPoF für das gesamte System erweisen.

Sie können Redundanz auf Datenbankebene erstellen, indem Sie zwei Datenbanken betreiben: eine primäre Datenbank, die im Normalfall alle oder die Mehrzahl der Verarbeitungsprozesse der Anwendungen verarbeitet, und eine sekundäre Datenbank, die die Verarbeitungsprozess übernehmen kann, wenn die primäre Datenbank ausfällt. In einer DB2-HADR-Umgebung (High Availability Disaster Recovery) wird die sekundäre Datenbank als "Bereitschaftsdatenbank" bezeichnet.

Hohe Verfügbarkeit durch Funktionsübernahme (Failover)

Die Funktionsübernahme (Failover) ist die Übertragung von Workload von einem primären System auf ein sekundäres System, wenn das primäre System ausfällt. Wenn Workload auf diese Weise übertragen wurde, lautet die Bezeichnung dafür auch, dass das sekundäre System die Verarbeitungsprozesse (Workload) des fehlgeschlagenen primären Systems "übernommen" hat.

Beispiel 1

Schlägt in einer Clusterumgebung eine der Maschinen im Cluster fehl, kann die Software für das Cluster-Management Prozesse, die auf der fehlgeschlagenen Maschine ausgeführt wurden, auf eine andere Maschine im Cluster versetzen.

Beispiel 2

Ist in einer Datenbanklösung mit mehreren IBM® Data Server-Servern eine Datenbank nicht mehr verfügbar, kann der Datenbankmanager Anwendungen, die mit dem nicht mehr verfügbaren Datenbankserver verbunden waren, an einen sekundären Datenbankserver weiterleiten.

Die beiden häufigsten im Markt vorhandenen Funktionsübernahmestrategien sind der Bereitschaftsmodus und die gegenseitige Übernahme:

Bereitschaftsmodus (Idle Standby)

In dieser Konfiguration verarbeitet ein primäres System die gesamte Workload, während ein sekundäres oder Bereitschaftsmodussystem sich im Bereitschaftsmodus befindet und bereit ist, diese Workload zu übernehmen, falls das primäre System ausfällt.

Gegenseitige Übernahme (Mutual Takeover)

In dieser Konfiguration gibt es mehrere Systeme, die jeweils das designierte sekundäre System für ein anderes System sind. Wenn ein System fehlschlägt, wirkt sich dies negativ auf die Gesamtleistung aus, da das sekundäre System des ausgefallenen Systems nicht nur weiter seine eigene Workload, sondern auch die des ausgefallenen Systems verarbeiten muss.

Hohe Verfügbarkeit durch Clustering

Ein Cluster ist eine Gruppe verbundener Systeme, die wie ein einziges System zusammenarbeiten. Wenn eine Maschine in einem Cluster fehlschlägt, transferiert eine Clusterverwaltungssoftware die Verarbeitungsprozesse des fehlgeschlagenen Systems auf andere Systeme.

Überwachungssignalfunktion

Die Software für die Funktionsübernahme kann zur Feststellung eines Fehlers auf einem System des Clusters zwischen den Systemen die Überwachungssignalfunktion oder Keepalive-Pakete verwenden, um die Verfügbarkeit zu bestätigen. Die Überwachungssignalfunktion bezieht Systemservices ein, die die konstante Kommunikation zwischen allen Systemen eines Clusters verwalten. Wenn ein Überwachungssignal nicht ermittelt werden kann, wird die Funktionsübernahme durch ein Backup-System eingeleitet.

IP-Adressübernahme

Bei einer Störung auf einem System im Cluster können die Cluster-Manager Workload von einem System auf ein anderes übertragen, indem die IP-Adresse von einem System auf das andere übertragen wird. Dieser Vorgang wird als "IP-Adressübernahme" oder "IP-Übernahme" bezeichnet. Die Übertragung ist für Clientanwendungen nicht sichtbar, die auch weiterhin die ursprüngliche IP-Adresse verwenden und nicht bemerken, dass die physische Maschine, zu der die IP-Adresse gehört, sich geändert hat.

DB2 High Availability Feature ermöglicht die Integration zwischen IBM Data Server und Cluster-Management-Software.

Datenbankprotokollierung

Die Datenbankprotokollierung ist ein wichtiger Teil Ihres Lösungsentwurfs für eine Datenbank mit hoher Verfügbarkeit, da man anhand von Datenbankprotokollen nach einem Fehler eine Recovery durchführen sowie primäre und sekundäre Datenbanken synchronisieren kann.

Jeder Datenbank sind Protokolldateien zugeordnet. In diesen Protokolldateien werden die Datenbankänderungen aufgezeichnet. Wenn eine Datenbank über den Stand des letzten vollständigen Offline-Backups hinaus wiederhergestellt werden muss, sind Protokolle erforderlich, um die Datenbank bis zu dem Zeitpunkt des Fehlers aktualisierend wiederherstellen zu können.

Es werden zwei Typen der DB2-Datenbankprotokollierung unterstützt: *Umlaufprotokollierung* und *Archivprotokollierung*. Jeder Typ stellt eine andere Stufe der Wiederherstellbarkeit bereit:

- „Umlaufprotokollierung" auf Seite 6
- „Archivprotokollierung" auf Seite 7

Der Vorteil der Archivprotokollierung liegt darin, dass von der aktualisierenden Recovery sowohl archivierte als auch aktive Protokolldateien verwendet werden können, um eine Datenbank entweder bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt wiederherzustellen.

Die Archivprotokolldateien können zur Recovery von Änderungen verwendet werden, die nach dem Backup vorgenommen wurden. Bei der Umlaufprotokollierung hingegen ist eine Recovery nur bis zum Zeitpunkt des letzten Backups möglich, alle später vorgenommenen Änderungen gehen verloren.

Umlaufprotokollierung

Umlaufprotokollierung ist die Standardprotokollierung, wenn eine neue Datenbank erstellt wird. (Die Datenbankkonfigurationsparameter *logarchmeth1* und *logarchmeth2* sind auf `OFF` gesetzt.) Bei dieser Art der Protokollierung sind nur vollständige Offline-Backups der Datenbank zulässig. Die Datenbank muss offline sein (d. h. für Benutzer nicht zugänglich), wenn ein Gesamtbackup erstellt wird.

Wie der Name bereits andeutet, verwendet die Umlaufprotokollierung einen „Ring" von Onlineprotokollen, die eine Recovery nach Transaktionsfehlern oder Systemabstürzen ermöglichen. Die Protokolle werden nur so lange verwendet und behalten, wie es erforderlich ist, um die Integrität der aktuellen Transaktionen zu gewährleisten. Bei der Umlaufprotokollierung ist es nicht möglich, eine Datenbank durch frühere Transaktionen, die seit dem letzten Gesamtbackup durchgeführt wurden, aktualisierend wiederherzustellen. Alle Änderungen, die seit dem letzten Backup vorgenommen wurden, gehen verloren. Da dieser Typ des Restores die Daten auf dem Stand zum Zeitpunkt des Gesamtbackups wiederherstellt, wird er als *Versionsrecovery* bezeichnet.

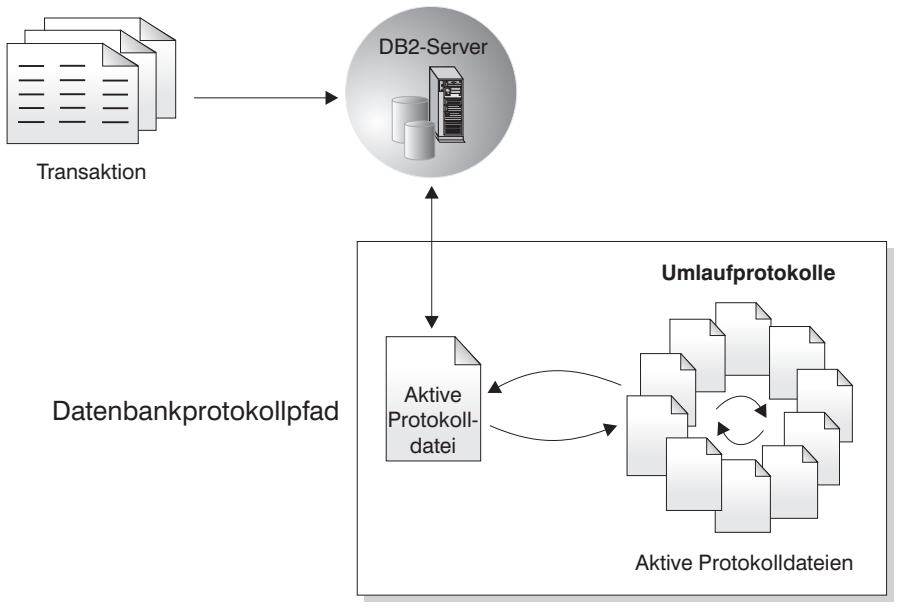


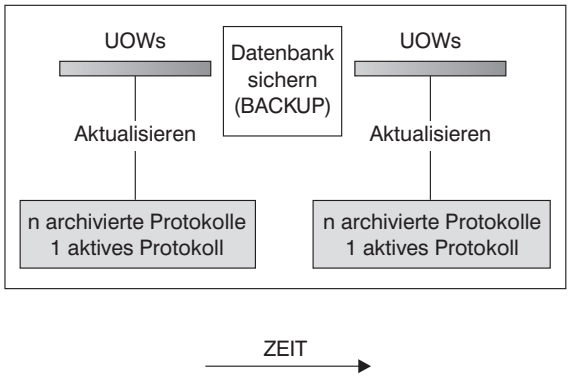
Abbildung 1. Umlaufprotokollierung

Aktive Protokolldateien werden bei der Recovery nach einem Systemabsturz verwendet, um zu verhindern, dass eine Datenbank nach einer Störung (Stromausfall oder Anwendungsfehler) in einem inkonsistenten Status zurückbleibt. Aktive Protokolldateien befinden sich im Verzeichnispfad der Datenbankprotokolle.

Archivprotokollierung

Archivprotokollierung wird besonders für die aktualisierende Recovery verwendet. Archivprotokolldateien sind Protokolldateien, die aus dem Pfad für aktive Protokolldateien an eine andere Position kopiert wurden.

Sie können nur einen oder beide Datenbankkonfigurationsparameter *logarchmeth1* oder *logarchmeth2* verwenden, damit Sie selbst oder der Datenbankmanager den Protokollarchivierungsprozess verwalten können.



Protokolle werden zwischen Backups verwendet, um die Änderungen an den Datenbanken zu protokollieren.

Abbildung 2. Aktive und archivierte Datenbankprotokolle bei aktualisierender Recovery. Im Falle einer lange laufenden Transaktion kann es mehr als ein aktives Protokoll geben.

Das Erstellen von Online-Backups wird nur dann unterstützt, wenn die Datenbank für Archivprotokollierung konfiguriert ist. Während eines Online-Backups werden alle Aktivitäten an der Datenbank protokolliert. Wenn ein Online-Backup wiederhergestellt wird, muss mithilfe der Protokolle die Datenbank mindestens bis zu dem Zeitpunkt aktualisierend wiederhergestellt werden, zu dem das Backup abgeschlossen wurde. Dazu müssen die Protokolle archiviert worden sein und beim Restore der Datenbank zur Verfügung stehen. Nachdem ein Online-Backup abgeschlossen ist, erzwingt der Datenbankmanager das Schließen der zu dem Zeitpunkt aktiven Protokolldatei, die als Ergebnis archiviert wird. Hierdurch wird sichergestellt, dass Ihrem Online-Backup alle zur Recovery benötigten archivierten Protokolldateien zur Verfügung stehen.

Die Datenbankkonfigurationsparameter *logarchmeth1* und *logarchmeth2* ermöglichen Ihnen, die Speicherposition der Archivprotokolle zu ändern. Der Parameter *logarchmeth2* ermöglicht Ihnen, Protokolldateien an einer zweiten separaten Position zu archivieren. Der Parameter *newlogpath* wirken sich außerdem auf die Speicherposition aus, an der die aktiven Protokolle gespeichert werden.

Wenn Sie nicht zuvor angegeben haben, dass Sie die aktiven Protokolle verwalten möchten (indem Sie den Wert LOGRETAIN) angegeben haben), entfernt der Datenbankmanager Protokolldateien aus dem Pfad für aktive Protokolldateien, nachdem diese Dateien archiviert wurden und nicht mehr für eine Recovery nach Systemabsturz benötigt werden. Wurde die Endlosprotokollierung aktiviert und muss Speicherbereich für weitere aktive Protokolldateien verfügbar sein, benennt der Datenbankserver die Protokolldateien um, nachdem diese archiviert worden sind.

Protokollsteuerdateien

Wenn eine Datenbank nach einem Fehler erneut startet, wendet der Datenbankmanager in Protokolldateien gespeicherte Transaktionsinformationen an, um die Datenbank wieder in einen konsistenten Status zu versetzen. Für die Feststellung, welche Datensätze aus den Protokolldateien auf die Datenbank angewendet werden müssen, verwendet der Datenbankmanager Informationen, die in einer Protokollsteuerdatei aufgezeichnet sind.

Ausfallsicherheit der Datenbank durch Redundanz

Der Datenbankmanager pflegt zwei Kopien der Protokollsteuerdatei mit den Namen SQLOGCTL.LFH.1 und SQLOGCTL.LFH.2, sodass im Falle der Beschädigung einer Kopie der Datenbankmanager noch die andere Kopie verwenden kann.

Leistungsaspekte

Das Anwenden der in den Protokollsteuerdateien enthaltenen Transaktionsinformationen trägt zu dem Systemaufwand beim Neustart einer Datenbank nach einem Fehler bei. Sie können die Frequenz konfigurieren, mit der der Datenbankmanager Transaktionen auf Platte schreibt, um die Anzahl der Protokollsätze zu reduzieren, die während der Recovery nach einem Systemabsturz verarbeitet werden müssen. Verwenden Sie dazu den Konfigurationsparameter „softmax - Vor bedingtem Prüfpunkt zu schreibende Protokollsätze" im Handbuch *Datenserver, Datenbanken und Datenbankobjekte*.

Kapitel 2. Hohe Verfügbarkeit mit DB2-Datenbankprodukten

DB2-Datenbankprodukte für DB2-Server enthalten Funktionen, die eine Vielzahl von Hochverfügbarkeitsstrategien unterstützen.

Automatische Clientweiterleitung - Übersicht

Die automatische Clientweiterleitung ist eine Funktion von IBM Data Server, die Clientanwendungen von einem fehlgeschlagenen Server an einen alternativen Server weiterleitet, sodass die Anwendungen mit nur minimaler Unterbrechung weiter funktionieren können. Die automatische Clientweiterleitung kann nur erfolgen, wenn vor dem Ausfall oder Fehler ein alternativer Server festgelegt worden ist.

Tabelle 1 listet die relevanten Themen in den jeweiligen Kategorien auf.

Tabelle 1. Übersicht über die Informationen zur automatischen Clientweiterleitung

Kategorie	Zugehörige Themen
Allgemeine Informationen	• „Automatische Clientweiterleitung - Einschränkungen" auf Seite 21 • „Automatische Clientweiterleitung - Beschreibung und Einrichtung" auf Seite 17 • „Automatische Clientweiterleitung - Beschreibung und Einrichtung (DB2 Connect)" im Handbuch *DB2 Connect-Server - Einstieg*
Konfiguration	• „Angeben eines alternativen Servers für die automatische Clientweiterleitung" auf Seite 21 • „Konfiguration der Hochverfügbarkeitsunterstützung für Java-Clients bei DB2 Database für Linux, UNIX und Windows" in Developing Java™ Applications • „Konfiguration der Hochverfügbarkeitsunterstützung für Java-Clients bei DB2 Database für Linux, UNIX und Windows" in Call Level Interface Guide and Reference, Volume 1
Beispiele	• „Automatische Clientweiterleitung - Beispiele" auf Seite 202
Interaktion mit anderen DB2-Funktionen	• „Automatische Clientweiterleitung und HADR (High Availability Disaster Recovery) konfigurieren" auf Seite 29 • „Konfiguration der Hochverfügbarkeitsunterstützung für Java-Clients bei DB2 Database für Linux, UNIX und Windows" in Developing Java Applications • „Konfiguration der Hochverfügbarkeitsunterstützung für Java-Clients bei DB2 Database für Linux, UNIX und Windows" in Call Level Interface Guide and Reference, Volume 1
Fehlerbehebung	• „Konfigurieren der automatischen Clientweiterleitung für die Distributortechnologie für die Clientverbindung" auf Seite 19

Anmerkung: Ab Version 9.5 Fixpack 3 steht die automatische Clientweiterleitung für DB2 für z/OS Sysplex in IBM Data Server-Clients und in nicht auf Java basierten IBM Data Server-Treibern zur Verfügung. Durch diese Unterstützung können Anwendungen, die auf ein DB2 für z/OS-Sysplex zugreifen, die vom Client bereitgestellten Funktionen der automatischen Clientweiterleitung verwenden und müssen hierfür keinen DB2 Connect-Server verwenden.

Weitere Informationen zu dieser Funktion enthält der Abschnitt zur automatischen Clientweiterleitung (clientseitig) in der DB2-Informationszentrale.

DB2-Fehlermonitorfunktion für Linux und UNIX

Die nur auf UNIX-basierten Systemen verfügbaren DB2-Fehlermonitorfunktionen sorgen dafür, dass die DB2-Datenbanken betriebsbereit bleiben, u. a. durch das Überwachen von DB2-Datenbankmanagerinstanzen und das erneute Starten von Instanzen, die zu früh beendet werden.

Der FMC ist der Prozess der Fehlermonitorfunktion, der während der UNIX-Startreihenfolge gestartet wird. Der Dämon *init* startet den FMC und startet ihn erneut, falls der FMC abnormal beendet wird. Der FMC startet für jede DB2-Instanz einen Fehlermonitor. Jeder Fehlermonitor wird als Dämonprozess ausgeführt und hat dieselben Benutzerzugriffsrechte wie die DB2-Instanz.

Nach dem Start eines Fehlermonitors wird er überwacht, um sicherzustellen, dass er nicht vorzeitig beendet wird. Wenn ein Fehlermonitor fehlschlägt, wird er vom FMC erneut gestartet. Jeder Fehlermonitor überwacht wiederum eine DB2-Instanz. Wenn die DB2-Instanz vorzeitig beendet wird, wird sie vom Fehlermonitor neu gestartet. Die Fehlermonitorfunktion kann nur durch Absetzen des Befehls db2stop inaktiviert werden. Wenn eine DB2-Instanz auf andere Weise beendet wird, wird sie von der Fehlermonitorfunktion neu gestartet.

DB2-Fehlermonitor - Einschränkungen

Wenn Sie ein Clusteringprodukt mit hoher Verfügbarkeit wie z. B. HACMP, MSCS oder IBM Tivoli System Automation for Multiplatforms verwenden, muss die Fehlermonitorfunktion inaktiviert werden, da der Start und das Herunterfahren der Instanz von dem Clusteringprodukt gesteuert werden.

Unterschiede zwischen dem DB2-Fehlermonitor und dem DB2-Diagnosemonitor

Der Diagnosemonitor und der Fehlermonitor sind Tools, die in einer einzigen Datenbankinstanz aktiv sind. Der Diagnosemonitor verwendet *Diagnoseanzeiger*, um bestimmte Leistungsaspekte des Datenbankmanagers oder von Datenbanken auf ihren ordnungsgemäßen Betrieb hin zu bewerten. Ein Diagnoseanzeiger misst den Status eines Aspekts einer bestimmten Klasse von Datenbankobjekten, wie zum Beispiel eines Tabellenbereichs. Diagnoseanzeiger können anhand spezieller Kriterien bewertet werden, um den Status der betreffenden Klasse von Datenbankobjekt festzustellen. Außerdem können Diagnoseanzeiger Alerts generieren, um Sie zu benachrichtigen, wenn ein Bezugswert einen Schwellenwert überschreitet oder angibt, dass ein Datenbankobjekt sich in einem nicht normalen Zustand befindet.

Demgegenüber ist der Fehlermonitor nur dafür zuständig, die von ihm überwachte Instanz betriebsbereit und aktiv zu halten. Wenn die überwachte DB2-Instanz unerwartet beendet wird, startet der Fehlermonitor die Instanz neu. Die Fehlermonitorfunktion steht unter Windows nicht zur Verfügung.

HADR (High Availability Disaster Recovery)

Die Funktion DB2 Data Server High Availability Disaster Recovery (HADR) ist eine Replikationsfunktion für Datenbanken, die eine Hochverfügbarkeitslösung für Teil- oder Komplettausfälle von Standorten bereitstellt. HADR schützt durch die Replikation von Daten einer Quellendatenbank, der sogenannten Primärdatenbank, auf einer Zieldatenbank, der sogenannten Bereitschaftsdatenbank, vor Datenverlusten.

HADR ist besonders geeignet, wenn Sie die meisten oder alle Ihre Datenbanken schützen müssen oder wenn Sie DDL-Operationen ausführen, die automatisch in die Bereitschaftsdatenbank repliziert werden müssen.

Anwendungen können nur auf die aktuelle Primärdatenbank zugreifen. Aktualisierungen an der Bereitschaftsdatenbank erfolgen durch aktualisierendes Wiederherstellen von Protokolldaten, die auf der Primärdatenbank generiert und an die Bereitschaftsdatenbank übertragen werden.

Ein Teilausfall eines Standorts kann durch einen Ausfall von Hardware, Software (DB2-Datenbanksystem oder Betriebssystem) oder des Netzwerks verursacht werden. Ohne HADR muss bei einem Teilausfall eines Standorts der DBMS-Server (DBMS - Datenbankmanagementsystem) mit der Datenbank erneut gestartet werden. Die für den Neustart der Datenbank und des Servers, auf dem sich die Datenbank befindet, erforderliche Zeit lässt sich nicht vorhersehen. Es kann einige Minuten in Anspruch nehmen, bevor die Datenbank wieder in einen konsistenten Zustand versetzt und verfügbar ist. Mit HADR kann die Bereitschaftsdatenbank innerhalb von Sekunden den Datenbankbetrieb übernehmen. Außerdem können Sie die automatische Clientweiterleitung oder Wiederholungslogik in der Anwendung nutzen, um die Clients, die die ursprüngliche Primärdatenbank verwendeten, an die Bereitschaftsdatenbank (die neue Primärdatenbank) umzuleiten.

Ein kompletter Standortausfall kann auftreten, wenn durch eine Katastrophe wie einen Brand der gesamte Standort zerstört wird. Da HADR zur Kommunikation zwischen der Primärdatenbank und der Bereitschaftsdatenbank TCP/IP verwendet, können sich die Datenbanken an verschiedenen Standorten befinden. Ihre Primärdatenbank könnte sich z. B. in Ihrer Zentrale in einer Stadt befinden, während Ihre Bereitschaftsdatenbank sich in einer Verkaufsniederlassung in einer anderen Stadt befindet. Im Fall eines Fehlers wird auf der primären Site die Datenverfügbarkeit dadurch gewährleistet, dass die ferne Bereitschaftsdatenbank die Rolle der Primärdatenbank mit vollständiger DB2-Funktionalität übernehmen kann. Nach einer Funktionsübernahmeoperation können Sie die ursprüngliche Primärdatenbank wieder online verfügbar machen und in ihren Status als Primärdatenbank zurückversetzen. Dies wird als Zurücksetzung bezeichnet.

Mit HADR können Sie die gewünschte Ebene für den Schutz vor potenziellem Datenverlust auswählen, indem Sie einen der drei Synchronisationsmodi angeben: synchron (SYNC), fast synchron (NEARSYNC) oder asynchron (ASYNC).

Wenn der ausgefallene primäre Server wieder in Stand gesetzt ist, kann er dem HADR-Paar als Bereitschaftsdatenbank hinzugefügt werden, sofern eine Konsistenz der beiden Datenbankkopien erzielt werden kann. Nachdem die ursprüngliche Primärdatenbank dem HADR-Paar als Bereitschaftsdatenbank hinzugefügt wurde, können Sie die Rollen der Datenbanken wieder tauschen, sodass die ursprüngliche Primärdatenbank erneut als aktuelle Primärdatenbank genutzt werden kann.

HADR ist nur eine von verschiedenen Replikationslösungen, die mit der DB2-Produktfamilie angeboten werden. WebSphere Information Integrator und das DB2-

Datenbanksystem enthalten SQL Replication- und Q Replication-Lösungen, die in einigen Konfigurationen ebenfalls zur Einrichtung einer hohen Verfügbarkeit verwendet werden können. Diese Funktionen verwalten logisch konsistente Kopien von Datenbanktabellen an verschiedenen Standorten. Zusätzlich bieten sie Flexibilität und komplexe Funktionalität, wie Unterstützung für Spalten- und Zeilenfilter, Datenkonvertierung und Aktualisierungen an beliebigen Kopien einer Tabelle, und sie können in Umgebungen mit partitionierten Datenbanken verwendet werden.

DB2 High Availability Feature

DB2 High Availability Feature ermöglicht die Integration zwischen IBM Data Server und Cluster-Management-Software.

Wenn Sie eine Datenbankmanagerinstanz in einer Clusterumgebung stoppen, müssen Sie den Cluster-Manager darüber informieren, dass die Instanz gestoppt wurde. Wenn der Cluster-Manager über das Stoppen der Instanz nicht informiert wird, versucht er möglicherweise, eine Operation wie beispielsweise eine Funktionsübernahme für die gestoppte Instanz auszuführen. DB2 High Availability Feature stellt eine Infrastruktur bereit, die es dem Datenbankmanager ermöglicht, mit dem Cluster-Manager zu kommunizieren, wenn Konfigurationsänderungen bei Instanzen, wie beispielsweise das Stoppen einer Datenbankmanagerinstanz, Clusteränderungen erforderlich machen.

Wenn der Datenbankmanager mit dem Cluster-Manager kommuniziert, sobald Änderungen bei Instanzen Clusteränderungen erforderlich machen, sind nach der Durchführung von Konfigurationsänderungen bei Instanzen keine separaten Clusteroperationen mehr erforderlich.

DB2 High Availability Feature besteht aus den folgenden Elementen:

- IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist in IBM Data Server unter AIX- und Linux-Betriebssystemen als Teil von DB2 High Availability Feature integriert. Verwenden Sie für eine Installation, ein Upgrade oder eine Deinstallation von SA MP Base Component entweder das DB2-Installationsprogramm oder die Scripts installSAM und uninstallSAM, die sich auf dem Installationsdatenträger von IBM Data Server befinden. Unter Windows-Betriebssystemen ist SA MP Base Component Teil des Produktpakets von DB2 High Availability Feature, es ist jedoch nicht in das DB2-Installationsprogramm integriert. Siehe: „Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm" auf Seite 75
- In einer Clusterumgebung erfordern bestimmte Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz entsprechende Änderungen an der Clusterkonfiguration. Mit DB2 High Availability Feature kann der Datenbankmanager automatisch Änderungen an der Konfiguration des Cluster-Managers anfordern, wenn bestimmte Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz ausgeführt werden. Siehe: „Automatisches Konfigurieren eines Clusters mit DB2 High Availability Feature" auf Seite 96
- DB2 High Availability Instance Configuration Utility (db2haicu) ist ein textbasiertes Dienstprogramm zur Konfiguration und Verwaltung hoch verfügbarer Datenbanken in einer Clusterumgebung. db2haicu stellt Abfragen an das System, um Informationen zur Datenbankinstanz, zur Clusterumgebung und zum Cluster-Manager zu sammeln. Zur Angabe weiterer Informationen über Parameter für den Aufruf db2haicu, eine Eingabedatei oder während der Laufzeit können Sie die Eingabeaufforderungen von db2haicu verwenden. Siehe: „db2haicu" auf Seite 104

- Die DB2-Cluster-Manager-API definiert eine Reihe von Funktionen, mit deren Hilfe der Datenbankmanager dem Cluster-Manager Konfigurationsänderungen mitteilen kann. Siehe: „DB2-Cluster-Manager-API" auf Seite 140

Hohe Verfügbarkeit durch Protokollübertragung

Die Protokollübertragung ist das Kopieren ganzer Protokolldateien auf eine Bereitschaftsmaschine, entweder von einer Archivierungseinheit aus oder mithilfe eines Benutzerexitprogramms, das für die primäre Datenbank ausgeführt wird.

Die Bereitschaftsdatenbank wird anhand der von der Produktionsmaschine generierten Protokolldateien kontinuierlich aktualisierend wiederhergestellt. Wenn die Produktionsmaschine ausfällt, findet eine Funktionsübernahme statt. Dabei wird Folgendes ausgeführt:

- Die verbleibenden Protokolle werden auf die Bereitschaftsmaschine übertragen.
- Die Bereitschaftsdatenbank wird bis zum Ende der Protokolle aktualisierend wiederhergestellt und gestoppt.
- Die Clients stellen die Verbindung zur Bereitschaftsdatenbank wieder her und nehmen ihre Operationen wieder auf.

Die Bereitschaftsmaschine verfügt über eigene Ressourcen (d. h. Platten), sie muss jedoch dieselben physischen und logischen Definitionen besitzen wie die Produktionsdatenbank. Mit dieser Methode wird die primäre Datenbank auf der Bereitschaftsmaschine wiederhergestellt, wobei das Restoredienstprogramm oder die Funktion zur Erstellung einer Spiegeldatenbank verwendet wird.

Damit Sie Ihre Datenbank nach einem Katastrophenfall garantiert wiederherstellen können, sollten Sie Folgendes beachten:

- Die Position des Archivs sollte von der Primärstation geographisch getrennt sein.
- Führen Sie eine ferne Spiegelung des Protokolls am Standort der Bereitschaftsdatenbank durch.
- Verwenden Sie einen synchronen Spiegel, bei dem Sie keine Daten verlieren. Sie können hierzu moderne Plattensubsysteme wie ESS und EMC oder eine andere Technologie zum fernen Spiegeln verwenden. Es wird ebenfalls empfohlen, NV-RAM-Cache (lokal und fern) zu verwenden, um die Leistungsauswirkungen einer Recoverysituation nach einem Katastrophenfall zu minimieren.

Anmerkung:

1. Wenn die Bereitschaftsdatenbank einen Protokollsatz verarbeitet, der angibt, dass in der primären Datenbank eine Indexwiederherstellung erfolgte, werden die Indizes auf dem Bereitschaftsserver nicht automatisch wiederhergestellt. Der Index auf dem Bereitschaftsserver wird entweder bei der ersten Verbindung zur Datenbank wiederhergestellt, oder beim ersten Versuch, auf den Index zuzugreifen, nachdem der Bereitschaftsserver aus dem Status 'Aktualisierende Recovery anstehend' genommen wurde. Wenn auf dem Primärserver Indizes wiederhergestellt werden, wird empfohlen, den Bereitschaftsserver erneut mit dem Primärserver zu synchronisieren. Sie können die erneute Erstellung von Indizes bei Rollforward-Operationen aktivieren, wenn Sie den Datenbankkonfigurationsparameter LOGINDEXBUILD definieren.
2. Wenn für die primäre Datenbank das Dienstprogramm LOAD mit der Option COPY YES ausgeführt wird, muss der Bereitschaftsserver Zugriff auf das Kopierimage haben.

3. Wenn für die primäre Datenbank das Dienstprogramm LOAD mit der Option COPY NO ausgeführt wird, sollte die Bereitschaftsdatenbank erneut synchronisiert werden, da andernfalls der Tabellenbereich in den Status 'Restore anstehend' versetzt wird.
4. Eine Bereitschaftsmaschine kann auf zwei Weisen initialisiert werden:
   a. Durch Wiederherstellen auf die Maschine von einem Backup-Image.
   b. Durch Erstellen einer geteilten Spiegeldatenbank des Produktionssystems und Absetzen des Befehls db2inidb mit der Option STANDBY.

   Sie können den Befehl ROLLFORWARD erst dann auf dem fehlertoleranten System absetzen, wenn die Bereitschaftsmaschine initialisiert wurde.
5. Operationen, die nicht protokolliert sind, werden nicht auf die Bereitschaftsdatenbank angewendet. Daher wird empfohlen, dass Sie die Bereitschaftsdatenbank nach solchen Operationen erneut synchronisieren. Dies ist durch die Unterstützung der ausgesetzten E/A und der Onlineteilung einer Spiegeldatenbank möglich.

Spiegeln von Protokollen

IBM Data Server unterstützt das Spiegeln von Protokollen auf Datenbankebene. Durch das Spiegeln von Protokolldateien wird eine Datenbank vor dem versehentlichen Löschen einer aktiven Protokolldatei sowie vor Datenverlust durch Hardwarefehler geschützt.

Wenn Sie befürchten, dass Ihre aktiven Protokolle aufgrund eines Plattenfehlers beschädigt werden könnten, sollten Sie in Betracht ziehen, mit dem Konfigurationsparameter MIRRORLOGPATH einen sekundären Pfad für die Datenbank zur Verwaltung von Kopien der aktiven Protokolldatei anzugeben, sodass die Datenträger gespiegelt werden, auf denen die Protokolle gespeichert sind.

Der Konfigurationsparameter MIRRORLOGPATH ermöglicht es der Datenbank, eine identische zweite Kopie der Protokolldateien in einen anderen Pfad zu schreiben. Es wird empfohlen, den sekundären Protokollpfad auf einem physisch getrennten Datenträger einzurichten (der sich vorzugsweise ebenfalls auf einem anderen Plattencontroller befinden sollte). Auf diese Weise kann der Plattencontroller nicht zum SPoF werden.

Wird MIRRORLOGPATH zum ersten Mal aktiviert, wird der Parameter erst beim nächsten Start der Datenbank verwendet. Dies verhält sich ähnlich wie beim Konfigurationsparameter NEWLOGPATH.

Wenn beim Schreiben in den Pfad für aktive Protokolldateien oder in den Pfad für gespiegelte Protokolldateien ein Fehler auftritt, markiert die Datenbank den entsprechenden Pfad als „unbrauchbar", schreibt eine Nachricht in das Protokoll mit den Benachrichtigungen für die Systemverwaltung und schreibt alle folgenden Protokollsätze ausschließlich in den verbleibenden „brauchbaren" Protokollpfad. DB2 versucht nicht, den „unbrauchbaren" Pfad erneut zu verwenden, bis die aktuelle Protokolldatei entweder voll ist oder abgeschnitten wird. Wenn DB2 die nächste Protokolldatei öffnen muss, wird sichergestellt, dass dieser Pfad gültig ist. Ist dies der Fall, wird der Pfad verwendet. Falls nicht, versucht DB2 nicht, den Pfad erneut zu verwenden, bis auf die nächste Protokolldatei zum ersten Mal zugegriffen wird. Es findet kein Versuch zur Synchronisierung der Protokollpfade statt, DB2 bewahrt jedoch die Informationen zu auftretenden Zugriffsfehlern auf, sodass die korrekten

Pfade verwendet werden, wenn die Protokolldateien archiviert werden. Wenn beim Schreiben in den verbleibenden „brauchbaren" Pfad ebenfalls ein Fehler auftritt, wird die Datenbank abnormal beendet.

Hohe Verfügbarkeit durch Unterstützung der ausgesetzten E/A und der Onlineteilung einer Spiegeldatenbank

Durch die Unterstützung der ausgesetzten E/A von IBM Data Server können Sie gespiegelte Kopien Ihrer Primärdatenbank teilen, ohne die Datenbank in den Offlinestatus versetzen zu müssen. Sie können diese Funktion verwenden, um rasch eine Bereitschaftsdatenbank zu erstellen, die die Arbeit übernimmt, wenn die Primärdatenbank fehlschlägt.

Die Plattenspiegelung ist ein Vorgang, bei dem Daten gleichzeitig auf zwei unterschiedliche Festplatten geschrieben werden. Die eine Kopie der Daten wird als der "Spiegel" der anderen Kopie bezeichnet. Das Teilen einer solchen Spiegelung bedeutet, dass die beiden Kopien voneinander getrennt werden.

Sie können mit der Plattenspiegelung eine Kopie Ihrer Primärdatenbank verwalten. Sie können die IBM Data Server-Funktionalität der ausgesetzten E/A dazu verwenden, die primären und sekundären gespiegelten Kopien der Datenbank zu teilen, ohne die Datenbank in den Offlinestatus versetzen zu müssen. Sobald die primären und sekundären Kopien der Datenbank geteilt sind, kann die sekundäre Datenbank die Operationen übernehmen, falls die Primärdatenbank ausfällt.

Wenn Sie eine große Datenbank lieber nicht mit dem Backup-Dienstprogramm von IBM Data Server sichern möchten, können Sie Kopien von einem Spiegelimage erstellen, indem Sie dazu die ausgesetzte E/A und die Funktion zur Erstellung einer Spiegeldatenbank verwenden. Außerdem erreichen Sie mit dieser Methode Folgendes:

- Sie vermeiden hohen Backup-Aufwand auf der Produktionsmaschine.
- Sie verfügen über eine schnelle Methode, Systeme zu klonen.
- Sie verfügen über eine schnelle Implementierung der Funktionsübernahme aus dem Bereitschaftsmodus. Ein einleitender Restore findet nicht statt, und sollte eine aktualisierende Recovery sich als zu langsam erweisen oder sollten Fehler auftreten, ist die erneute Initialisierung sehr schnell.

Mit dem Befehl db2inidb wird die geteilte Spiegeldatenbank initialisiert, sodass Sie sie wie folgt einsetzen können:

- Als Klondatenbank
- Als Bereitschaftsdatenbank
- Als Backup-Image

Dieser Befehl kann nur für eine geteilte Spiegeldatenbank abgesetzt werden, und die geteilte Spiegeldatenbank kann erst nach Ausführung des Befehls verwendet werden.

In einer Umgebung mit partitionierten Datenbanken müssen Sie E/A-Schreibvorgänge nicht auf allen Datenbankpartitionen gleichzeitig aussetzen. Sie können die E/A für eine Untergruppe aus einer oder mehreren Datenbankpartitionen aussetzen, um zum Ausführen von Offline-Backups geteilte Spiegeldatenbanken zu erstellen. Wenn in dieser Untergruppe die Katalogpartition enthalten ist, muss das Aussetzen für diese Datenbankpartition zuletzt erfolgen.

In einer Umgebung mit partitionierten Datenbanken müssen Sie den Befehl db2inidb auf jeder Datenbankpartition ausführen, bevor Sie das geteilte Image irgendeiner der Datenbankpartitionen verwenden können. Sie können das Tool in allen Datenbankpartitionen gleichzeitig ausführen, indem Sie den Befehl db2_all verwenden. Wenn Sie jedoch die Option RELOCATE USING verwenden, können Sie den Befehl db2inidb nicht mit dem Befehl db2_all auf allen Datenbankpartitionen gleichzeitig ausführen. Für jede Datenbankpartition muss eine eigene Konfigurationsdatei angegeben werden, die den Wert für NODENUM der zu ändernden Datenbankpartition enthält. Wenn beispielsweise der Name der Datenbank geändert wird, sind alle Datenbankpartitionen von dieser Änderung betroffen, und der Befehl db2relocatedb muss auf jeder Datenbankpartition mit einer eigenen Konfigurationsdatei ausgeführt werden. Wenn Container versetzt werden, die zu einer einzelnen Datenbankpartition gehören, muss der Befehl db2relocatedb nur einmal auf dieser Datenbankpartition ausgeführt werden.

Anmerkung: Stellen Sie sicher, dass die geteilte Spiegeldatenbank alle Container und Verzeichnisse enthält, aus denen die Datenbank besteht, einschließlich des Datenträgerverzeichnisses. Eine Zusammenstellung dieser Informationen finden Sie in der Verwaltungssicht DBPATHS, in der alle Dateien und Verzeichnisse der Datenbank angezeigt werden, die geteilt werden müssen.

Kapitel 3. Konfiguration für hohe Verfügbarkeit

Zur Konfiguration Ihrer DB2-Datenbanklösung für hohe Verfügbarkeit müssen Sie folgende Aufgaben erledigen: Datenbankverwaltungsaktivitäten terminieren, die Primär- und Bereitschaftsdatenbankserver so konfigurieren, dass sie aufeinander eingestellt sind und ihre jeweiligen Rollen im Falle einer Störung kennen, sowie das Konfigurieren von Cluster-Management-Software, um ggf. die Workload von einem fehlgeschlagenen Cluster-Knoten übertragen zu können.

Bevor Sie Ihre Datenbanklösung konfigurieren, müssen die folgenden Schritte ausgeführt werden:

- Montieren und installieren Sie die zugrunde liegenden Hardware- und Softwarekomponenten, aus denen Ihre Lösung besteht. Zu diesen zugrunde liegenden Komponenten können Folgende gehören: Stromversorgung, Netzkonnektivität, Netzkarten, Platten oder andere Speichereinheiten, Betriebssysteme und Software für das Cluster-Management.
- Testen Sie diese zugrunde liegenden Komponenten jeweils ohne eine Auslastung der Datenbank, um sicherzustellen, dass Sie ordnungsgemäß funktionieren, bevor Sie versuchen, sie in Operationen wie Datenbanklastausgleich, Übernahme oder Recovery zu verwenden.

Redundanz ist ein wichtiger Teil einer Hochverfügbarkeitslösung. Wenn Sie jedoch Ihre Verwaltungsaktivitäten auf einen ungünstigen Zeitpunkt legen, nicht mehr über ausreichend Speicherplatz für benötigte Recoveryprotokolle verfügen oder Ihre Cluster-Management-Software nicht korrekt konfiguriert ist, ist Ihre Lösung möglicherweise nicht verfügbar, wenn die Benutzer kritische Aufgaben mit der Datenbank zu erledigen haben.

Zur Konfiguration für hohe Verfügbarkeit gehört Folgendes:

- Konfiguration der Clientweiterleitung

  „Automatische Clientweiterleitung - Beschreibung und Einrichtung"
- Konfiguration des Fehlermonitors

  „DB2-Registrierungsdatei für Fehlermonitore" auf Seite 24
- Konfiguration von DB2 High Availability Disaster Recovery (HADR)

  „Initialisieren von High Availability Disaster Recovery (HADR)" auf Seite 27
- Terminierung von Verwaltungsaktivitäten

  „Terminieren von Verwaltungs- und Wartungsaktivitäten für hohe Verfügbarkeit" auf Seite 55
- Konfiguration der Protokollierung

  „Konfigurieren der Optionen zur Datenbankprotokollierung" auf Seite 58
- Konfiguration der Cluster-Management-Software

  „Konfigurieren einer Clusterumgebung für hohe Verfügbarkeit" auf Seite 74

Automatische Clientweiterleitung - Beschreibung und Einrichtung

Ziel der Funktion für automatische Clientweiterleitung ist es in erster Linie, nach einem Übertragungsausfall bei einer IBM Data Server-Clientanwendung eine Wiederherstellung zu ermöglichen, sodass die Anwendung den Betrieb nach einer möglichst kurzen Unterbrechung fortsetzen kann. Die Weiterleitung ist für die Un-

terstützung des Dauerbetriebs von zentraler Bedeutung. Eine Weiterleitung ist jedoch nur möglich, wenn für die Clientverbindung eine alternative Position angegeben und verfügbar ist.

Die Funktion für automatische Clientweiterleitung kann in den folgenden konfigurierbaren Umgebungen verwendet werden, wenn es sich bei dem Server um einen DB2-Server handelt, der unter Linux, UNIX oder Windows ausgeführt wird:

1. Enterprise Server Edition (ESE) mit dem Database Partitioning Feature (DPF)
2. WebSphere Replication Server
3. High Availability Cluster Multiprocessor (HACMP)
4. High Availability Disaster Recovery (HADR)

   Die automatische Clientweiterleitung funktioniert in Verbindung mit HADR, um einer Clientanwendung die Fortsetzung ihrer Arbeit bei minimaler Unterbrechung nach einer Funktionsübernahme (Failover) der Datenbank zu ermöglichen, auf die zugegriffen wird.

Die Funktion für automatische Clientweiterleitung kann ebenfalls in den folgenden Konfigurationen verwendet werden, wenn der Datenbankserver auf einem System i oder System z ausgeführt wird:

1. IBM Data Server Client stellt eine Verbindung zu einem z/OS- oder i5/OS-System über einen DB2 Connect-Server her, der über einen alternativen Server verfügt. Die automatische Clientweiterleitung wird zwischen IBM Data Server Client und zwei DB2 Connect-Servern eingesetzt.
2. DB2 Connect Personal- oder DB2 Connect-Serverprodukte, die auf eine DB2 für z/OS-Sysplexumgebung für die gemeinsame Nutzung von Daten zugreifen. Die automatische Clientweiterleitung wird zwischen DB2 Connect und dem z/OS-Sysplexsystem eingesetzt. Ab Version 9.5 Fixpack 3 unterstützt die Funktion der automatischen Clientweiterleitung die unterbrechungsfreie Funktionsübernahme zwischen einem für DB2 Connect lizenzierten Client und dem Sysplex. Weitere Informationen über die unterbrechungsfreie Funktionsübernahme enthält der Abschnitt zur automatischen Clientweiterleitung (clientseitig) in der DB2-Informationszentrale.

Da es nicht erforderlich ist, lokale Datenbanken zu synchronisieren, müssen Sie beim DB2 Connect-Server lediglich sicherstellen, dass für den ursprünglichen und den alternativen DB2 Connect-Server die Zielhost- bzw. System i-Datenbank so katalogisiert ist, dass sie über einen identischen Aliasnamen für die Datenbank zugänglich ist.

Um dem DB2-Datenbanksystem die Möglichkeit zu geben, einen Ausfall der Kommunikationsverbindung zu beheben, muss ein alternativer Serverstandort angegeben werden, bevor der Verlust der Kommunikation auftritt. Zum Definieren des alternativen Serverstandorts in einer bestimmten Datenbank wird der Befehl UPDATE ALTERNATE SERVER FOR DATABASE verwendet.

Wenn Sie den alternativen Serverstandort in einer bestimmten Datenbank auf der Serverinstanz angegeben haben, werden die Positionsinformationen dieses alternativen Servers an den IBM Data Server-Client im Rahmen des Verbindungsaufbauprozesses zurückgegeben. Bei einer automatischen Clientweiterleitung zwischen DB2 Connect Personal- oder -Serverprodukten und einem Host- oder System i-Datenbankserver muss der ferne Server mindestens eine alternative Adresse für sich selbst bereitstellen. Im Fall von DB2 für z/OS sind mehrere Adressen bekannt, wenn es sich bei der Datenbank um eine Sysplexumgebung für die gemeinsame Nutzung von Daten handelt, sodass die Katalogisierung eines alternativen Servers

unter DB2 Connect nicht erforderlich ist. Wenn die Kommunikation zwischen dem Client und dem Server aus irgendeinem Grund ausfällt, versucht der IBM Data Server-Client die Verbindung unter Verwendung der Informationen zu dem alternativen Server wiederherzustellen. Der IBM Data Server-Client wird versuchen, die Verbindung über einen Datenbankserver wiederherzustellen, bei dem es sich um den ursprünglichen Server, einen alternativen Server, der in der Datenbankverzeichnisdatei auf dem Server enthalten ist, oder einen alternativen Server handelt, der in der vom z/OS-Sysplexsystem zurückgegebenen Serverliste enthalten ist. Der Zeitraum zwischen diesen Versuchen zum Wiederherstellen einer Verbindung beginnt mit sehr kurzen Intervallen, die anschließend im Laufe der weiteren Versuche schrittweise verlängert werden.

Wenn eine Verbindung erfolgreich hergestellt werden kann, wird der SQLCODE-Wert -30108 zurückzugeben, um anzuzeigen, dass eine erneute Datenbankverbindung nach dem Kommunikationsausfall hergestellt wurde. Es werden der Hostname oder die IP-Adresse und der Servicename oder die Portnummer zurückgegeben. Der IBM Data Server-Client gibt den Fehler des ursprünglichen Kommunikationsausfalls nur an die Anwendung zurück, wenn die erneute Herstellung der Clientkommunikation mit dem ursprünglichen oder dem alternativen Server nicht möglich ist.

Darüber hinaus sind folgende Aspekte der Konnektivität alternativer Server in einer DB2 Connect-Serverumgebung zu beachten:

- Wenn Sie einen DB2 Connect-Server verwenden, um den Zugriff auf eine Host- oder System i-Datenbank für ferne und lokale Clients bereitzustellen, kann es hinsichtlich der Informationen zur Verbindung zu alternativen Servern in einem Eintrag des Systemdatenbankverzeichnisses zu Unklarheiten kommen. Wenn Sie diese Unklarheiten möglichst gering halten möchten, empfiehlt es sich, zwei Einträge im Systemdatenbankverzeichnis zu katalogisieren, die für denselben Host oder dieselbe System i-Datenbank stehen. Katalogisieren Sie einen Eintrag für ferne Clients und einen weiteren für lokale Clients.
- SYSPLEX-Informationen, die von einem DB2 für z/OS-Zielserver zurückgegeben werden, werden auf dem DB2 Connect-Server nur im Cache gespeichert. Es werden nur Informationen eines einzigen alternativen Servers auf Platte geschrieben. Sind mehrere alternative Server oder mehrere aktive Server vorhanden, werden die Informationen nur im Hauptspeicher verwaltet und stehen nach Prozessende nicht mehr zur Verfügung.

Im Allgemeinen wird bei angegebenem alternativen Server die automatische Clientweiterleitung aktiviert, wenn ein Kommunikationsfehler (SQLCODE-Wert -30081) oder ein SQLCODE-Wert -1224 erkannt wird. In einer HADR-Umgebung (High Availability Disaster Recovery) wird sie außerdem aktiviert, wenn SQLCODE-Wert -1776 vom HADR-Bereitschaftsserver zurückgegeben wird.

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 'DThostname'
- 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 **DThostname** 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 (DThostname) 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.)

Angeben eines alternativen Servers für die automatische Clientweiterleitung

Sobald ein DB2-Server oder ein DB2 Connect-Server ausfällt, empfängt jeder mit diesem Server verbundene Client einen Kommunikationsfehler, der die Verbindung beendet und einen Anwendungsfehler verursacht. In Fällen, in denen die Verfügbarkeit eine wichtige Rolle spielt, sollten Sie entweder eine redundante Konfiguration oder die Möglichkeit zur Übernahme der Serverfunktion durch einen Bereitschaftsknoten implementiert haben. In beiden Fällen versucht der DB2-Clientcode die Verbindung zu dem ursprünglichen Server, der jetzt auf einem Übernahmeknoten (die IP-Adresse wird ebenfalls übernommen) ausgeführt wird, oder zu einem neuen Server wiederherzustellen.

Gehen Sie wie folgt vor, um einen neuen oder alternativen Server zu definieren:

Verwenden Sie den Befehl **UPDATE ALTERNATE SERVER FOR DATABASE** oder **UPDATE ALTERNATE SERVER FOR LDAP DATABASE**.
Diese Befehle aktualisieren die Informationen zum alternativen Server für einen Aliasnamen einer Datenbank im Systemdatenbankverzeichnis.

Automatische Clientweiterleitung - Einschränkungen

Beachten Sie die Einschränkungen bei der Clientweiterleitung für DB2-Datenbanken, wenn Sie Ihre hoch verfügbare DB2-Datenbanklösung gestalten.

Im Folgenden werden die Einschränkungen aufgelistet, die für die Funktion zur automatischen Clientweiterleitung für DB2-Datenbanken gelten:

- Die automatische Clientweiterleitung wird nur unterstützt, wenn das Kommunikationsprotokoll TCP/IP zur Herstellung der Verbindung zum DB2-Datenbankserver bzw. zum DB2 Connect-Server verwendet wird. Dies bedeutet, dass die Funktion zur automatischen Clientweiterleitung nicht aktiviert wird, wenn die Verbindung mit einem anderen Protokoll als TCP/IP arbeitet. Auch wenn das DB2-Datenbanksystem mit der Loopback-Funktion konfiguriert ist, muss das Kommunikationsprotokoll TCP/IP verwendet werden, um die Funktion der automatischen Clientweiterleitung nutzen zu können.
- Bei einer automatischen Weiterleitung zwischen den DB2 Connect Pseronal- oder Server-Produkten und einem Host- oder System i-Datenbankserver sind in den folgenden Situationen gewisse Aspekte zu beachten:
  - Wenn Sie einen DB2 Connect-Server verwenden, um den Zugriff auf eine Host- oder System i-Datenbank für ferne und lokale Clients bereitzustellen, kann es hinsichtlich der Informationen zur Verbindung zu alternativen Servern in einem Eintrag des Systemdatenbankverzeichnisses zu Unklarheiten kommen. Wenn Sie diese Unklarheiten möglichst gering halten möchten, empfiehlt es sich, zwei Einträge im Systemdatenbankverzeichnis zu katalogisieren, die für denselben Host oder dieselbe System i-Datenbank stehen. Katalogisieren Sie einen Eintrag für ferne Clients und einen weiteren für lokale Clients.
  - SYSPLEX-Informationen, die von einem DB2 für z/OS-Zielserver zurückgegeben werden, werden auf dem DB2 Connect-Server nur im Cache gespeichert. Es werden nur Informationen eines einzigen alternativen Servers auf Platte geschrieben. Sind mehrere alternative Server oder mehrere aktive Server vorhanden, werden die Informationen nur im Hauptspeicher verwaltet und stehen nach Prozessende nicht mehr zur Verfügung.
- Wenn die Verbindung zum alternativen Serverstandort wiederhergestellt wird, wird jede neue Verbindung zum gleichen Aliasnamen der Datenbank zum alter-

nativen Serverstandort hergestellt. Wenn Sie wollen, dass neue Verbindungen zum ursprünglichen Standort hergestellt werden, nachdem das Problem am ursprünglichen Standort behoben ist, steht eine Reihe von Optionen dazu zur Auswahl:

  – Sie müssen den alternativen Server offline nehmen, sodass die Verbindungen zum ursprünglichen Server wiederhergestellt werden. (Dies setzt voraus, dass der ursprüngliche Server mit dem Befehl UPDATE ALTERNATE SERVER katalogisiert wurde, sodass er als alternativer Standort für den alternativen Server definiert ist.)
  – Sie könnten einen neuen Aliasnamen der Datenbank katalogisieren, der von den neuen Verbindungen zu verwenden ist.
  – Sie könnten den Datenbankeintrag entkatalogisieren und erneut katalogisieren.

- DB2 Database für Linux, UNIX und Windows unterstützt die Funktion der automatischen Clientweiterleitung sowohl für den Client als auch für den Server, wenn der Client und der Server diese Funktion unterstützen. Andere Familien von DB2-Datenbankprodukten unterstützen diese Funktion zurzeit nicht.
- Das Verhalten der Funktion zur automatischen Clientweiterleitung und das Verhalten der automatischen Clientweiterleitung in einer DB2 für z/OS Sysplex-Umgebung sind in gewisser Hinsicht anders. Dabei ist insbesondere zu beachten:

  – Die automatische Clientweiterleitungsfunktion setzt voraus, dass der Primärserver nur einen einzigen alternativen Server angibt. Dies geschieht mithilfe des Befehls UPDATE ALTERNATE SERVER FOR DATABASE oder UPDATE ALTERNATE SERVER FOR LDAP DATABASE, der auf dem primären Server ausgeführt wird. Dieser Befehl aktualisiert das lokale Datenbankverzeichnis mit den Informationen zum alternativen Server, sodass andere Anwendungen auf dem gleichen Client Zugriff auf diese Informationen haben. Im Gegensatz dazu verwaltet ein Sysplex mit gemeinsamer Datennutzung, das für DB2 für z/OS verwendet wird, im Speicher eine Liste mit Servern, zu denen der Client eine Verbindung herstellen kann. Wenn ein Kommunikationsfehler auftritt, bestimmt der Client mithilfe dieser Liste von Servern die Position des entsprechenden alternativen Servers.
  – Im Fall der automatischen Clientweiterleitungsfunktion informiert der Server den Client über die aktuellsten Einstellungen von Sonderregistern, wenn sich die Einstellung eines Sonderregisters ändert. Dies erlaubt dem Client, die Laufzeitumgebung nach besten Möglichkeiten wiederherzustellen, nachdem eine Weiterleitung stattgefunden hat. Im Gegensatz dazu gibt ein für DB2 für z/OS verwendetes Sysplex die Sonderregistereinstellungen zu Commitgrenzen an den Client zurück. Änderungen an innerhalb der weitergeleiteten UOW verwendeten Sonderregistern müssen deshalb nachvollzogen werden. Alle sonstigen Änderungen werden automatisch nachvollzogen.

  Mit dem Stand von DB2 Universal Database Version 8 FixPak 7 steht eine vollständige Unterstützung für die automatische Clientweiterleitung nur zwischen einem Linux-, UNIX- oder Windows-Client und einem Linux-, UNIX- oder Windows-Server zur Verfügung. Sie steht nicht zwischen einem Linux-, UNIX- oder Windows-Client und einem Sysplex-Server für DB2 für z/OS (unabhängig von der unterstützten Version) zur Verfügung. Lediglich die Weiterleitungsfunktion wird unterstützt.
- Der DB2-Datenbankserver, der auf dem alternativen Server installiert ist, muss dieselbe Version wie die DB2-Datenbankinstanz haben, die auf dem ursprünglichen Host-Server installiert ist (jedoch kann die FixPak-Version höher sein).

- Unabhängig davon, ob Sie die Berechtigung zum Aktualisieren des Datenbankverzeichnisses auf dem Clientsystem haben, werden die Informationen über den alternativen Server immer im Arbeitsspeicher verwaltet. Das heißt mit anderen Worten, dass andere Anwendungen, wenn Sie keine Berechtigung zum Aktualisieren des Datenbankverzeichnisses hatten (oder weil es sich um ein schreibgeschütztes Datenbankverzeichnis handelt), nicht in der Lage sind, den alternativen Server zu bestimmen und zu verwenden, weil der Arbeitsspeicher nicht von Anwendungen gemeinsam genutzt wird.
- Für alle alternativen Standorte wird die gleiche Authentifizierung ausgeführt. Das bedeutet, dass der Client keine neue Datenbankverbindung herstellen kann, wenn der alternative Standort mit einem anderen Authentifizierungstyp als der ursprüngliche Standort arbeitet.
- Wenn es zu einem Kommunikationsausfall kommt, gehen alle Sitzungsressourcen wie zum Beispiel globale temporäre Tabellen, Identitätswerte, Sequenzen, Cursor, Serveroptionen (SET SERVER OPTION) zur Verarbeitung in föderierten Umgebungen und Sonderregister verloren. Die Anwendung ist dafür zuständig, die Sitzungsressourcen neu einzurichten, um die Verarbeitung fortzusetzen. Sie brauchen nach Wiederherstellung der Verbindung keine der Anweisungen für Sonderregister auszuführen, weil die DB2-Datenbank die Anweisungen für Sonderregister automatisch noch einmal nachvollzieht, die vor dem Verbindungsfehler ausgeführt wurden. Allerdings werden einige Sonderregister nicht neu eingestellt. Dazu gehören die Folgenden:
  - SET ENCRYPTPW
  - SET EVENT MONITOR STATE
  - SET SESSION AUTHORIZATION
  - SET TRANSFORM GROUP

  Bei Problemen mit DB2 Connect empfiehlt es sich, die Liste der für DB2 Connect auf Datenservern spezifischen Sonderregister hinzuzuziehen.
- Wenn der Client mit einem CLI-JCC-Treiber des Typs 2 oder 4 arbeitet, werden nach der Wiederherstellung der Verbindung nach einem Übertragungsfehler die SQL- und XQuery-Anweisungen, die mit dem ursprünglichen Server bereits vorbereitet wurden, implizit mit dem neuen Server erneut vorbereitet. Eingebettete SQL-Routinen (z. B. SQC- oder SQX-Anwendungen) werden jedoch nicht mit dem neuen Server erneut vorbereitet.
- Führen Sie HADR-Befehle nicht mit Aliasnamen von Datenbanken mit aktivierter Clientweiterleitung aus. HADR-Befehle sind so implementiert, dass sie die Zieldatenbank mithilfe von Datenbankaliasnamen ermitteln. Wenn für die Zieldatenbank eine alternative Datenbank definiert ist, haben HADR-Befehle daher Schwierigkeiten, die Datenbank zu bestimmen, an der der Befehl tatsächlich ausgeführt wird. Während ein Client die Verbindung möglicherweise über einen Aliasnamen mit aktivierter Clientweiterleitung herstellen muss, müssen HADR-Befehle auf eine bestimmte Datenbank angewendet werden. Um diesem Umstand Rechnung zu tragen, können Sie Aliasnamen definieren, die für die Primär- und die Bereitschaftsdatenbank spezifisch sind, und lediglich HADR-Befehle mit diesen Aliasnamen ausführen.

Eine alternative Methode zur Implementierung der automatischen Clientweiterleitung ist die Verwendung des DNS-Eintrags zur Angabe einer alternativen IP-Adresse für einen DNS-Eintrag. Dieser Methode liegt die Idee zugrunde, eine zweite IP-Adresse (einen alternativen Serverstandort) in dem DNS-Eintrag anzugeben. Der Client besitzt dann zwar keine Informationen über einen alternativen Server, jedoch kann das DB2-Datenbanksystem beim Verbindungsaufbau zwischen zwei IP-Adressen für den DNS-Eintrag alternierend auswählen.

DB2-Registrierungsdatei für Fehlermonitore

Wenn der Dämonprozess des Fehlermonitors gestartet wird, wird eine Registrierungsdatei für Fehlermonitore für jede DB2-Datenbankmanagerinstanz auf allen physischen Maschinen gestartet. Die in dieser Datei enthaltenen Schlüsselwörter und Werte legen das Verhalten des Fehlermonitors fest.

Die Registrierungsdatei für Fehlermonitore befindet sich im Verzeichnis /sqllib/ und hat den Namen fm.<machine_name>.reg. Diese Datei kann mithilfe des Befehls db2fm geändert werden.

Wenn keine Registrierungsdatei für Fehlermonitore vorhanden ist, werden die Standardwerte verwendet.

Es folgt ein Beispiel für den Inhalt der Registrierungsdatei für Fehlermonitore:

```
FM_ON = no
FM_ACTIVE = yes
START_TIMEOUT = 600
STOP_TIMEOUT = 600
STATUS_TIMEOUT = 20
STATUS_INTERVAL = 20
RESTART_RETRIES = 3
ACTION_RETRIES = 3
NOTIFY_ADDRESS = <instance_name>@<machine_name>
```

Schlüsselwörter der Registrierungsdatei für Fehlermonitore

FM_ON

Gibt an, ob der Fehlermonitor gestartet werden soll oder nicht. Wenn der Wert auf NO gesetzt ist, wird der Fehlermonitordämon nicht gestartet bzw. wird inaktiviert, falls er bereits gestartet wurde. Der Standardwert ist NO.

FM_ACTIVE

Gibt an, ob der Fehlermonitor aktiv ist. Der Fehlermonitor wird nur dann Aktionen ausführen, wenn die Parameter FM_ON und FM_ACTIVE auf YES gesetzt sind. Wenn FM_ON auf YES und FM_ACTIVE auf NO gesetzt ist, wird der Fehlermonitordämon gestartet, er wird jedoch keine Aktionen ausführen. Das bedeutet, er wird nicht versuchen, DB2 erneut in den Onlinestatus zu versetzen, falls dieses beendet wird. Der Standardwert ist YES.

START_TIMEOUT

Gibt die Zeitspanne an, innerhalb der der Fehlermonitor den Service starten muss, den er überwacht. Der Standardwert ist 600 Sekunden.

STOP_TIMEOUT

Gibt die Zeitspanne an, innerhalb der der Fehlermonitor den Service beenden muss, den er überwacht. Der Standardwert ist 600 Sekunden.

STATUS_TIMEOUT

Gibt die Zeitspanne an, innerhalb der der Fehlermonitor den Status des Service abrufen muss, den er überwacht. Der Standardwert ist 20 Sekunden.

STATUS_INTERVAL

Gibt die Mindestzeit zwischen zwei aufeinander folgenden Aufrufen zum Abrufen des Status des überwachten Service an. Der Standardwert ist 20 Sekunden.

RESTART_RETRIES

Gibt an, wie oft der Fehlermonitor versuchen wird, nach einem fehlgeschlagenen Aufruf den Status des überwachten Service erneut abzurufen. Wenn der hier angegebene Wert erreicht wird, versucht der Fehlermonitor, den Service erneut in den Onlinestatus zu versetzen. Der Standardwert ist 3.

ACTION_RETRIES

Gibt an, wie oft der Fehlermonitor versuchen wird, den überwachten Service wieder in den Onlinestatus zu versetzen. Der Standardwert ist 3.

NOTIFY_ADDRESS

Gibt die E-Mail-Adresse an, an die der Fehlermonitor Benachrichtigungen sendet. Der Standardwert ist <instanzname>@<name_der_maschine>)

DB2-Fehlermonitor mit dem Befehl db2fm konfigurieren

Sie können die DB2-Registrierdatei für Fehlermonitore mit dem Befehl db2fm ändern.

Es folgen einige Beispiele für die Verwendung des Befehls db2fm zur Aktualisierung der Registrierdatei für Fehlermonitore:

Beispiel 1: START_TIMEOUT aktualisieren

Wenn Sie den Wert für START_TIMEOUT für die Instanz DB2INST1 auf 100 Sekunden aktualisieren möchten, geben Sie den folgenden Befehl in ein DB2-Datenbankbefehlsfenster ein:

```
db2fm -i db2inst1 -T 100
```

Beispiel 2: STOP_TIMEOUT aktualisieren

Wenn Sie den Wert für STOP_TIMEOUT für die Instanz DB2INST1 auf 200 Sekunden aktualisieren wollen, geben Sie den folgenden Befehl ein:

```
db2fm -i db2inst1 -T /200
```

Beispiel 3: START_TIMEOUT und STOP_TIMEOUT aktualisieren

Wenn Sie für die Instanz DB2INST1 den Wert START_TIMEOUT auf 100 Sekunden und den Wert für STOP_TIMEOUT auf 200 aktualisieren wollen, geben Sie den folgenden Befehl ein:

```
db2fm -i db2inst1 -T 100/200
```

Beispiel 4: Fehlermonitor aktivieren

Wenn Sie die Fehlermonitorfunktion für die Instanz DB2INST1 aktivieren wollen, geben Sie den folgenden Befehl ein:

```
db2fm -i db2inst1 -f yes
```

Beispiel 5: Fehlermonitor inaktivieren

Wenn Sie die Fehlermonitorfunktion für die Instanz DB2INST1 inaktivieren wollen, geben Sie den folgenden Befehl ein:

```
db2fm -i db2inst1 -f no
```

Zum Nachweis, dass die Fehlermonitorfunktion für DB2INST1 nicht mehr aktiv ist, geben Sie auf UNIX-Systemen den folgenden Befehl ein:

```
ps -ef|grep -i fm
```

Unter Linux geben Sie den folgenden Befehl ein:

```
ps auxw|grep -i fm
```

Ein Eintrag, der db2fmd und DB2INST1 enthält, gibt an, dass der Fehlermonitor für diese Instanz immer noch aktiv ist. Zur Inaktivierung des Fehlermonitors geben Sie den folgenden Befehl als Instanzeigner ein:

```
db2fm -i db2inst1 -D
```

DB2-Fehlermonitor mit db2fmc und Systembefehlen konfigurieren

Sie können den DB2-Fehlermonitor mithilfe des Befehls db2fmcu der DB2 Fault Monitor Controller Utility (FMCU) oder mit Systembefehlen konfigurieren.

Es folgen einige Beispiele für die Verwendung des Befehls db2fmcu und von Systembefehlen, um den Fehlermonitor zu konfigurieren:

Beispiel 1: Verhindern, dass der FMC gestartet wird

Mithilfe der Fault Monitor Controller Utility (FMCU) von DB2 können Sie verhindern, dass der FMC gestartet wird. Die FMCU muss mit Rootberechtigung ausgeführt werden, da sie auf die Datei 'inittab' des Systems zugreift. Führen Sie den folgenden Befehl aus, um zu verhindern, dass der FMC ausgeführt wird:

```
db2fmcu -d
```

Anmerkung: Wenn Sie ein DB2 Data Server-Fixpack anwenden, wird dieses zurückgesetzt, sodass die Datei 'inittab' erneut so konfiguriert wird, dass sie den FMC enthält. Um zu verhindern, dass der FMC gestartet wird, nachdem Sie ein Fixpack angewendet haben, müssen Sie den oben angegebenen Befehl erneut ausführen.

Beispiel 2: Den FMC für den Start wiederaufnehmen

Wenn Sie den Befehl `db2fmcu -d` rückgängig machen und den FMC wieder in die Datei 'inittab' aufnehmen möchten, geben Sie den folgenden Befehl ein:

```
db2fmcu -u -p  <vollständiger_pfad>
```

Dabei gilt: <vollständiger_pfad> gibt den vollständigen Pfad zum db2fmcd-Objekt an, z. B. '/opt/IBM/db2/bin/db2fmcd'.

Beispiel 3: Die DB2-Datenbankmanagerinstanz automatisch starten

Darüber hinaus können Sie den FMC so einrichten, dass er die Instanz startet, wenn das System anfangs gebootet wird. Zur Aktivierung dieser Funktion für die Instanz DB2INST1 geben Sie den folgenden Befehl ein:

```
db2iauto -on db2inst1
```

Beispiel 4: Automatisches Starten der Instanz inaktivieren

Wenn Sie die automatische Startfunktion inaktivieren möchten, geben Sie den folgenden Befehl ein:

```
db2iauto -off db2inst1
```

Beispiel 5: Verhindern, dass Fehlermonitorprozesse gestartet werden

Sie haben auch die Möglichkeit, das Starten von Fehlermonitorprozessen für bestimmte Instanzen auf dem System zu verhindern, indem Sie ein Feld im globalen Registrierdatensatz für die Instanz ändern. Zum Ändern

des globalen Registrierfeldes zur Inaktivierung von Fehlermonitoren für die Instanz DB2INST1 geben Sie zum Beispiel den folgenden Befehl als Rootbenutzer ein:

```
db2greg -updinstrec instancename=db2inst1!startatboot=0
```

Wenn Sie diesen Befehl rückgängig machen und Fehlermonitore für die Instanz DB2INST1 reaktivieren möchten, geben Sie den folgenden Befehl als Rootbenutzer ein:

```
db2greg -updinstrec instancename=db2inst1!startatboot=1
```

Initialisieren von High Availability Disaster Recovery (HADR)

Verwenden Sie die folgende Prozedur, um die primäre und die Bereitschaftsdatenbank für DB2 High Availability Disaster Recovery (HADR) einzurichten und zu initialisieren.

HADR kann über den Befehlszeilenprozessor (CLP), den Assistenten zum Konfigurieren von HADR in der Steuerzentrale oder durch Aufrufen der Anwendungsprogrammierschnittstelle (API) *db2HADRStart* initialisiert werden.

Gehen Sie wie folgt vor, um auf Ihrem System HADR über den Befehlszeilenprozessor zum ersten Mal zu initialisieren:

1. Ermitteln Sie für jede der HADR-Datenbanken den Hostnamen, die IP-Adresse und den Servicenamen oder die Portnummer.

   Wenn ein Host über mehrere Netzschnittstellen verfügt, stellen Sie sicher, dass der HADR-Hostname oder die IP-Adresse mit der gewünschten übereinstimmt. Sie müssen für jede geschützte Datenbank separate HADR-Ports in der Datei '/etc/services' zuordnen. Diese dürfen nicht mit den der Instanz zugeordneten Ports identisch sein. Der Hostname kann nur einer einzigen IP-Adresse zugeordnet werden.

   Anmerkung: Die Instanznamen für die Primärdatenbank und Bereitschaftsdatenbanken müssen nicht identisch sein.
2. Erstellen Sie die Bereitschaftsdatenbank, indem Sie ein Backup-Image wiederherstellen oder indem Sie basierend auf der vorhandenen Datenbank, die als Primärdatenbank verwendet werden soll, eine geteilte Spiegeldatenbank initialisieren.

   Im folgenden Beispiel werden die Befehle BACKUP DATABASE und RESTORE DATABASE verwendet, um die Datenbank SOCKS als Bereitschaftsdatenbank zu initialisieren. Im Fall dieses Beispiels kann auf ein angehängtes NFS-Dateisystem von beiden Standorten aus zugegriffen werden.

   Setzen Sie in der Primärdatenbank den folgenden Befehl ab:

   ```
   backup db socks to /nfs1/backups/db2/socks
   ```

   Setzen Sie in der Bereitschaftsdatenbank den folgenden Befehl ab:

   ```
   restore db socks from /nfs1/backups/db2/socks replace history file
   ```

   Das folgende Beispiel zeigt, wie mit dem Dienstprogramm db2inidb unter Verwendung einer geteilten Spiegeldatenbank der Primärdatenbank die Bereitschaftsdatenbank initialisiert wird. Diese Prozedur ist eine Alternative zur oben beschriebenen Prozedur zum Sichern und Wiederherstellen.

   Setzen Sie in der Bereitschaftsdatenbank den folgenden Befehl ab:

   ```
   db2inidb socks as standby
   ```

Anmerkung:

a. Die Datenbanknamen für die Primärdatenbank und die Bereitschaftsdatenbank müssen identisch sein.

b. Es wird empfohlen, dass Sie nach der Restoreoperation oder der Initialisierung der geteilten Spiegeldatenbank nicht den Befehl ROLLFORWARD DATABASE für die Bereitschaftsdatenbank absetzen. Die Ergebnisse einer Operation zur aktualisierenden Recovery können geringfügig von dem Ergebnis abweichen, das durch die Anwendung der Protokolle auf die Bereitschaftsdatenbank unter Verwendung von HADR erzielt wird. Wenn die Datenbanken nicht identisch sind, schlägt das Absetzen des Befehls START HADR mit der Option AS STANDBY fehl.

c. Wenn Sie den Befehl RESTORE DATABASE verwenden, wird empfohlen, die Option REPLACE HISTORY FILE zu verwenden.

d. Wenn Sie die Bereitschaftsdatenbank mithilfe des Befehls RESTORE DATABASE erstellen, müssen Sie sicherstellen, dass die Bereitschaftsdatenbank im Modus für aktualisierende Recovery verbleibt. Das bedeutet, dass Sie den Befehl ROLLFORWARD DATABASE weder mit der Option COMPLETE noch mit der Option STOP ausführen können. Wenn in der Datenbank nach Beendigung der aktualisierenden Recovery versucht wird, den Befehl START HADR mit der Option AS STANDBY auszuführen, wird ein Fehler zurückgegeben.

e. Beim Konfigurieren der Bereitschaftsdatenbank sollten Sie die folgenden Optionen des Befehls RESTORE DATABASE vermeiden: TABLESPACE, INTO, REDIRECT und WITHOUT ROLLING FORWARD.

f. Wenn Sie die Bereitschaftsdatenbank mit dem Dienstprogramm db2inidb konfigurieren, verwenden Sie nicht die Optionen SNAPSHOT oder MIRROR. Sie können die Option RELOCATE USING angeben, um wahlweise die folgenden Konfigurationsattribute zu ändern: Instanzname, Protokollpfad und Datenbankpfad. Sie dürfen jedoch nicht den Namen der Datenbank oder die Pfade der Tabellenbereichscontainer ändern.

3. Setzen Sie die HADR-Konfigurationsparameter in der Primärdatenbank und der Bereitschaftsdatenbank.

   Anmerkung: Es ist sehr wichtig, dass Sie die folgenden Konfigurationsparameter im Anschluss an die Erstellung der Bereitschaftsdatenbank definieren:

   - HADR_LOCAL_HOST
   - HADR_LOCAL_SVC
   - HADR_REMOTE_HOST
   - HADR_REMOTE_SVC
   - HADR_REMOTE_INST

   Wenn diese Parameter vor dem Erstellen der Bereitschaftsdatenbank definiert werden, geben die Einstellungen in der Bereitschaftsdatenbank die Einstellungen der Primärdatenbank wieder.

4. Stellen Sie eine Verbindung zur Bereitschaftsinstanz her, und starten Sie HADR für die Bereitschaftsdatenbank, wie im folgenden Beispiel gezeigt:

   ```
   START HADR ON DB SOCKS AS STANDBY
   ```

   Anmerkung: In der Regel wird die Bereitschaftsdatenbank zuerst gestartet. Wenn Sie die Primärdatenbank zuerst starten, schlägt diese Startprozedur fehl, wenn die Bereitschaftsdatenbank nicht innerhalb des vom Datenbankkonfigurationsparameter HADR_TIMEOUT angegebenen Zeitlimits gestartet wird.

5. Stellen Sie eine Verbindung zur Primärinstanz her, und starten Sie HADR für die Primärdatenbank, wie im folgenden Beispiel gezeigt:

   ```
   START HADR ON DB SOCKS AS PRIMARY
   ```

6. HADR wird nun für die Primärdatenbank und die Bereitschaftsdatenbank gestartet:

   Gehen Sie wie folgt vor, um den Assistenten zum Konfigurieren von HADR-Datenbanken zu öffnen:

   a. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie die Datenbank finden, für die HADR konfiguriert werden soll.
   b. Klicken Sie die Datenbank mit der rechten Maustaste an, und klicken Sie im Kontextmenü die Option **HADR (High Availability Disaster Recovery) → Einrichten** an. Der Assistent zum Konfigurieren von HADR-Datenbanken wird geöffnet.

   Weitere Informationen enthält die Kontexthilfefunktion der Steuerzentrale.

   Anmerkung: Sie können HADR im Assistenten zur Konfiguration von HADR-Datenbanken starten, oder den Assistenten nur zur Initialisierung von HADR verwenden und HADR zu einem anderen Zeitpunkt starten. Öffnen Sie das Fenster **HADR starten** wie folgt:

   a. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie die Datenbank finden, für die HADR verwaltet werden soll. Klicken Sie die Datenbank mit der rechten Maustaste an, und klicken Sie im Kontextmenü die Option **HADR (High Availability Disaster Recovery) → Verwalten** an. Das Fenster **HADR verwalten** wird geöffnet.
   b. Klicken Sie **HADR starten** an. Das Fenster **HADR starten** wird geöffnet.

Automatische Clientweiterleitung und HADR (High Availability Disaster Recovery) konfigurieren

Sie können die Funktion für automatische Clientweiterleitung mit der Funktion HADR (High Availability Disaster Recovery) verwenden, um die Anforderungen der Clientanwendungen eines fehlgeschlagenen Datenbankservers an einen Bereitschaftsdatenbankserver weiterzuleiten.

Einschränkungen

- Eine Weiterleitung ist nur möglich, wenn auf dem Server eine alternative Datenbankposition angegeben wurde.
- Die automatische Clientweiterleitung wird nur mit TCP/IP-Protokoll unterstützt.

Konfigurationsdetails

- Verwenden Sie den Befehl UPDATE ALTERNATE SERVER FOR DATABASE, um die automatische Clientweiterleitung zu aktivieren.
- Die Clientweiterleitung wird standardmäßig aktiviert, wenn Sie HADR mithilfe des Assistenten zum Konfigurieren von HADR-Datenbanken in der Steuerzentrale einrichten.
- Die automatische Clientweiterleitung verwendet die Datenbankkonfigurationsparameter HADR_REMOTE_HOST und HADR_REMOTE_SVC nicht.
- Die alternative Hostposition ist in der Systemdatenbankverzeichnisdatei auf dem Server gespeichert.
- Wenn die automatische Clientweiterleitung nicht aktiviert ist, erhalten Clientanwendungen die Fehlernachricht SQL30081, und es werden keine weiteren Versuche unternommen, eine Verbindung zum Server herzustellen.

Befehl UPDATE ALTERNATE SERVER FOR DATABASE zum Einrichten der automatischen Clientweiterleitung mit HADR verwenden

Ihr System ist wie folgt eingerichtet:

- Sie haben einen Client, auf dem die Datenbank MUSIC mit der Information katalogisiert ist, dass sie sich auf dem Host HORNET befindet.
- Die Datenbank MUSIC ist die Primärdatenbank, und ihre zugehörige Bereitschaftsdatenbank, die ebenfalls den Namen MUSIC hat, befindet sich auf Host MONTERO mit der Portnummer 456, die durch den Konfigurationsparameter SVCENAME zugewiesen ist.

Um die automatische Clientweiterleitung zu aktivieren, aktualisieren Sie den alternativen Server für die Datenbank MUSIC auf Host HORNET:

```
db2 update alternate server for database music using hostname montero port 456
```

Nachdem dieser Befehl abgesetzt wurde, muss der Client eine erfolgreiche Verbindung zum Host HORNET herstellen, um die Informationen zum alternativen Server abzurufen. Wenn dabei ein Kommunikationsfehler zwischen dem Client und der Datenbank MUSIC auf dem Host HORNET auftritt, versucht der Client zunächst, erneut eine Verbindung zur Datenbank MUSIC auf dem Host HORNET herzustellen. Wenn dies fehlschlägt, versucht der Client anschließend, eine Verbindung zur Bereitschaftsdatenbank MUSIC auf dem Host MONTERO herzustellen.

Indexprotokollierung und High Availability Disaster Recovery (HADR)

Es empfiehlt sich, die Datenbankkonfigurationsparameter LOGINDEXBUILD und INDEXREC für DB2-HADR-Datenbanken einzurichten.

Verwenden des Datenbankkonfigurationsparameters LOGINDEXBUILD

Empfehlung: Setzen Sie den Datenbankkonfigurationsparameter LOGINDEXBUILD für HADR-Datenbanken auf ON, um sicherzustellen, dass für die Erstellung, Neuerstellung und Reorganisation von Indizes die vollständigen Informationen protokolliert werden. Auch wenn dies bedeutet, dass die Indexerstellung auf dem primären System mehr Zeit beansprucht und mehr Protokollspeicherbereich erforderlich ist, werden die Indizes während der Wiederholung des HADR-Protokolls erneut erstellt und stehen für den Fall einer Funktionsübernahme zur Verfügung. Wenn die Indexerstellung auf dem primären System nicht protokolliert wird und eine Funktionsübernahme erforderlich wird, müssen alle ungültigen Indizes, die nach Abschluss der Funktionsübernahme übrig bleiben, erneut erstellt werden, bevor auf sie zugegriffen werden kann. Während der erneuten Erstellung der Indizes stehen diese anderen Anwendungen nicht zur Verfügung.

Anmerkung: Wenn das Tabellenattribut LOG INDEX BUILD auf seinen Standardwert NULL gesetzt ist, verwendet DB2 den für den Datenbankkonfigurationsparameter LOGINDEXBUILD angegebenen Wert. Wenn das Tabellenattribut LOG INDEX BUILD auf ON oder OFF gesetzt ist, wird der Wert für den Datenbankkonfigurationsparameter LOGINDEXBUILD ignoriert.

Sie können das Tabellenattribut LOG INDEX BUILD aus einem der folgenden Gründe für eine Tabelle oder mehrere Tabellen auf OFF setzen:

- Sie verfügen nicht über ausreichend Speicherbereich für die aktiven Protokolldateien, um die Protokollierung der Indexerstellung zu unterstützen.

- Die Indexdaten sind sehr umfangreich, und auf die Tabelle wird nicht häufig zugegriffen; daher ist es vertretbar, die Indizes im Anschluss an die Übernahmeoperation erneut zu erstellen. Setzen Sie in diesem Fall den Konfigurationsparameter INDEXREC auf RESTART. Da auf die Tabelle nicht häufig zugegriffen wird, bewirkt diese Einstellung, dass das System die Indizes am Ende der Übernahmeoperation erneut erstellt und nicht auf den ersten Zugriff auf die Tabelle nach der Übernahmeoperation wartet.

Wenn das Tabellenattribut LOG INDEX BUILD für mindestens eine Tabelle auf OFF gesetzt ist, kann eine beliebige Indexerstellungsoperation für diese Tabellen dazu führen, dass die Indizes bei jedem Auftreten einer Funktionsübernahmeoperation erneut erstellt werden. Ist das Tabellenattribut LOG INDEX BUILD auf seinen Standardwert NULL und der Datenbankkonfigurationsparameter LOGINDEXBUILD auf OFF gesetzt, kann ebenfalls eine beliebige Indexerstellungsoperation für diese Tabellen dazu führen, dass die Indizes bei jedem Auftreten einer Übernahmeoperation erneut erstellt werden. Sie können die erneute Erstellung der Indizes verhindern, indem Sie eine der folgenden Aktionen ausführen:

- Nachdem alle ungültigen Indizes in der neuen Primärdatenbank erneut erstellt wurden, erstellen Sie ein Backup der Datenbank, und wenden Sie dieses Backup auf die Bereitschaftsdatenbank an. Dadurch muss die Bereitschaftsdatenbank nicht die zur erneuten Erstellung ungültiger Indizes verwendeten Protokolle auf die Primärdatenbank anwenden, wodurch diese Indizes in der Bereitschaftsdatenbank eine Markierung erhalten würden, dass eine erneute Erstellung erforderlich ist.
- Setzen Sie das Tabellenattribut LOG INDEX BUILD auf ON, oder setzen Sie das Tabellenattribut LOG INDEX BUILD auf NULL und den Konfigurationsparameter LOGINDEXBUILD für die Bereitschaftsdatenbank auf ON, um sicherzustellen, dass die Indexerstellung protokolliert wird.

Verwenden des Datenbankkonfigurationsparameters INDEXREC

Empfehlung: Setzen Sie den Datenbankkonfigurationsparameter INDEXREC für die Primärdatenbank und die Bereitschaftsdatenbank auf RESTART (Standardeinstellung). Diese Einstellung bewirkt, dass ungültige Indizes nach Abschluss einer Übernahmeoperation erneut erstellt werden. Wenn noch keine Indexerstellungen protokolliert wurden, ermöglicht diese Einstellung DB2 die Suche nach ungültigen Indizes und das erneute Erstellen dieser Indizes. Dieser Prozess findet im Hintergrund statt, und die Datenbank steht nach der erfolgreichen Beendigung der Übernahmeoperation wieder zur Verfügung.

Wenn eine Transaktion auf eine Tabelle zugreift, die ungültige Indizes enthält, bevor die Indizes durch den Hintergrundprozess erneut erstellt wurden, werden die ungültigen Indizes durch diese erste Transaktion erneut erstellt.

Datenbankkonfiguration für HADR (High Availability Disaster Recovery)

Mit der Hilfe von Datenbankkonfigurationsparametern können Sie mit DB2 High Availability Disaster Recovery (HADR) eine optimale Leistung erreichen.

Um eine optimale Leistung mit der DB2 High Availability Disaster Recovery (HADR) zu erreichen, sollten Sie sicherstellen, dass Ihre Datenbankkonfiguration die folgenden Voraussetzungen erfüllt.

Empfehlung: Die Datenbankkonfigurationsparameter und Datenbankmanagerkonfigurationsparameter sollten auf allen Systemen, auf denen sich die Primärdaten-

bank und die Bereitschaftsdatenbank befinden, möglichst identisch definiert sein. Wenn die Konfigurationsparameter für die Bereitschaftsdatenbank nicht richtig gesetzt wurden, können folgende Probleme auftreten:

- Die Bereitschaftsdatenbank gibt Fehlernachrichten zurück, während sie die von der Primärdatenbank übertragenen Protokolldateien anwendet.
- Nach einer Übernahmeoperation kann die neue Primärdatenbank die Auslastung nicht bewältigen, wodurch es zu Leistungsproblemen oder zur Ausgabe von Fehlernachrichten kommen kann, welche die Anwendungen während ihrer Verbindungen zur ursprünglichen Primärdatenbank nicht empfingen.

Änderungen an den Konfigurationsparametern in der Primärdatenbank werden nicht automatisch an die Bereitschaftsdatenbank weitergegeben und müssen in der Bereitschaftsdatenbank manuell vorgenommen werden. Für dynamische Konfigurationsparameter werden diese Änderungen wirksam, ohne dass das Datenbankverwaltungssystem (DBMS - Database Management System) oder die Datenbank heruntergefahren und erneut gestartet werden müssen. Bei nicht dynamischen Konfigurationsparametern werden die Änderungen nach dem Neustart der Bereitschaftsdatenbank wirksam.

Konfigurationsparameter für die Größe der Protokolldateien in der Bereitschaftsdatenbank

Eine Ausnahme von der oben beschriebenen Funktionsweise der Konfigurationsparameter ist der Datenbankkonfigurationsparameter LOGFILSIZ. Auch wenn dieser Parameter nicht in die Bereitschaftsdatenbank repliziert wird, ignoriert die Bereitschaftsdatenbank die lokale Konfiguration für LOGFILSIZ und erstellt lokale Protokolldateien, die in der Größe den Protokolldateien der Primärdatenbank entsprechen, um identische Protokolldateien für beide Datenbanken zu garantieren.

Nach einer Übernahme verwendet das ursprüngliche Bereitschaftssystem (das neue primäre System) weiterhin den Wert, der auf dem ursprünglich primären System festgelegt wurde, bis die Datenbank erneut gestartet wird. An diesem Punkt kehrt das neue primäre System zu seinem lokal konfigurierten Wert zurück. Darüber hinaus schneidet das neue primäre System die aktuelle Protokolldatei ab und ändert die Größen aller im Voraus erstellten Protokolldateien.

Wenn die Datenbanken die Rollen im Rahmen einer nicht erzwungenen Übernahme tauschen und keine Datenbank inaktiviert wird, bleibt die Protokolldateigröße immer diejenige, die durch das zu Anfang erste primäre System festgelegt wurde. Wenn das ursprüngliche Bereitschaftssystem (das neue primäre System) jedoch inaktiviert und erneut gestartet wird, verwendet es die lokal konfigurierte Protokolldateigröße. Diese Protokolldateigröße wird anschließend auch dann weiter verwendet, wenn das ursprünglich primäre System erneut übernimmt. Erst nach einer Inaktivierung und einem Neustart des ursprünglich primären Systemen nimmt die Protokolldateigröße wieder den Wert des ursprünglich primären Systems an.

Größe des Protokollempfangspuffers in der Bereitschaftsdatenbank

Standardmäßig entspricht die Größe des Protokollempfangspuffers in der Bereitschaftsdatenbank dem doppelten Wert des in der Primärdatenbank angegebenen Konfigurationsparameters LOGBUFSZ. Diese Größe kann in manchen Fällen nicht ausreichen. Wenn beispielsweise der HADR-Modus ASYNC verwendet wird und sich die Primärdatenbank und die Bereitschaftsdatenbank im Peerstatus befinden, kann sich die Kapazität des Protokollempfangspuffers in der Bereitschaftsdatenbank bei Auftreten einer hohen Transaktionslast erschöpfen und die Protokollüber-

tragungsoperation von der Primärdatenbank zum Erliegen kommen. Um diese temporären hohen Systemauslastungen zu bewältigen, können Sie die Größe des Protokollempfangspuffers in der Bereitschaftsdatenbank erhöhen, indem Sie die Registrierdatenbankvariable DB2_HADR_BUF_SIZE modifizieren.

Ladeoperationen und HADR

Wenn eine Ladeoperation in der Primärdatenbank mit der Option COPY YES ausgeführt wird, wird der Befehl in der Primärdatenbank ausgeführt, und die Daten werden in die Bereitschaftsdatenbank repliziert, solange über den mit dem Befehl LOAD angegebenen Pfad oder der angegebenen Einheit auf die Kopie zugegriffen werden kann. Kann die Bereitschaftsdatenbank auf diese Daten nicht zugreifen, wird der Tabellenbereich, in dem die Tabelle gespeichert ist, in der Bereitschaftsdatenbank als ungültig markiert. Die Bereitschaftsdatenbank überspringt in der Folge zukünftige Protokollsätze, die zu diesem Tabellenbereich gehören. Um sicherzustellen, dass die Ladeoperation auf die Kopie in der Bereitschaftsdatenbank zugreifen kann, wird empfohlen, eine gemeinsam genutzte Speicherposition für die Ausgabedatei der Option COPY YES zu verwenden. Alternativ können Sie auch die Bereitschaftsdatenbank für die Dauer der Ladeoperation inaktivieren, das Laden in der Primärdatenbank durchführen, eine Kopie der Ausgabedatei in den Bereitschaftspfad stellen und anschließend die Bereitschaftsdatenbank wieder aktivieren.

Wenn eine Ladeoperation in der Primärdatenbank mit der Option NONRECOVERABLE ausgeführt wird, wird der Befehl in der Primärdatenbank ausgeführt, und die Tabelle wird in der Bereitschaftsdatenbank als fehlerhaft markiert. Die Bereitschaftsdatenbank überspringt zukünftige Protokollsätze, die zu diesem Tabellenbereich gehören. Sie können den Befehl LOAD wahlweise mit den Optionen COPY YES und REPLACE angeben, um die Tabelle zurückzuholen, oder Sie können die Tabelle löschen, um den Speicherbereich freizugeben.

Da HADR das Ausführen einer Ladeoperation mit der Option COPY NO nicht unterstützt, wird der Befehl automatisch in eine Ladeoperation mit der Option NONRECOVERABLE konvertiert. Definieren Sie in der Primärdatenbank die Registrierdatenbankvariable DB2_LOAD_COPY_NO_OVERRIDE, wodurch eine Ladeoperation mit der Option COPY NO in eine Ladeoperation mit der Option COPY YES konvertiert wird. Diese Registrierdatenbankvariable wird von der Bereitschaftsdatenbank ignoriert. Stellen Sie sicher, dass die Bereitschaftsdatenbank auf die in der Primärdatenbank angegebene Einheit oder das angegebene Verzeichnis über denselben Pfad bzw. mit derselben Einheit oder Ladebibliothek zugreifen kann.

Wenn Sie Tivoli Storage Manager (TSM) zum Ausführen einer Ladeoperation mit der Option COPY YES verwenden, müssen Sie möglicherweise in der Primärdatenbank und der Bereitschaftsdatenbank den Konfigurationsparameter VENDOROPT setzen. Je nach der Konfiguration von TSM sind die Werte in der Primärdatenbank und der Bereitschaftsdatenbank möglicherweise unterschiedlich. Wenn Sie TSM zum Ausführen einer Ladeoperation mit der Option COPY YES verwenden, müssen Sie außerdem den Befehl db2adutl mit der Option GRANT absetzen, um der Bereitschaftsdatenbank Lesezugriff für die geladenen Dateien zu erteilen.

Wenn Tabellendaten von einer Ladeoperation mit der Option COPY YES repliziert werden, werden die Indizes wie folgt repliziert:

- Wenn der Indexierungsmodus auf REBUILD gesetzt ist und das Tabellenattribut auf LOG INDEX BUILD oder wenn das Tabellenattribut auf DEFAULT gesetzt ist und der Datenbankkonfigurationsparameter LOGINDEXBUILD auf ON,

schließt die Primärdatenbank das erneut erstellte Indexobjekt in die Kopierdatei ein, um der Bereitschaftsdatenbank das Replizieren des Indexobjekts zu ermöglichen. Wenn das Indexobjekt in der Bereitschaftsdatenbank vor der Ladeoperation als ungültig markiert wurde, wird es nach der Ladeoperation durch die erneute Indexerstellung wieder verwendbar.

- Wenn der Indexierungsmodus auf INCREMENTAL gesetzt ist und das Tabellenattribut auf LOG INDEX BUILD, oder wenn das Tabellenattribut auf NULL gesetzt ist und der Datenbankkonfigurationsparameter LOGINDEXBUILD in der Primärdatenbank auf ON, wird das Indexobjekt in der Bereitschaftsdatenbank nur dann aktualisiert, wenn es nicht vor der Ladeoperation als ungültig markiert wurde. Andernfalls wird der Index in der Bereitschaftsdatenbank als ungültig markiert.

Registrierdatenbankvariable DB2_HADR_PEER_WAIT_LIMIT

Durch das Setzen der Registrierdatenbankvariablen *DB2_HADR_PEER_WAIT_LIMIT* wird für die HADR-Primärdatenbank der Peerstatus aufgehoben, wenn die Protokollierung in der Primärdatenbank für die angegebene Anzahl von Sekunden blockiert wurde, weil das Protokoll in die Bereitschaftsdatenbank repliziert wird. Wenn dieser Grenzwert erreicht wird, unterbricht die Primärdatenbank die Verbindung zur Bereitschaftsdatenbank. Wenn das Peerfenster inaktiviert ist, wird die Primärdatenbank in den Unterbrechungsstatus versetzt, und die Protokollierung wird wieder aufgenommen. Wenn das Peerfenster aktiviert ist, wird die Primärdatenbank in den Status 'Getrennter Peer' versetzt, in dem die Protokollierung weiterhin blockiert ist. Der Status 'Getrennter Peer' wird für die Primärdatenbank aufgehoben, sobald die Verbindung erneut hergestellt wird oder das Peerfenster abgelaufen ist. Die Protokollierung wird wieder aufgenommen, sobald der Status 'Getrennter Peer' für die Primärdatenbank aufgehoben ist.

Die Berücksichtigung der Statusänderung für das Peerfenster beim Aufheben des Peerstatus stellt die Peerfenstersemantik für eine sichere Übernahme in allen Fällen sicher. Bei einem Ausfall der Primärdatenbank während der Statusänderung gilt weiterhin der normale Peerfensterschutz (sichere Übernahme von der Bereitschaftsdatenbank, während sie sich im Status 'Getrennter Peer' befindet).

Auf der Seite der Bereitschaftsdatenbank werden nach dem Trennen der Verbindung die bereits empfangenen Protokolle wiederholt. Nachdem die empfangenen Protokolle wiederholt wurden, stellt die Bereitschaftsdatenbank die Verbindung zur Primärdatenbank wieder her. Bei der Wiederherstellung der Verbindung erfolgt der normale Statusübergang (zuerst in den Status 'Fernes Catch-up', dann in den Peerstatus).

Beziehung zu HADR_TIMEOUT:

Der Datenbankkonfigurationsparameter **HADR_TIMEOUT** bewirkt keine Aufhebung des Peerstatus für die Primärdatenbank, wenn die Primärdatenbank während der Blockierung weiterhin Überwachungssignalnachrichten von der Bereitschaftsdatenbank empfängt. **HADR_TIMEOUT** stellt ein Zeitlimit für die HADR-Netzebene dar. Eine HADR-Datenbank trennt die Verbindung zu ihrer Partnerdatenbank, wenn sie von der Partnerdatenbank innerhalb des durch **HADR_TIMEOUT** angegebenen Zeitraums keine Nachrichten erhalten hat. Das Zeitlimit für Operationen höherer Ebenen, wie beispielsweise die Protokollübertragung und die Empfangsbestätigung, wird durch diesen Parameter nicht gesteuert. Wenn die Protokollwiederholung in der Bereitschaftsdatenbank bei einer großen Operation, wie beispielsweise dem Laden oder der Reorganisation, blockiert ist, sendet die HADR-Komponente dennoch dem normalen Zeitplan entsprechend Überwachungssignalnachrich-

ten an die Primärdatenbank. In einem solchen Szenario bleibt die Primärdatenbank so lange blockiert wie die Wiederholung in der Bereitschaftsdatenbank, es sei denn, *DB2_HADR_PEER_WAIT_LIMIT* ist definiert.

DB2_HADR_PEER_WAIT_LIMIT hebt die Blockierung der Protokollierung in der Primärdatenbank auf, unabhängig vom Verbindungsstatus. Auch wenn *DB2_HADR_PEER_WAIT_LIMIT* nicht definiert ist, wird der Peerstatus für die Primärdatenbank stets aufgehoben, wenn ein Netzfehler festgestellt oder die Verbindung getrennt wird (z. B. als Ergebnis von **HADR_TIMEOUT**).

HADR-Konfigurationsparameter

Zur Unterstützung von HADR stehen mehrere neue Datenbankkonfigurationsparameter zur Verfügung. Die Einstellung dieser Parameter hat keine Auswirkungen auf die Rolle einer Datenbank. Zum Ändern der Rolle einer Datenbank müssen Sie die Befehle START HADR oder STOP HADR absetzen.

Die HADR-Konfigurationsparameter sind nicht dynamisch. Die an den HADR-Konfigurationsparametern vorgenommenen Änderungen treten erst dann in Kraft, wenn die Datenbank heruntergefahren und erneut gestartet wird. In einer Umgebung mit partitionierten Datenbanken sind die HADR-Konfigurationsparameter sichtbar und können geändert werden, sie werden jedoch ignoriert.

Der lokale Hostname der Primärdatenbank muss mit dem fernen Hostnamen der Bereitschaftsdatenbank übereinstimmen, und der lokale Hostname der Bereitschaftsdatenbank muss mit dem fernen Hostnamen der Primärdatenbank übereinstimmen. Verwenden Sie die Konfigurationsparameter HADR_LOCAL_HOST und HADR_REMOTE_HOST, um die lokalen und fernen Hosts für die beiden Datenbanken festzulegen. Die Konsistenz der Konfiguration der lokalen und fernen Hostnamen wird beim Herstellen einer Verbindung überprüft, um sicherzustellen, dass es sich bei dem angegebenen fernen Host um die erwartete Datenbank handelt.

Eine HADR-Datenbank kann zur Verwendung von IPv4 oder IPv6 zur Lokalisierung der Partnerdatenbank konfiguriert werden. Wenn der Host-Server IPv6 nicht unterstützt, verwendet die Datenbank IPv4. Wenn der Server IPv6 unterstützt, hängt die Verwendung von IPv4 oder IPv6 durch die Datenbank von dem Format der Adresse ab, die in den Konfigurationsparametern HADR_LOCAL_HOST und HADR_REMOTE_HOST angegeben ist. Die Datenbank versucht, die beiden Parameter in dasselbe IP-Format aufzulösen. Die folgende Tabelle zeigt, wie der IP-Modus durch IPv6-fähige Server bestimmt wird:

In HADR_LOCAL_HOST verwendeter IP-Modus	**In HADR_REMOTE_HOST verwendeter IP-Modus**	**Zur HADR-Kommunikation verwendeter IP-Modus**
IPv4-Adresse	IPv4-Adresse	IPv4
IPv4-Adresse	IPv6-Adresse	Fehler
IPv4-Adresse	Hostname, nur in v4-Adresse auflösbar	IPv4
IPv4-Adresse	Hostname, nur in v6-Adresse auflösbar	Fehler
IPv4-Adresse	Hostname, in v4- und v6-Adresse auflösbar	IPv4
IPv6-Adresse	IPv4-Adresse	Fehler
IPv6-Adresse	IPv6-Adresse	IPv6

In HADR_LOCAL_HOST verwendeter IP-Modus	In HADR_REMOTE_HOST verwendeter IP-Modus	Zur HADR-Kommunikation verwendeter IP-Modus
IPv6-Adresse	Hostname, nur in v4-Adresse auflösbar	Fehler
IPv6-Adresse	Hostname, nur in v6-Adresse auflösbar	IPv6
IPv6-Adresse	Hostname, in v4- und v6-Adresse auflösbar	IPv6
Hostname, nur in v4-Adresse auflösbar	IPv4-Adresse	IPv4
Hostname, nur in v4-Adresse auflösbar	IPv6-Adresse	Fehler
Hostname, nur in v4-Adresse auflösbar	Hostname, nur in v4-Adresse auflösbar	IPv4
Hostname, nur in v4-Adresse auflösbar	Hostname, nur in v6-Adresse auflösbar	Fehler
Hostname, nur in v4-Adresse auflösbar	Hostname, in v4- und v6-Adresse auflösbar	IPv4
Hostname, nur in v6-Adresse auflösbar	IPv4-Adresse	Fehler
Hostname, nur in v6-Adresse auflösbar	IPv6-Adresse	IPv6
Hostname, nur in v6-Adresse auflösbar	Hostname, nur in v4-Adresse auflösbar	Fehler
Hostname, nur in v6-Adresse auflösbar	Hostname, nur in v6-Adresse auflösbar	IPv6
Hostname, nur in v6-Adresse auflösbar	Hostname, in v4- und v6-Adresse auflösbar	IPv6
Hostname, in v4- und v6-Adresse auflösbar	IPv4-Adresse	IPv4
Hostname, in v4- und v6-Adresse auflösbar	IPv6-Adresse	IPv6
Hostname, in v4- und v6-Adresse auflösbar	Hostname, nur in v4-Adresse auflösbar	IPv4
Hostname, in v4- und v6-Adresse auflösbar	Hostname, nur in v6-Adresse auflösbar	IPv6
Hostname, in v4- und v6-Adresse auflösbar	Hostname, in v4- und v6-Adresse auflösbar	IPv6

Die Primärdatenbank und die Bereitschaftsdatenbank können HADR-Verbindungen nur herstellen, wenn sie dasselbe Format verwenden. Wenn auf dem einen Server IPv6 eingerichtet ist (jedoch auch IPv4 unterstützt wird) und der andere Server nur IPv4 unterstützt, muss mindestens einer der Parameter HADR_LOCAL_HOST oder HADR_REMOTE_HOST eine IPv4-Adresse angeben. Dadurch wird die Datenbank angewiesen, IPv4 zu verwenden, auch wenn der Server IPv6 unterstützt.

Wenn Sie bei der Vorbereitung eines Befehls zur Aktualisierung der Datenbank (update database configuration) Werte für die Parameter für den lokalen HADR-Service (HADR_LOCAL_SVC) und den fernen HADR-Service (HADR_REMOTE_SVC) angeben, müssen die angegebenen Werte Ports bezeichnen, die nicht für ei-

nen anderen Service, das heißt, auch nicht für andere DB2-Komponenten oder andere HADR-Datenbanken, genutzt werden. Insbesondere dürfen Sie keinen der Parameterwerte auf den TCP/IP-Port, über den der Server die Kommunikation von fernen Clients erwartet (Datenbankkonfigurationsparameter SVCENAME), oder den nächsten Port (SVCENAME + 1) setzen.

Wenn sich die Primärdatenbank und die Bereitschaftsdatenbank auf verschiedenen Systemen befinden, können sie die gleiche Portnummer bzw. den gleichen Servicenamen verwenden. Ansonsten sollten unterschiedliche Werte verwendet werden. Die Parameter HADR_LOCAL_SVC und HADR_REMOTE_SVC können entweder auf eine Portnummer oder einen Servicenamen gesetzt werden.

Der Synchronisationsmodus (HADR_SYNCMODE) und das Zeitlimitintervall (HADR_TIMEOUT) müssen für die Primärdatenbank und die Bereitschaftsdatenbank identisch sein. Die Konsistenz dieser Konfigurationsparameter wird überprüft, wenn ein HADR-Paar eine Verbindung herstellt.

Für die Kommunikation zwischen der Primärdatenbank und der Bereitschaftsdatenbank werden TCP-Verbindungen verwendet. Eine Primärdatenbank, die nicht mit einer Bereitschaftsdatenbank verbunden ist, da sie entweder gerade gestartet wird oder die Verbindung unterbrochen wurde, ist an ihrem lokalen Port empfangsbereit für neue Verbindungen. Eine Bereitschaftsdatenbank, die nicht mit einer Primärdatenbank verbunden ist, sendet weiter Verbindungsanforderungen an ihren fernen Host.

Zwar werden der lokale Host und die lokalen Serviceparameter (HADR_LOCAL_HOST, HADR_LOCAL_SVC) nur auf der Primärdatenbank verwendet, Sie sollten Sie aber trotzdem auf der Bereitschaftsdatenbank definieren, um sicherzustellen, dass sie bereit sind, wenn die Bereitschaftsdatenbank die Operationen als Primärdatenbank übernehmen muss.

Wenn die Primärdatenbank gestartet wird, wartet sie mindestens 30 Sekunden oder die Anzahl von Sekunden, die durch den Wert des Datenbankkonfigurationsparameters HADR_TIMEOUT definiert ist, (je nachdem, welcher Zeitraum länger ist) darauf, dass die Bereitschaftsdatenbank eine Verbindung herstellt. Wenn die Bereitschaftsdatenbank keine Verbindung innerhalb des angegeben Zeitraums herstellt, schlägt der Start fehl. (Die einzige Ausnahme gilt für den Fall, dass der Befehl START HADR mit der Option BY FORCE ausgeführt wird.)

Nach der Herstellung der Verbindung eines HADR-Paares, tauschen die Datenbanken Überwachungssignalnachrichten aus. Das Intervall der Überwachungssignale beträgt ein Viertel des Werts des Datenbankkonfigurationsparameters HADR_TIMEOUT oder 30 Sekunden, je nachdem, welches Intervall kürzer ist. Das Monitorelement HADR_HEARTBEAT zeigt die Anzahl von Überwachungssignalen, deren Empfang eine Datenbank von der anderen Datenbank erwartete, jedoch nicht empfangen hat. Wenn eine Datenbank überhaupt keine Nachricht von der anderen Datenbank innerhalb der durch HADR_TIMEOUT definierten Sekunden empfängt, leitet sie eine Trennung der Verbindung ein. Dies bedeutet, dass es mindestens die im Parameter HADR_TIMEOUT angegebene Anzahl von Sekunden dauert, bis eine Primärdatenbank den Ausfall entweder der Bereitschaftsdatenbank oder des verbindenden Netzwerks erkennt. Wenn Sie den Konfigurationsparameter HADR_TIMEOUT zu niedrig einstellen, sind Fehlalarme und häufige Trennungen von Verbindungen die Folge.

Wenn der Datenbankkonfigurationsparameter HADR_PEER_WINDOW auf Null gesetzt ist und sich die Primär- und die Bereitschaftsdatenbank im Peerstatus be-

finden, blockieren Probleme mit der Bereitschaftsdatenbank oder dem Netzwerk die Transaktionsverarbeitung auf der Primärdatenbank höchstens für die Anzahl an Sekunden, die im Konfigurationsparameter HADR_TIMEOUT angegeben ist. Wenn Sie HADR_PEER_WINDOW auf einen Wert ungleich Null setzen, werden Transaktionen von der Primärdatenbank erst dann festgeschrieben, wenn die Verbindung mit der Bereitschaftsdatenbank wiederhergestellt ist bzw. der Zeitwert für HADR_PEER_WINDOW abläuft.

Anmerkung: Für maximale Verfügbarkeit wird der Standardwert für den Datenbankkonfigurationsparameter HADR_PEER_WINDOW auf Null gesetzt. Wenn HADR_PEER_WINDOW auf Null gesetzt ist, verlässt die Primärdatenbank den Peerstatus, sobald die Verbindung zwischen der Primär- und der Bereitschaftsdatenbank geschlossen wurde (entweder weil die Bereitschaftsdatenbank die Verbindung geschlossen hat, ein Netzwerkfehler festgestellt oder das Zeitlimit erreicht wurde), um eine Blockierung von Transaktionen zu vermeiden. Für höhere Datenkonsistenz (was jedoch eine verringerte Verfügbarkeit zur Folge hat) können Sie den Datenbankkonfigurationsparameter HADR_PEER_WINDOW auf einen Wert ungleich Null setzen; in diesem Fall verbleibt die Primärdatenbank für den über HADR_PEER_WINDOW angegebenen Zeitraum im Status 'Unterbrochener Peer'.

Im Folgenden sehen Sie eine Beispielkonfiguration für die Primärdatenbank und die Bereitschaftsdatenbank.

Für die Primärdatenbank:

```
HADR_LOCAL_HOST           host1.ibm.com
HADR_LOCAL_SVC            hadr_service
HADR_REMOTE_HOST          host2.ibm.com
HADR_REMOTE_SVC           hadr_service
HADR_REMOTE_INST          dbinst2
HADR_TIMEOUT              120
HADR_SYNCMODE             NEARSYNC   HADR_PEER_WINDOW          120
```

Für die Bereitschaftsdatenbank:

```
HADR_LOCAL_HOST           host2.ibm.com
HADR_LOCAL_SVC            hadr_service
HADR_REMOTE_HOST          host1.ibm.com
HADR_REMOTE_SVC           hadr_service
HADR_REMOTE_INST          dbinst1
HADR_TIMEOUT              120
HADR_SYNCMODE             NEARSYNC   HADR_PEER_WINDOW          120
```

Konfigurieren der Datenbankkonfigurationsparameter 'hadr_timeout' und 'hadr_peer_window'

Konfigurationsparameter hadr_timeout

Wenn eine HADR-Datenbank für länger als ein Zehntel der Zeit, die durch den Datenbankkonfigurationsparameter **hadr_timeout** vorgegeben ist, keine Kommunikation von ihrer Partnerdatenbank erhält, geht die Datenbank davon aus, dass die Verbindung zur Partnerdatenbank unterbrochen wurde. Wenn sich die Datenbank zum Zeitpunkt der Verbindungsunterbrechung im Peerzustand befindet, geht sie in den Status 'Getrennter Peer' (der Wert des Datenbankkonfigurationsparameters **hadr_peer_window** ist größer als Null) oder in den Status 'Fernes Catch-up anstehend' (der Wert von **hadr_peer_window** ist nicht größer als Null) über. Die Statusänderung betrifft sowohl Primär- als auch Bereitschaftsdatenbanken.

Konfigurationsparameter hadr_peer_window

Der Konfigurationsparameter **hadr_peer_window** ersetzt den Konfigurationsparameter **hadr_timeout** nicht. Der Konfigurationsparameter **hadr_ti-**

meout bestimmt, wie lange eine HADR-Datenbank wartet, bevor sie davon ausgeht, dass ihre Verbindung zur Partnerdatenbank unterbrochen wurde. Der Konfigurationsparameter **hadr_peer_window** bestimmt, ob die Datenbank in den Status 'Getrennter Peer' übergeht, wenn die Verbindung unterbrochen wird, und wie lange die Datenbank in diesem Status verbleiben soll. HADR unterbricht die Verbindung, wenn ein Netzfehler während eines Sende-, Empfangs- oder Polling-Vorgangs am TCP-Socket festgestellt wird. HADR fragt das Socket alle 100 Millisekunden ab. Dadurch kann rasch auf Netzfehler reagiert werden, die vom Betriebssystem erkannt werden. Nur im ungünstigsten Fall wartet HADR bis zur Zeitlimitüberschreitung, um eine fehlerhafte Verbindung zu trennen. In diesem Fall kann eine Datenbankanwendung, die bei Auftreten des Fehlers ausgeführt wird, für einen bestimmten Zeitraum blockiert werden, der der Summe der Datenbankkonfigurationsparameter **hadr_timeout** und **hadr_peer_window** entspricht.

Konfigurieren der Datenbankkonfigurationsparameter hadr_timeout und hadr_peer_window

Es ist wünschenswert, die Wartezeit für eine Datenbankanwendung zu minimieren. Wenn für die Konfigurationsparameter **hadr_timeout** und **hadr_peer_window** niedrige Werte angegeben werden, verkürzt dies die Wartezeit für die Datenbankanwendung, wenn die Verbindung einer HADR-Bereitschaftsdatenbank zur Primärdatenbank unterbrochen wird. Bei der Festlegung der Werte für die Konfigurationsparameter **hadr_timeout** und **hadr_peer_window** sollte aber Folgendes beachtet werden:

- Der Datenbankkonfigurationsparameter **hadr_timeout** sollte auf einen Wert gesetzt werden, der lang genug ist, um Fehlalarme bei der HADR-Verbindung zu vermeiden, die durch kurze vorübergehende Netzunterbrechungen verursacht werden. Beispielsweise beträgt der Standardwert von **hadr_timeout** 120 Sekunden - ein Wert, der für viele Netze geeignet ist.
- Der Datenbankkonfigurationsparameter **hadr_peer_window** sollte auf einen Wert eingestellt werden, der lang genug ist, um es dem System zu ermöglichen, beim Auftreten von Fehler automatisch zu intervenieren. Wenn das HA-System - z. B. ein Cluster-Manager - den Primärdatenbankfehler erkennt, bevor der Status 'Getrennter Peer' endet, erfolgt eine Funktionsübernahme durch die Bereitschaftsdatenbank. Bei der Funktionsübernahme gehen keine Daten verloren, da alle Daten von der alten Primärdatenbank in die neue Primärdatenbank repliziert werden. Wenn der Wert für **hadr_peer_window** zu kurz ist, hat das HA-System möglicherweise nicht genug Zeit, um den Fehler zu erkennen und entsprechend zu intervenieren.

Konfiguration der Protokollarchivierung für DB2 High Availability Disaster Recovery (HADR)

Konfigurieren Sie für die Verwendung der Protokollarchivierung mit DB2 High Availability Disaster Recovery (HADR) die Primärdatenbank und die Bereitschaftsdatenbank so, dass die Funktion für das automatische Abrufen von Protokollen aus allen Protokollarchivpositionen eingerichtet ist.

Wenn entweder die Bereitschaftsdatenbank oder die Primärdatenbank keinen Zugriff auf alle Protokollarchivpositionen hat, müssen Sie Protokolldateien manuell aus dem Protokollarchiv an die folgenden Positionen kopieren:

- In den Protokollpfad oder an die Archivposition der Bereitschaftsdatenbank zum lokalen Catch-up

- In den Protokollpfad oder an die Archivposition der Primärdatenbank zum fernen Catch-up

Nur die aktuelle Primärdatenbank kann eine Protokollarchivierung ausführen. Wenn die Primär- und die Bereitschaftsdatenbank mit separaten Archivierungspositionen konfiguriert sind, werden nur an der Archivierungsposition der Primärdatenbank Protokolle archiviert. Im Falle einer Übernahme wird die Bereitschaftsdatenbank zur neuen Primärdatenbank. Alle nachfolgend archivierten Protokolle werden an der Archivierungsposition der ursprünglichen Bereitschaftsdatenbank gespeichert. In einer solchen Konfiguration werden Protokolle entweder an der einen oder an der anderen Position gespeichert, jedoch nicht an beiden. Eine Ausnahme ist möglich, wenn die neue Primärdatenbank nach einer Übernahme einige wenige Protokolle archiviert, die von der ursprünglichen Primärdatenbank bereits archiviert wurden.

Wenn nach einer Übernahme die neue Primärdatenbank (d. h. die ursprüngliche Bereitschaftsdatenbank) einen Datenträgerausfall hat, sodass ein Restore und eine aktualisierende Recovery ausgeführt werden müssen, muss sie eventuell auf Protokolle zugreifen, die nur an der Archivposition der ursprünglichen Primärdatenbank vorhanden sind.

Die Bereitschaftsdatenbank löscht keine Protokolldatei aus ihrem lokalen Protokollpfad, bis sie von der Primärdatenbank benachrichtigt wurde, das die Primärdatenbank diese archiviert hat. Diese Funktionsweise stellt einen zusätzlichen Schutz gegen den Verlust von Protokolldateien bereit. Wenn die Primärdatenbank ausfällt und ihre Protokollplatte beschädigt wird, bevor eine bestimmte Protokolldatei in der Primärdatenbank archiviert wurde, löscht die Bereitschaftsdatenbank diese Protokolldatei von ihrer eigenen Platte nicht, weil sie keine Benachrichtigung empfangen hat, dass die Primärdatenbank die Protokolldatei erfolgreich archiviert hat. Wenn die Bereitschaftsdatenbank später als neue Primärdatenbank übernimmt, archiviert sie diese Protokolldatei, bevor sie die Datei wieder freigibt. Wenn sowohl der Konfigurationsparameter *logarchmeth1* als auch der Konfigurationsparameter *logarchmeth2* verwendet werden, gibt die Bereitschaftsdatenbank eine Protokolldatei erst frei, wenn die Primärdatenbank sie mit beiden Methoden archiviert hat.

Anmerkung: Um den Catch-up-Prozess zu beschleunigen, können Sie eine gemeinsam genutzte Protokollarchiveinheit für die Primärdatenbank und die Bereitschaftsdatenbank verwenden. Dies ermöglicht es der Bereitschaftsdatenbank, ältere Protokolldateien direkt aus dem Archiv im lokalen Catch-up-Status abzurufen, anstatt diese Dateien indirekt über die Primärdatenbank im fernen Catch-up-Status abzurufen. Es wird jedoch empfohlen, keine serielle Archivierungseinheit wie z. B. ein Bandlaufwerk für HADR-Datenbanken zu verwenden. Bei seriellen Einheiten kann es aufgrund von gemischten Lese- und Schreiboperationen sowohl auf der Primär- als auch auf der Bereitschaftsdatenbank zu Leistungseinbußen kommen. Die Primärdatenbank schreibt auf die Einheit, wenn sie Protokolldateien archiviert, und die Bereitschaftsdatenbank liest von der Einheit, wenn sie Protokolle wiedergibt. Diese Leistungsbeeinträchtigung kann auch dann auftreten, wenn die Einheit nicht als gemeinsam genutzt konfiguriert ist.

HADR-Leistung

Das Konfigurieren verschiedener Aspekte Ihres Datenbanksystems einschließlich der Netzbandbreite, der Leistung der Systemeinheit und der Puffergröße kann die Leistung Ihrer DB2-HADR-Datenbanken verbessern.

Beachten Sie zur Erzielung einer optimalen Leistung mit HADR (High Availability Disaster Recovery) die folgenden Empfehlungen zur Verwaltung Ihres Systems:

- Die Netzbandbreite muss größer als die Generierungsgeschwindigkeit des Datenbankprotokolls sein.
- Verzögerungen bei der Netzübertragung wirken sich nur auf die Modi SYNC und NEARSYNC aus.
- Die Verlangsamung der Systemleistung infolge des SYNC-Modus kann erheblich größer sein als die der anderen Synchronisationsmodi. Im SYNC-Modus sendet die Primärdatenbank Protokollseiten an die Bereitschaftsdatenbank erst, nachdem die Protokollseiten erfolgreich auf die Protokollplatte der Primärdatenbank geschrieben wurden. Zum Schutz der Integrität des Systems wartet die Primärdatenbank auf eine Bestätigung von der Bereitschaftsdatenbank, bevor sie eine Anwendung benachrichtigt, dass eine Transaktion vorbereitet (prepared) oder festgeschrieben (committed) wurde. Die Bereitschaftsdatenbank sendet die Bestätigung erst, nachdem sie die empfangenen Protokollseiten auf ihre Platte, d. h. die Platte der Bereitschaftsdatenbank, geschrieben hat. Der resultierende Zeitaufwand setzt sich demnach aus dem Schreiben der Protokolle in der Bereitschaftsdatenbank sowie dem Hin- und Hersenden von Nachrichten zusammen.
- Im NEARSYNC-Modus schreibt die Primärdatenbank Protokollseiten und sendet sie gleichzeitig in einer parallelen Operation. Anschließend wartet die Primärdatenbank auf die Bestätigung von der Bereitschaftsdatenbank. Die Bereitschaftsdatenbank sendet die Bestätigung, sobald sie die Protokollseiten in ihrem Speicher empfangen hat. In einem schnellen Netzwerk ist der Zeitaufwand für die Primärdatenbank minimal. Die Bestätigung kann bereits eingetroffen sein, wenn die Primärdatenbank das lokale Schreiben ihrer Protokolle beendet.
- Beim ASYNC-Modus erfolgen das Schreiben und Senden von Protokollseiten ebenfalls parallel. In diesem Modus wartet die Primärdatenbank jedoch nicht auf eine Bestätigung von der Bereitschaftsdatenbank. Daher spielen Verzögerungen bei der Netzübertragung keine Rolle. Die Leistungseinbußen sind beim ASYNC-Modus sogar noch geringer als beim NEARSYNC-Modus.
- Für jeden Schreibvorgang von Protokollseiten in der Primärdatenbank werden diese Protokollseiten auch an die Bereitschaftsdatenbank gesendet. Jede Schreiboperation wird als *Flushoperation* bezeichnet. Der Umfang der Flushoperation ist auf die Protokollpuffergröße in der Primärdatenbank begrenzt (die durch den Datenbankkonfigurationsparameter **logbufsz** gesteuert wird). Die exakte Größe der einzelnen Flushoperationen ist nicht deterministisch. Ein größerer Protokollpuffer führt nicht zwangsläufig zu einem größeren Flushvolumen.
- Die Leistungsfähigkeit der Bereitschaftsdatenbank sollte ausreichen, um die protokollierten Operationen der Datenbank ebenso schnell nachvollziehen zu können, wie sie in der Primärdatenbank generiert werden. Daher wird für die Primär- und die Bereitschaftsdatenbank eine identische Hardwareausstattung empfohlen.
- In den meisten Systemen wird die Protokollierungsfunktion nicht bis zur Kapazitätsgrenze ausgenutzt. Selbst im SYNC-Modus lässt sich vielleicht keine Verlangsamung in der Primärdatenbank beobachten. Wenn die Protokollierkapazität bei aktiviertem HADR zum Beispiel 40 Mb pro Sekunde beträgt, das System vor der Aktivierung von HADR jedoch nur mit 30 Mb arbeitete, bemerken Sie möglicherweise keinen Unterschied in der Gesamtsystemleistung.
- Um den Catch-up-Prozess zu beschleunigen, können Sie eine gemeinsam genutzte Protokollarchiveinheit verwenden. Wenn die gemeinsam genutzte Einheit jedoch eine serielle Einheit ist (z. B. ein Bandlaufwerk), kann durch die gemischten Lese- und Schreibvorgänge die Leistung verringert werden.

Netzüberlastung

Wenn die Bereitschaftsdatenbank bei der Anwendung der Protokollseiten zu langsam arbeitet, füllt sich der Protokollempfangspuffer vollständig an, sodass der Puffer keine weiteren Protokollseiten mehr empfangen kann. Wenn die Datenbanken im SYNC- oder NEARSYNC-Modus arbeiten und die Primärdatenbank ein weiteres Mal ihren Protokollpuffer durch eine Flushoperation leert und die Protokollseiten sendet, werden die Daten wahrscheinlich in der Netzwerkpipeline gepuffert, die aus der primären Maschine, dem Netzwerk und der Bereitschaftsdatenbank besteht. Da die Bereitschaftsdatenbank keinen freien Puffer zum Empfang der Daten hat, kann sie keine Bestätigung senden, sodass die Primärdatenbank blockiert wird, solange sie auf die Bestätigung der Bereitschaftsdatenbank wartet.

Im ASYNC-Modus fährt die Primärdatenbank mit dem Senden von Protokollseiten fort, bis sich die Pipeline gänzlich füllt und sie keine weiteren Protokollseiten senden kann. Eine solche Bedingung wird als *Überlastung* bezeichnet. Eine Überlastung wird durch das Monitorelement **hadr_connect_status** gemeldet. In den Modi SYNC und NEARSYNC kann die Pipeline in der Regel eine einzelne Flushoperation auffangen, sodass keine Überlastung auftritt. Die Primärdatenbank bleibt jedoch blockiert, solange sie auf eine Bestätigung von der Bereitschaftsdatenbank für die Flushoperation wartet.

Eine Überlastung kann darüber hinaus auftreten, wenn die Bereitschaftsdatenbank Protokollsätze anwendet, deren Anwendung zeitaufwendig ist, wie zum Beispiel Protokollsätze für eine Datenbank- oder Tabellenreorganisation.

Eine Vergrößerung des Protokollempfangspuffers der Bereitschaftsdatenbank kann helfen, die Wahrscheinlichkeit einer Überlastung zu verringern, obwohl sie möglicherweise nicht alle Ursachen einer Überlastung beseitigen kann. Standardmäßig beträgt die Größe des Protokollempfangspuffers der Bereitschaftsdatenbank das Zweifache der Größe des Protokollschreibpuffers der Primärdatenbank. Der Datenbankkonfigurationsparameter **logbufsz** definiert die Größe des Protokollschreibpuffers der Primärdatenbank. Mithilfe der DB2-Registrierdatenbankvariablen **DB2_HADR_BUF_SIZE** kann die Größe des Protokollempfangspuffers der Bereitschaftsdatenbank optimal eingestellt werden.

Registrierdatenbankvariable DB2_HADR_PEER_WAIT_LIMIT

Ab DB2 Version 9.5 Fixpack 1 wird durch das Setzen der Registrierdatenbankvariablen **DB2_HADR_PEER_WAIT_LIMIT** für die HADR-Primärdatenbank der Peerstatus aufgehoben, wenn die Protokollierung in der Primärdatenbank für die angegebene Anzahl von Sekunden blockiert wurde, weil das Protokoll in die Bereitschaftsdatenbank repliziert wird. Wenn dieser Grenzwert erreicht wird, unterbricht die Primärdatenbank die Verbindung zur Bereitschaftsdatenbank. Wenn das Peerfenster inaktiviert ist, wird die Primärdatenbank in den Unterbrechungsstatus versetzt, und die Protokollierung wird wieder aufgenommen. Wenn das Peerfenster aktiviert ist, wird die Primärdatenbank in den Status 'Getrennter Peer' versetzt, in dem die Protokollierung weiterhin blockiert ist. Der Status 'Getrennter Peer' wird für die Primärdatenbank aufgehoben, sobald die Verbindung erneut hergestellt wird oder das Peerfenster abgelaufen ist. Die Protokollierung wird wieder aufgenommen, sobald der Status 'Getrennter Peer' für die Primärdatenbank aufgehoben ist.

Die Berücksichtigung der Statusänderung für das Peerfenster beim Aufheben des Peerstatus stellt die Peerfenstersemantik für eine sichere Übernahme in allen Fällen sicher.

Bei einem Ausfall der Primärdatenbank während der Statusänderung gilt weiterhin der normale Peerfensterschutz (sichere Übernahme von der Bereitschaftsdatenbank, während sie sich im Status 'Getrennter Peer' befindet).

Registrierdatenbankvariablen DB2_HADR_SOSNDBUF und DB2_HADR_SORCVBUF

Um eine maximale Netz- und HADR-Leistung zu erreichen, ist möglicherweise eine Optimierung der Werte für die TCP-Socketpuffergröße erforderlich. Die HADR-Protokollübertragungsauslastung, die Netzbandbreite und die Übertragungsverzögerung sind wichtige Faktoren, die hierbei berücksichtigt werden müssen. Vor DB2 Version 9.5 Fixpack 2 konnte eine Änderung der TCP-Socketpuffergröße für HADR-Verbindungen ausschließlich auf Systemebene vorgenommen werden, und die Einstellungen waren für alle TCP-Verbindungen auf der Maschine gültig. Durch die Definition eines hohen Werts für die Socketpuffergröße auf Systemebene wird eine große Speichermenge belegt.

Die beiden Registrierdatenbankvariablen **DB2_HADR_SOSNDBUF** und **DB2_HADR_SORCVBUF** stehen ab Fixpack 2 zur Verfügung. Mit diesen können Sie die TCP-Socketsende- und -empfangspuffergröße speziell für HADR-Verbindungen optimieren. Der Wertebereich dieser beiden Variablen ist 1024 bis 4294967295. Der Standardwert ist die Socketpuffergröße des Betriebssystems, die je nach Betriebssystem variiert. Auf manchen Betriebssystemen wird der vom Benutzer angegebenen Wert automatisch gerundet oder begrenzt.

Cluster-Manager und HADR (High Availability Disaster Recovery)

Sie können DB2-HADR-Datenbanken auf den Knoten eines Clusters implementieren und einen Cluster-Manager verwenden, um die Verfügbarkeit Ihrer Datenbanklösung zu verbessern. Sie können die Primärdatenbank und die Bereitschaftsdatenbank entweder von demselben oder von unterschiedlichen Cluster-Managern verwalten lassen.

Konfigurieren eines HADR-Paares, bei dem die Primär- und die Bereitschaftsdatenbank von demselben Cluster-Manager verwaltet werden

Diese Konfiguration eignet sich am besten für Umgebungen, in denen sich die Primärdatenbank und die Bereitschaftsdatenbank am selben Standort befinden und die schnellstmögliche Funktionsübernahme erforderlich ist. Das Verwenden von HADR bietet diesen Umgebungen den Vorteil, dass die DBMS-Verfügbarkeit gewährleistet wird statt Recovery nach einem Systemabsturz oder andere Recoverymethoden zu verwenden.

Mit dem Cluster-Manager können Sie auf schnelle Weise Probleme aufspüren und eine Übernahmeoperation einleiten. Da HADR separaten Speicher für das DBMS benötigt, sollte der Cluster-Manager mit separater Datenträgersteuerung konfiguriert werden. Diese Konfiguration verhindert, dass der Cluster-Manager auf das Ausführen einer Funktionsübernahme auf dem Datenträger wartet, bevor er das DBMS auf dem Bereitschaftssystem verwendet. Sie können die automatische Clientweiterleitung verwenden, um Clientanwendungen an die neue Primärdatenbank umzuleiten.

Konfigurieren eines HADR-Paares, bei dem die Primär- und die Bereitschaftsdatenbank von unterschiedlichen Cluster-Managern verwaltet werden

Diese Konfiguration eignet sich am besten für Umgebungen, in denen sich die Primärdatenbank und die Bereitschaftsdatenbank an verschiedenen Standorten befinden und im Fall eines vollständigen Ausfalls eines Standorts hohe Verfügbarkeit für die Recovery nach einem Katastrophenfall erforderlich ist. Sie können diese Konfiguration auf verschiedene Weise implementieren. Wenn eine Primärdatenbank oder eine Bereitschaftsdatenbank Teil eines Clusters ist, sind zwei Funktionsübernahmeszenarios möglich.

- Bei einem teilweisen Ausfall eines Standorts können Sie eine Clusterfunktionsübernahme ausführen, sofern ein Knoten verfügbar blieb, den das DBMS für die Funktionsübernahme verwenden kann. In diesem Fall wird die Funktionsübernahme für IP-Adresse und Datenträger mit dem Cluster-Manager ausgeführt, und HADR ist nicht betroffen.
- Bei einem vollständigen Ausfall eines Standorts, an dem sich die Primärdatenbank befindet, können Sie mit HADR die DBMS-Verfügbarkeit gewährleisten, indem Sie eine Übernahmeoperation einleiten. Bei einem vollständigen Ausfall eines Standorts, an dem sich die Bereitschaftsdatenbank befindet, können Sie eine Reparatur des Standorts ausführen oder die Bereitschaftsdatenbank an einen anderen Standort versetzen.

Initialisieren einer Bereitschaftsdatenbank

Eine der Strategien für eine Datenbanklösung mit hoher Verfügbarkeit besteht darin, eine primäre Datenbank für die Verarbeitung von Benutzeranforderungen zu betreiben, sowie eine sekundäre bzw. Bereitschaftsdatenbank, die die Datenbankoperationen für die Primärdatenbank übernehmen kann, falls diese ausfällt. Für das Initialisieren der Bereitschaftsdatenbank ist das Kopieren der Primärdatenbank in die Bereitschaftsdatenbank erforderlich.

Die Bereitschaftsmaschine kann auf verschiedene Weisen initialisiert werden. Beispiel:

- Verwenden Sie die Plattenspiegelung, um die Primärdatenbank zu kopieren, und verwenden Sie die DB2-Datenbankunterstützung für ausgesetzte E/A, um die Spiegeldatenbank zu teilen und damit die zweite Datenbank zu erstellen.
- Erstellen Sie ein Backup-Image der Primärdatenbank und führen Sie für dieses Image eine Recovery in der Bereitschaftsdatenbank aus.
- Verwenden Sie die SQL Replication, um Daten aus der Primärdatenbank zu erfassen und sie in der Bereitschaftsdatenbank anzuwenden.

Nach dem Initialisieren der Bereitschaftsdatenbank müssen Sie Ihre Datenbanklösung so konfigurieren, dass die Primärdatenbank und die Bereitschaftsdatenbank synchronisiert werden, sodass die Bereitschaftsdatenbank die Operationen für die Primärdatenbank übernehmen kann, falls diese ausfällt.

Verwenden einer geteilten Spiegeldatenbank als Bereitschaftsdatenbank

Verwenden Sie die folgende Prozedur, um eine geteilte Spiegeldatenbank einer Datenbank zu erstellen, die als Bereitschaftsdatenbank verwendet werden soll. Wenn in der Primärdatenbank ein Fehler auftritt und eine Recovery nach einem Systemabsturz erforderlich wird, können Sie die Bereitschaftsdatenbank die Rolle der Primärdatenbank übernehmen lassen.

Gehen Sie wie folgt vor, um eine geteilte Spiegeldatenbank als Bereitschaftsdatenbank zu verwenden:

1. Setzen Sie für die primäre Datenbank die E/A aus:

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

   Führen Sie keine weiteren Dienstprogramme oder Tools aus, während sich die Datenbank im ausgesetzten Status befindet. Das Erstellen einer Datenbankkopie sollte die einzige Aktivität sein.
2. Verwenden Sie geeignete Befehle auf Betriebssystemebene, um die Spiegeldatenbank(en) aus der primären Datenbank zu erstellen.

   Anmerkung: Stellen Sie sicher, dass Sie das gesamte Datenbankverzeichnis einschließlich des Datenträgerverzeichnisses kopieren. Sie müssen auch das Protokollverzeichnis und alle Containerverzeichnisse kopieren, die außerhalb des Datenbankverzeichnisses vorhanden sind. Eine Zusammenstellung dieser Informationen finden Sie in der Verwaltungssicht DBPATHS, in der alle Dateien und Verzeichnisse der Datenbank angezeigt werden, die geteilt werden müssen.
3. Nehmen Sie den E/A-Betrieb auf der primären Datenbank wieder auf:

   ```
   db2 set write resume for database
   ```
4. Katalogisieren Sie die gespiegelte Datenbank auf dem sekundären System.

   Anmerkung: Eine gespiegelte Datenbank kann standardmäßig nicht in demselben System vorhanden sein wie die Primärdatenbank. Sie muss sich auf einem sekundären System mit der gleichen Verzeichnisstruktur befinden und denselben Instanznamen als primäre Datenbank verwenden. Wenn die gespiegelte Datenbank auf demselben System als Primärdatenbank vorhanden sein muss, können Sie das Dienstprogramm db2relocatedb oder die Option RELOCATE USING des Befehls db2inidb für diese Aufgabe verwenden.
5. Starten Sie die Datenbankinstanz auf dem sekundären System:

   ```
   db2start
   ```
6. Initialisieren Sie die gespiegelte Datenbank auf dem sekundären System, indem Sie sie in den Status für anstehende aktualisierende Recovery versetzen:

   ```
   db2inidb aliasname-der-datenbank as standby
   ```

   Falls erforderlich, geben Sie die Option RELOCATE USING des Befehls d2inidb an, um die Bereitschaftsdatenbank zu verlagern:

   ```
   db2inidb aliasname-der-datenbank as standby relocate using relocatedbcfg.txt
   ```

   'relocatedbcfg.txt' ist die Datei, die die zum Verlagern der Datenbank erforderlichen Informationen enthält.

   Anmerkung:

   a. Sie können ein vollständiges Datenbankbackup mit geteilter Spiegeldatenbank durchführen, wenn Sie über DMS-Tabellenbereiche oder Tabellenbereiche des dynamischen Speichers verfügen. Das Durchführen eines Backups mit geteilter Spiegeldatenbank ermöglicht es, den Systemaufwand für ein Backup der Produktionsdatenbank zu vermeiden. Wenn Sie dieses Backup-Image zu einem späteren Zeitpunkt wiederherstellen und für die wiederhergestellte Datenbank eine aktualisierende Recovery bis zum Ende der Protokolle vornehmen, wird der Fehler SQL4970 ausgegeben, es sei denn, es liegen Protokollsätze vor, deren Zeitmarke nach dem Zeitpunkt von SET WRITE SUSPEND liegt. Wenn keine neueren Protokolldateien von der ursprünglichen Datenbank auf die wiederhergestellte Datenbank ko-

piert werden oder wenn die kopierte Protokolldatei leer ist (also keine neueren Protokollsätze enthält), wird während der aktualisierenden Wiederherstellung der Fehler SQL4970 ausgegeben.

b. Das Datenbankverzeichnis (einschließlich des Datenträgerverzeichnisses), das Protokollverzeichnis und die Containerverzeichnisse müssen an die gewünschte Position versetzt werden, bevor Sie die Option RELOCATE USING verwenden.

7. Definieren Sie ein Benutzerexitprogramm, um die Protokolldateien vom primären System abzurufen.
8. Stellen Sie die Datenbank bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt aktualisierend wieder her.
9. Rufen Sie weiter Protokolldateien ab, und stellen Sie die Datenbank anhand der Protokolle aktualisierend wieder her, entweder bis zum Ende der Protokolle oder bis zu dem für die Bereitschaftsdatenbank erforderlichen Zeitpunkt.
10. Wenn Sie die Bereitschaftsdatenbank in den Onlinestatus versetzen wollen, setzen Sie den Befehl ROLLFORWARD mit der Option STOP ab.

    Anmerkung: Die Protokolle der primären Datenbank können nicht mehr auf die gespiegelte Datenbank angewandt werden, nachdem für sie der Status für anstehende aktualisierende Recovery aufgehoben wurde.

Synchronisationsmodus von DB2 HADR (High Availability Disaster Recovery) konfigurieren

Der Konfigurationsparameter HADR_SYNCMODE legt den Schutzgrad fest, den Ihre DB2-HADR-Datenbanklösung vor Transaktionsverlusten haben soll. Der Synchronisationsmodus legt fest, wann der primäre Datenbankserver eine Transaktion als abgeschlossen betrachtet. Dies basiert auf dem Status der Protokollierung auf der Bereitschaftsdatenbank. Je strikter der Wert des Konfigurationsparameters für den Synchronisationsmodus, desto mehr Schutz weist Ihre Datenbanklösung vor Verlusten von Transaktionsdaten auf, aber desto langsamer wird auch die Transaktionsverarbeitungsleistung. Sie müssen also die Schutzerfordernisse im Hinblick auf Transaktionsverluste gegen die Leistungserfordernisse abwägen.

Diese Modi können nur dann angewendet werden, wenn sich die Primär- und die Bereitschaftsdatenbank im Peerstatus oder im Status 'Unterbrochener Peer' befinden.

Verwenden Sie den Konfigurationsparameter HADR_SYNCMODE, um den Synchronisationsmodus festzulegen. Gültige Werte:

SYNC (synchron)
Dieser Modus bietet den größten Schutz vor Datenverlust, benötigt jedoch von allen drei Modi die längste Transaktionsantwortzeit.

In diesem Modus wird das Schreiben von Protokollen nur dann als erfolgreich betrachtet, wenn die Protokolle in Protokolldateien in der Primärdatenbank geschrieben wurden und wenn die Primärdatenbank von der Bereitschaftsdatenbank eine Empfangsbestätigung erhalten hat, dass die Protokolle auch in Protokolldateien in der Bereitschaftsdatenbank geschrieben wurden. So wird sichergestellt, dass die Protokolldaten an beiden Standorten gespeichert werden.

Wenn die Bereitschaftsdatenbank ausfällt, bevor sie die Protokollsätze nachvollziehen kann, kann sie bei ihrem nächsten Start die Protokollsätze aus ihren lokalen Protokolldateien abrufen und nachvollziehen. Wenn die

Primärdatenbank ausfällt, wird durch eine Funktionsübernahme durch die Bereitschaftsdatenbank sichergestellt, dass alle in der Primärdatenbank festgeschriebenen Transaktionen auch in der Bereitschaftsdatenbank festgeschrieben wurden. Wenn der Client nach der Funktionsübernahmeoperation eine Verbindung zur neuen Primärdatenbank herstellt, können in der neuen Primärdatenbank Transaktionen festgeschrieben sein, die an die Anwendung auf der ursprünglichen Primärdatenbank nie als festgeschrieben gemeldet wurden. Dies geschieht, wenn die Primärdatenbank ausfällt, bevor sie eine Empfangsbestätigungsnachricht von der Bereitschaftsdatenbank verarbeiten konnte. Clientanwendungen sollten unter Umständen die Datenbank abfragen, um zu ermitteln, ob solche Transaktionen vorhanden sind.

Wenn die Primärdatenbank die Verbindung mit der Bereitschaftsdatenbank verliert, ist die weitere Vorgehensweise von der Konfiguration des Datenbankkonfigurationsparameters hadr_peer_window abhängig. Wenn hadr_peer_window auf einen Zeitwert ungleich Null gesetzt ist, wechselt die Primärdatenbank nach dem Verlust der Verbindung mit der Bereitschaftsdatenbank in den Status 'Unterbrochener Peer' und wartet weiterhin auf eine Bestätigung von der Bereitschaftsdatenbank, bevor Transaktionen festgeschrieben werden. Wenn der Datenbankkonfigurationsparameter hadr_peer_window auf Null gesetzt ist, wird nicht länger davon ausgegangen, dass sich die Primär- und die Bereitschaftsdatenbank im Peerstatus befinden. Transaktionen werden nicht länger zurückgehalten, um auf eine Bestätigung von der Bereitschaftsdatenbank zu warten. Wird die Funktionsübernahmeoperation ausgeführt, obwohl sich die Datenbanken nicht im Peerstatus oder im Status 'Unterbrochener Peer' befinden, gibt es keine Gewährleistung, dass alle in der Primärdatenbank festgeschriebenen Transaktionen auch in der Bereitschaftsdatenbank vorhanden sind.

Fällt die Primärdatenbank aus, während sich die Datenbanken im Peerstatus bzw. im Status 'Unterbrochener Peer' befinden, kann sie nach einer Funktionsübernahmeoperation dem HADR-Paar als Bereitschaftsdatenbank hinzugefügt werden. Da eine Transaktion erst dann als festgeschrieben betrachtet wird, wenn die Primärdatenbank eine Empfangsbestätigung von der Bereitschaftsdatenbank erhält, dass auch die Protokolle in Protokolldateien in der Bereitschaftsdatenbank geschrieben wurden, stimmt die Protokollfolge der Primärdatenbank mit der Protokollfolge der Bereitschaftsdatenbank überein. Die ursprüngliche Primärdatenbank (die nun als Bereitschaftsdatenbank fungiert) muss lediglich die neuen Protokollsätze nachvollziehen, die seit der Funktionsübernahmeoperation in der neuen Primärdatenbank generiert wurden.

Wenn sich die Primärdatenbank zum Zeitpunkt ihres Ausfalls nicht im Peerstatus befand, können Unterschiede zwischen ihrer Protokollfolge und der Protokollfolge der Bereitschaftsdatenbank bestehen. Wird eine Funktionsübernahmeoperation erforderlich, können diese Unterschiede dadurch entstehen, dass die Bereitschaftsdatenbank nach der Funktionsübernahme eine eigene Protokollfolge startet. Da einige Operationen nicht rückgängig gemacht werden können (z. B. das Löschen einer Tabelle), ist es nicht möglich, die Primärdatenbank auf den Zeitpunkt zurückzusetzen, zu dem die neue Protokollfolge erstellt wurde. Wenn die Protokollfolgen nicht übereinstimmen und Sie den Befehl START HADR mit der Option STANDBY in der ursprünglichen Primärdatenbank ausführen, empfangen Sie eine Nachricht, dass der Befehl erfolgreich war. Allerdings wird diese Nachricht ausgegeben, bevor die Reintegration versucht wird. Wenn die Reintegration fehlschlägt, werden Nachrichten über die Prüfung des Paares an das Ver-

waltungsprotokoll und das Diagnoseprotokoll in der Primärdatenbank und der Bereitschaftsdaten ausgegeben. Die reintegrierte Bereitschaftsdatenbank bleibt Bereitschaftsdatenbank, jedoch weist die Primärdatenbank die Bereitschaftsdatenbank bei der Prüfung des Paares zurück, sodass die Bereitschaftsdatenbank herunterfährt. Wenn die ursprüngliche Primärdatenbank dem HADR-Paar erneut erfolgreich hinzugefügt wurde, können Sie die Datenbank zurücksetzen, indem Sie den Befehl TAKEOVER HADR ohne die Option BY FORCE absetzen. Wenn die ursprüngliche Primärdatenbank dem HADR-Paar nicht wieder hinzugefügt werden kann, können Sie sie reinitialisieren, indem Sie ein Backup-Image der neuen Primärdatenbank wiederherstellen.

NEARSYNC (fast synchron)

Dieser Modus hat eine kürzere Transaktionsantwortzeit als der Modus SYNC, er bietet aber auch einen etwas geringeren Schutz vor Transaktionsverlust.

In diesem Modus wird das Schreiben von Protokollen nur dann als erfolgreich betrachtet, wenn die Protokolle in Protokolldateien in der Primärdatenbank geschrieben wurden und wenn die Primärdatenbank von der Bereitschaftsdatenbank eine Empfangsbestätigung erhalten hat, dass die Protokolle auch in den Hauptspeicher des Bereitschaftssystems geschrieben wurden. Zu Datenverlust kann es nur dann kommen, wenn beide Standorte gleichzeitig ausfallen und wenn der Zielstandort nicht alle empfangenen Protokolldaten an nicht flüchtigen Speicher übertragen hat.

Wenn die Bereitschaftsdatenbank ausfällt, bevor sie die Protokollsätze aus dem Speicher auf Platte kopieren kann, gehen diese Protokollsätze in der Bereitschaftsdatenbank verloren. Normalerweise kann die Bereitschaftsdatenbank bei ihrem nächsten Start die fehlenden Protokollsätze aus der Primärdatenbank abrufen. Wenn jedoch in der Primärdatenbank ein Fehler auftritt oder das Netz das Abrufen der Protokollsätze unmöglich macht, werden die Protokollsätze ebenso wie die ihnen zugeordneten Transaktionen nie in der Bereitschaftsdatenbank angezeigt.

Gehen Transaktionen verloren, ist die neue Primärdatenbank nach einer Funktionsübernahme nicht mit der ursprünglichen Primärdatenbank identisch. Clientanwendungen sollten diese Transaktionen gegebenenfalls erneut übergeben, um den Anwendungsstatus zu aktualisieren.

Fällt die Primärdatenbank aus, während sich die Primärdatenbank und die Bereitschaftsdatenbank im Peerstatus befinden, ist es möglich, dass die ursprüngliche Primärdatenbank dem HADR-Paar nach einer Funktionsübernahmeoperation nur dann als Bereitschaftsdatenbank hinzugefügt werden kann, wenn sie mit einer Operation zum vollständigen Restore reinitialisiert wurde. Wenn die Funktionsübernahme auch verlorene Protokollsätze umfasst (bei einem Ausfall beider Datenbanken), stimmen die Protokollfolge der Primärdatenbank und der Bereitschaftsdatenbank nicht überein, sodass Versuche, die ursprüngliche Primärdatenbank als Bereitschaftsdatenbank erneut zu starten, fehlschlagen, wenn nicht zuerst eine Restoreoperation durchgeführt wird. Wenn die ursprüngliche Primärdatenbank dem HADR-Paar erneut erfolgreich hinzugefügt wurde, können Sie die Datenbank zurücksetzen, indem Sie den Befehl TAKEOVER HADR ohne die Option BY FORCE absetzen. Wenn die ursprüngliche Primärdatenbank dem HADR-Paar nicht wieder hinzugefügt werden kann, können Sie sie reinitialisieren, indem Sie ein Backup-Image der neuen Primärdatenbank wiederherstellen.

ASYNC (asynchron)

In diesem Modus ist das Risiko eines Transaktionsverlusts bei einem Ausfall des primären Systems am größten. Dieser Modus hat von den drei Modi jedoch die kürzeste Transaktionsantwortzeit.

In diesem Modus wird das Schreiben von Protokollen nur dann als erfolgreich betrachtet, wenn die Protokolle in Protokolldateien in der Primärdatenbank geschrieben und an die TCP-Schicht der Hostmaschine des primären Systems übergeben wurden. Da das primäre System nicht auf eine Empfangsbestätigung vom Bereitschaftssystem wartet, können Transaktionen bereits als festgeschrieben betrachtet werden, während sie noch an das Bereitschaftssystem weitergeleitet werden.

Bei einem Ausfall der Hostmaschine der Primärdatenbank, des Netzes oder der Bereitschaftsdatenbank können Protokollsätze, die zu diesem Zeitpunkt gerade weitergeleitet werden, verloren gehen. Steht die Primärdatenbank zur Verfügung, können die fehlenden Protokollsätze erneut an die Bereitschaftsdatenbank übertragen werden, wenn die Verbindung zwischen den beiden Datenbanken wiederhergestellt wird. Wird jedoch eine Funktionsübernahme erforderlich, während Protokollsätze fehlen, erreichen diese Protokollsätze nie die Bereitschaftsdatenbank, sodass die entsprechenden Transaktionen bei der Funktionsübernahme verloren gehen.

Gehen Transaktionen verloren, ist die neue Primärdatenbank nach einer Funktionsübernahme nicht mit der ursprünglichen Primärdatenbank identisch. Clientanwendungen sollten diese Transaktionen gegebenenfalls erneut übergeben, um den Anwendungsstatus zu aktualisieren.

Fällt die Primärdatenbank aus, während sich die Primärdatenbank und die Bereitschaftsdatenbank im Peerstatus befinden, ist es möglich, dass die ursprüngliche Primärdatenbank dem HADR-Paar nach einer Funktionsübernahmeoperation nur dann als Bereitschaftsdatenbank hinzugefügt werden kann, wenn sie einer vollständigen Restoreoperation reinitialisiert wurde. Wenn die Funktionsübernahme auch verlorene Protokollsätze umfasst, stimmen die Protokollfolge der Primärdatenbank und der Bereitschaftsdatenbank nicht überein, sodass Versuche, die ursprüngliche Primärdatenbank als Bereitschaftsdatenbank erneut zu starten, fehlschlagen, wenn nicht zuerst eine Restoreoperation ausgeführt wird. Da beim Auftreten einer Funktionsübernahme in asynchronem Modus ein Verlust von Protokollsätzen wahrscheinlicher ist, ist auch die Wahrscheinlichkeit größer, dass die Primärdatenbank dem HADR-Paar nicht wieder hinzugefügt werden kann. Wenn die ursprüngliche Primärdatenbank dem HADR-Paar erneut erfolgreich hinzugefügt wurde, können Sie die Datenbank zurücksetzen, indem Sie den Befehl TAKEOVER HADR ohne die Option BY FORCE absetzen. Wenn die ursprüngliche Primärdatenbank dem HADR-Paar nicht wieder hinzugefügt werden kann, können Sie sie reinitialisieren, indem Sie ein Backup-Image der neuen Primärdatenbank wiederherstellen.

Unterstützung für High Availability Disaster Recovery (HADR)

Um eine optimale Leistung mit der DB2 High Availability Disaster Recovery (HADR) zu erreichen, müssen Sie Systemanforderungen und Funktionseinschränkungen berücksichtigen, wenn Sie Ihre Datenbanklösung mit hoher Verfügbarkeit entwerfen.

Systemvoraussetzungen für High Availability Disaster Recovery (HADR)

Um eine optimale Leistung der High Availability Disaster Recovery (HADR) zu erreichen, sollten Sie sicherstellen, dass Ihr System die folgenden Voraussetzungen im Hinblick auf die Hardware, Betriebssysteme und das DB2-Datenbanksystem erfüllt.

Empfehlung: Verwenden Sie aus Leistungsgründen für die Systeme, auf denen sich die Primärdatenbank und die Bereitschaftsdatenbank befinden, die gleiche Hardware und Software. Wenn das System, auf dem sich die Bereitschaftsdatenbank befindet, über weniger Ressourcen verfügt, als das System, auf dem sich die Primärdatenbank befindet, ist es möglich, dass die Bereitschaftsdatenbank die von der Primärdatenbank generierte Transaktionslast nicht bewältigen kann. Dies kann dazu führen, dass die Bereitschaftsdatenbank nicht auf dem aktuellsten Stand bleibt bzw. dass die Leistung der Primärdatenbank verringert wird. Im Fall einer Funktionsübernahme sollte die neue Primärdatenbank über die erforderlichen Ressourcen verfügen, um die Clientanwendungen adäquat bedienen zu können.

Hardware- und Betriebssystemvoraussetzungen

Empfehlung: Verwenden Sie für die Primärdatenbank und die Bereitschaftsdatenbank identische Hostcomputer. Das heißt, die Computer sollten denselben Hersteller und dieselbe Architektur haben.

Das Betriebssystem für die Primärdatenbank und die Bereitschaftsdatenbank sollte dieselbe Version einschließlich der Programmkorrekturen haben. Diese Vorgabe können Sie während eines schrittweisen Upgrades vorübergehend außer Acht lassen, Sie sollten dabei jedoch äußerst vorsichtig vorgehen.

Zwischen den HADR-Hostmaschinen muss eine TCP/IP-Schnittstelle zur Verfügung stehen, und es wird ein Hochgeschwindigkeitsnetz mit hoher Kapazität empfohlen.

Voraussetzungen für DB2-Datenbanken

Die Datenbanksystemversionen für die Primär- und die Bereitschaftsdatenbank müssen identisch sein; es muss sich beispielsweise für beide um Version 8 oder Version 9 handeln. Bei schrittweisen Upgrades kann die Modifikationsstufe (z. B. die Fixpackstufe) des Datenbanksystems für die Bereitschaftsdatenbank kurzzeitig zum Testen der neuen Stufe neuer als die der Primärdatenbank sein. Allerdings sollte diese Konfiguration nicht über einen längeren Zeitraum verwendet werden. Die Primär- und die Bereitschaftsdatenbank stellen keine Verbindung zueinander her, wenn die Modifikationsstufe des Datenbanksystems für eine Primärdatenbank neuer als die für die Bereitschaftsdatenbank ist.

Die DB2-Datenbanksoftware muss für die Primärdatenbank und die Bereitschaftsdatenbank über dieselbe Bitgröße verfügen (32 oder 64 Bit). Tabellenbereiche und ihre Container müssen in der Primär- und der Bereitschaftsdatenbank identisch sein. Andere Merkmale, die ebenfalls identisch sein müssen, sind der Tabellenbereichstyp (DMS oder SMS), die Tabellenbereichsgröße, der Containerpfad und der Dateityp des Containers (Roheinheit oder Dateisystem). Der den Protokolldateien zugewiesene Speicherbereich sollte außerdem in der Primär- und der Bereitschaftsdatenbank gleich groß sein.

Wenn Sie für die Primärdatenbank eine Tabellenbereichsanweisung ausgeben, wie z. B. CREATE TABLESPACE, ALTER TABLESPACE oder DROP TABLESPACE, wird sie auch auf die Bereitschaftsdatenbank angewendet. Sie müssen sicherstellen, dass die einbezogenen Einheiten in beiden Datenbanken definiert sind, bevor Sie die Tabellenbereichsanweisung in der Primärdatenbank absetzen.

Wenn Sie einen Tabellenbereich für die Primärdatenbank erstellen und die Anwendung von Protokollen in der Bereitschaftsdatenbank fehlschlägt, weil die entsprechenden Container nicht verfügbar sind, empfängt die Primärdatenbank keine Fehlernachricht darüber, dass die Anwendung der Protokolle fehlgeschlagen ist.

Zur Überprüfung, ob beim Anwenden der Protokolle Fehler auftreten, müssen Sie die Datei 'db2diag.log' und das Verwaltungsprotokoll für die Bereitschaftsdatenbank überwachen, wenn Sie neue Tabellenbereiche erstellen.

Wenn eine Übernahmeoperation stattfindet, steht dieser Tabellenbereich für die neue Primärdatenbank nicht zur Verfügung. Um dies zu beheben, führen Sie einen Restore des Tabellenbereichs in der neuen Primärdatenbank von einem Backup-Image durch.

Im folgenden Beispiel wird der Tabellenbereich MEIN_TABELLENBEREICH für die Datenbank MEINE_DATENBANK mit Restore wiederhergestellt, bevor er als Teil der neuen Primärdatenbank eingesetzt wird:

1. `db2 connect to meine_datenbank`
2. `db2 list tablespaces show detail`

   Anmerkung: Führen Sie den Befehl db2 list tablespaces show detail aus, um den Status aller Tabellenbereiche anzuzeigen und die für Schritt 5 erforderliche Tabellenbereichs-ID abzurufen.
3. `db2 stop hadr on database meine_datenbank`
4. `db2 "restore database meine_datenbank tablespace (mein_tabellenbereich) online redirect"`
5. `db2 "set tablespace containers for ID_nr_meines_tabellenbereichs ignore rollforward container operations using (path '/neuer_containerpfad/')"`
6. `db2 "restore database meine_datenbank continue"`
7. `db2 rollforward database meine_datenbank to end of logs and stop tablespace "(mein_tabellenbereich)"`
8. `db2 start hadr on database meine_datenbank as primary`

Die Primär- und die Bereitschaftsdatenbank müssen jedoch nicht denselben Datenbankpfad aufweisen. Werden relative Containerpfade verwendet, wird derselbe relative Pfad möglicherweise unterschiedlichen absoluten Containerpfaden in der Primär- und der Bereitschaftsdatenbank zugeordnet.

Datenbanken mit dynamischem Speicher werden von HADR vollständig unterstützt, einschließlich Replikation der Anweisung ALTER DATABASE mit der Klausel ADD STORAGE ON. Ähnlich wie bei Tabellenbereichscontainern muss der Speicherpfad sowohl in der Primärdatenbank als auch in der Bereitschaftsdatenbank vorhanden sein.

Die Primär- und die Bereitschaftsdatenbank müssen denselben Datenbanknamen besitzen. Dies bedeutet, dass sie sich in unterschiedlichen Instanzen befinden müssen.

Ein umgeleiteter Restore wird nicht unterstützt. Das heißt, HADR unterstützt keine Umleitung von Tabellenbereichscontainern. Änderungen am Datenbankverzeichnis und am Protokollverzeichnis werden jedoch unterstützt. Tabellenbereichscontainer, die in relativen Pfaden erstellt wurden, lassen sich in Pfaden relativ zum neuen Datenbankverzeichnis wiederherstellen.

Pufferpoolvoraussetzungen

Da Pufferpooloperationen auch auf die Bereitschaftsdatenbank angewendet werden, ist es wichtig, dass die Primärdatenbank und die Bereitschaftsdatenbank dieselbe Speicherkapazität haben.

Installations- und Speichervoraussetzungen für HADR (High Availability Disaster Recovery)

Um eine optimale Leistung mit der High Availability Disaster Recovery (HADR) zu erreichen, sollten Sie sicherstellen, dass Ihr System die folgenden Voraussetzungen im Hinblick auf die Installation und das Speichern erfüllt.

Installationsvoraussetzungen

Für HADR sollten die Instanzpfade in der Primär- und der Bereitschaftsdatenbank übereinstimmen. Eine Verwendung unterschiedlicher Instanzpfade kann in einigen Fällen Probleme verursachen, zum Beispiel wenn eine gespeicherte SQL-Prozedur eine benutzerdefinierte Funktion (UDF) aufruft und ein identischer Pfad zum UDF-Objektcode auf beiden Servern, das heißt dem primären Server und dem Bereitschaftsserver erwartet wird.

Speicheranforderungen

Datenbanken mit dynamischem Speicher werden von HADR vollständig unterstützt, einschließlich Replikation der Anweisung ALTER DATABASE mit der Klausel ADD STORAGE ON. Ähnlich wie bei Tabellenbereichscontainern muss der Speicherpfad sowohl in der Primärdatenbank als auch in der Bereitschaftsdatenbank vorhanden sein. Symbolische Links können zur Erstellung identischer Pfade verwendet werden. Die Primärdatenbank und die Bereitschaftsdatenbank können sich auf demselben Computer befinden. Obwohl ihre Datenbankspeicher mit demselben Pfad beginnen, geraten sie nicht in Konflikt, weil die tatsächlich verwendeten Verzeichnisse die Instanznamen in ihren Pfaden enthalten (da die Primär- und die Bereitschaftsdatenbank ja den gleichen Datenbanknamen besitzen müssen, müssen sie sich in verschiedenen Instanzen befinden). Der Speicherpfad wird folgendermaßen formuliert: `speicherpfadname/instanzname/dbpartitionsname/dbname/tabbereichsname/containername`.

Tabellenbereiche und ihre Container müssen in der Primär- und der Bereitschaftsdatenbank identisch sein. Andere Merkmale, die ebenfalls identisch sein müssen, sind der Tabellenbereichstyp (DMS oder SMS), die Tabellenbereichsgröße, der Containerpfad und der Dateityp des Containers (Roheinheit oder Dateisystem). Wenn für die Datenbank der dynamische Speicher aktiviert ist, müssen die Speicherpfade identisch sein. Hierzu gehören die Pfadnamen sowie die Menge des jeweils verfügbaren Speicherplatzes, der für die Datenbank reserviert ist. Der den Protokolldateien zugewiesene Speicherbereich sollte außerdem in der Primär- und der Bereitschaftsdatenbank gleich groß sein.

Wenn Sie für die Primärdatenbank eine Tabellenbereichsanweisung ausgeben, wie z. B. CREATE TABLESPACE, ALTER TABLESPACE oder DROP TABLESPACE, wird sie auch auf die Bereitschaftsdatenbank angewendet. Stellen Sie sicher, dass

die hierfür benötigten Einheiten in beiden Datenbanken eingerichtet sind, bevor Sie die Tabellenbereichsanweisung in der Primärdatenbank eingeben.

Wenn die Tabellenbereichskonfiguration in der Primär- und der Bereitschaftsdatenbank nicht identisch ist, können bei der Anwendung der Protokolle in der Bereitschaftsdatenbank Fehler wie KEIN SPEICHERPLATZ VERFÜGBAR oder TABELLENBEREICHSCONTAINER NICHT GEFUNDEN auftreten. Ebenso werden, wenn für die Datenbanken der dynamische Speicher aktiviert ist und die Speicherpfade nicht identisch sind, die Protokollsätze, die der Klausel ADD STORAGE ON der Anweisung ALTER DATABASE zugeordnet sind, nicht auf die Bereitschaftsdatenbank angewendet. Dies könnte dazu führen, dass für die vorhandenen Speicherpfade bereits frühzeitig kein Speicherplatz mehr auf dem Bereitschaftssystem verfügbar ist und Tabellenbereiche mit dynamischem Speicher nicht vergrößert werden können. Tritt eine dieser Situationen auf, wird der betreffende Tabellenbereich in den Status *aktualisierende Recovery anstehend* gesetzt und bei der folgenden Anwendung der Protokolle ignoriert. Wenn eine Übernahmeoperation stattfindet, steht dieser Tabellenbereich den Anwendungen nicht zur Verfügung.

Wird dieses Problem auf dem Bereitschaftssystem vor einer Übernahme festgestellt, besteht die Lösung darin, die Bereitschaftsdatenbank erneut zu erstellen und dabei die Speicherprobleme zu beheben. Folgende Schritte sind erforderlich:

- Inaktivieren der Bereitschaftsdatenbank.
- Löschen der Bereitschaftsdatenbank.
- Sicherstellen, dass die erforderlichen Dateisysteme vorhanden sind und ausreichend freien Speicherplatz enthalten, damit die folgenden Restore- und aktualisierenden Recoveryoperationen ausgeführt werden können.
- Restore der Datenbank auf dem Bereitschaftssystem mithilfe eines aktuellen Backups der Primärdatenbank (oder Reinitialisieren mithilfe einer geteilten Spiegeldatenbank oder Flash-Kopie und dem Befehl 'db2inidb'). Wenn für die Primärdatenbank der dynamische Speicher aktiviert ist, dürfen die Speicherpfade während des Restores nicht erneut definiert werden. Darüber hinaus dürfen Tabellenbereichscontainer beim Restore nicht umgeleitet werden.
- Erneutes Starten von HADR auf dem Bereitschaftssystem.

Wenn das Problem in der Bereitschaftsdatenbank jedoch erst nach der Übernahme festgestellt wird (oder wenn entschieden wurde, die Speicherprobleme bis zu diesem Zeitpunkt nicht zu beheben), ist die Lösung abhängig von der Art des aufgetretenen Problems.

Wenn für die Datenbank der dynamische Speicher aktiviert ist und in den Speicherpfaden, die der Bereitschaftsdatenbank zugeordnet sind, kein Speicherplatz zur Verfügung steht, führen Sie die folgenden Schritte aus:

1. Stellen Sie Speicherplatz in den Speicherpfaden zur Verfügung, indem Sie die Dateisysteme erweitern, oder indem Sie unnötige Dateien, bei denen es sich nicht um DB2-Dateien handelt, in ihnen entfernen.
2. Führen Sie eine ROLLFORWARD-Operation für den Tabellenbereich bis zum Ende der Protokolle durch.

Falls das Hinzufügen oder Erweitern von Containern als Teil der Anwendung der Protokolle auf die Bereitschaftsdatenbank nicht durchgeführt werden konnte: Wenn die erforderlichen Backup-Images und Protokolldateiarchive verfügbar sind, können Sie den Tabellenbereich möglicherweise wiederherstellen, indem Sie zunächst

die Anweisung SET TABLESPACE CONTAINERS mit der Option IGNORE ROLLFORWARD CONTAINER OPERATIONS und anschließend den Befehl ROLLFORWARD eingeben.

Die Primär- und die Bereitschaftsdatenbank müssen jedoch nicht denselben Datenbankpfad aufweisen. Werden relative Containerpfade verwendet, wird derselbe relative Pfad möglicherweise unterschiedlichen absoluten Containerpfaden in der Primär- und der Bereitschaftsdatenbank zugeordnet. Wenn sich also die Primär- und die Bereitschaftsdatenbank auf demselben Computer befinden, müssen alle Tabellenbereichscontainer mit relativen Pfaden definiert werden, sodass sie verschiedenen Pfaden für die Primärdatenbank und die Bereitschaftsdatenbank zugeordnet werden.

Einschränkungen für High Availability Disaster Recovery (HADR)

Um eine optimale Leistung mit der High Availability Disaster Recovery (HADR) zu erreichen ist es wichtig, die HADR-Einschränkungen bereits in Analyse und Entwurf Ihrer hoch verfügbaren DB2-Datenbanklösung zu berücksichtigen.

Die folgende Liste enthält eine Zusammenfassung der Einschränkungen für HADR:

- HADR wird nicht in einer Umgebung mit partitionierten Datenbanken unterstützt.
- Die Primär- und die Bereitschaftsdatenbanken müssen über dieselbe Betriebssystemversion und dieselbe Version des DB2-Datenbanksystems verfügen, bis auf einen kurzen Zeitraum, während dessen ein schrittweiser Upgrade ausgeführt wird.
- Das Release des DB2-Datenbanksystem muss auf der Primärdatenbank und der Bereitschaftsdatenbank dieselbe Bitgröße haben (32 oder 64 Bit).
- Lesevorgänge in der Bereitschaftsdatenbank werden nicht unterstützt. Clients können keine Verbindung zur Bereitschaftsdatenbank herstellen.
- Protokolldateien werden nur von der Primärdatenbank archiviert.
- Self Tuning Memory Manager (STMM) kann jeweils nur für die Primärdatenbank ausgeführt werden. Nach dem Starten der Primärdatenbank oder dem Konvertieren einer Bereitschaftsdatenbank in eine Primärdatenbank durch Übernahme wird die STMM-EDU (Self Tuning Memory Manager Engine-Dispatchable-Unit) möglicherweise erst bei der ersten ankommenden Clientverbindungsanforderung gestartet.
- Backup-Operationen in der Bereitschaftsdatenbank werden nicht unterstützt.
- Nicht protokollierte Operationen, wie Änderungen an Datenbankkonfigurationsparametern und der Datei des Recoveryprotokolls, werden nicht in die Bereitschaftsdatenbank repliziert.
- Ladeoperationen mit der Option COPY NO werden nicht unterstützt.
- HADR unterstützt die Verwendung einer unformatierten Ein-/Ausgabe (direkter Plattenzugriff) für Datenbankprotokolldateien nicht. Wenn HADR durch den Befehl START HADR gestartet wird oder die Datenbank mit konfiguriertem HADR aktiviert (erneut gestartet) wird und Protokolle auf Roheinheiten festgestellt werden, schlägt der entsprechende Befehl fehl.
- Bei einem einphasigen Commit kann eine HADR-Datenbank als Server mit föderierten Datenbanken (Transaktionsmanager) oder als Datenquelle (Ressourcenmanager) fungieren. Bei einem zweiphasigen Commit kann eine HADR-Datenbank nur als Datenquelle fungieren.
- HADR unterstützt keine Endlosprotokollierung.

Terminieren von Verwaltungs- und Wartungsaktivitäten für hohe Verfügbarkeit

Ihre DB2-Datenbanklösung erfordert regelmäßige Verwaltung und Wartung. Dazu gehören Aktivitäten wie Software- oder Hardware-Upgrades, Optimierung der Datenbankleistung, Datenbankbackups, Statistikerfassung und Überwachung zu Geschäftszwecken. Sie müssen die Beeinträchtigung der Verfügbarkeit Ihrer Datenbanklösung durch diese Verwaltungs- und Wartungsaktivitäten minimieren.

Vorbereitung

Bevor Sie Ihre Verwaltungs- und Wartungsaktivitäten terminieren können, müssen Sie feststellen, welche dieser Aktivitäten für Ihre Datenbanklösung ausgeführt werden müssen.

Vorgehensweise

Führen Sie die folgenden Schritte aus, um die Verwaltung und Wartung zu terminieren:

1. Identifizieren Sie Zeiträume mit niedriger Datenbankaktivität.

   Verwaltungs- und Wartungsaktivitäten werden am besten für Zeiten mit geringer Beanspruchung der Datenbank terminiert (d. h. Zeiträume, in denen am wenigsten Benutzeranwendungen Anforderungen an das Datenbanksystem senden). Je nach Typ der von Ihnen erstellten Geschäftsanwendung kann es sogar Zeiträume geben, in denen überhaupt keine Benutzeranwendung auf das Datenbanksystem zugreift.
2. Kategorisieren Sie die Verwaltungs- und Wartungsaktivitäten nach folgenden Kriterien:
   - Verwaltung und Wartung können automatisiert werden
   - Die Datenbanklösung muss offline geschaltet werden, während Sie die Verwaltung/Wartung durchführen
   - Die Verwaltung/Wartung kann durchgeführt werden, während die Datenbank online geschaltet ist
3. Konfigurieren Sie für die Verwaltungsaktivitäten, die automatisiert werden können, die automatische Verwaltung mithilfe einer der folgenden drei Methoden:
   - Verwenden Sie den Konfigurationsparameter **auto_maint**.
   - Verwenden Sie den Assistenten für die Konfiguration der automatischen Verwaltung.
   - Verwenden Sie die gespeicherte Systemprozedur AUTOMAINT_SET_POLICY oder AUTOMAINT_SET_POLICYFILE
4. Wenn es für eine der auszuführenden Wartungs- oder Verwaltungsaktivitäten erforderlich ist, den Datenbankserver offline zu schalten, terminieren Sie diese Offlineverwaltungsaktivitäten für Zeiträume mit geringer Beanspruchung der Datenbank.
5. Bei Wartungs- und Verwaltungsaktivitäten, die ausgeführt werden können, während der Datenbankserver online geschaltet ist:
   - Ermitteln Sie, wie stark die Verfügbarkeit Ihrer Datenbank durch das Ausführen der Onlineverwaltungsaktivitäten beeinträchtigt würde.
   - Terminieren Sie diese Onlineverwaltungsaktivitäten so, dass die Beeinträchtigung der Verfügbarkeit Ihres Datenbanksystems durch das Ausführen der Verwaltungsaktivitäten minimiert wird.

Beispiel: Terminieren Sie Onlineverwaltungsaktivitäten für Zeiten mit geringer Datenbankbeanspruchung, und verwenden Sie Mechanismen zum Drosseln bzw. zum Ausgleichen des Umfangs an Systemressourcen, die von den Verwaltungsaktivitäten beansprucht werden.

Erfassen von Informationen zur Richtlinie für automatische Verwaltung mit SYSPROC.AUTOMAINT_GET_POLICY oder SYSPROC.AUTOMAINT_GET_POLICYFILE

Sie können die gespeicherten Systemprozeduren AUTOMAINT_GET_POLICY und AUTOMAINT_GET_POLICYFILE verwenden, um die für eine Datenbank konfigurierte automatische Verwaltungsrichtlinie abzurufen.

Führen Sie die folgenden Schritte aus, um die Richtlinie für automatische Verwaltung für eine Datenbank abzurufen:

1. Stellen Sie eine Verbindung zur Datenbank her
2. Rufen Sie AUTOMAINT_GET_POLICY oder AUTOMAINT_GET_POLICYFILE auf
   - Die für AUTOMAINT_GET_POLICY erforderlichen Parameter sind die Folgenden:
     a. Der Verwaltungstyp. Er gibt den Typ der automatischen Verwaltungsaktivität an, über die Informationen zurückgegeben werden sollen.
     b. Verweis auf ein großes Binärobjekt (BLOB). In diesem Objekt gibt die Prozedur die Informationen zur Richtlinie für automatische Verwaltung im XML-Format zurück.
   - Die für AUTOMAINT_GET_POLICYFILE erforderlichen Parameter sind die Folgenden:
     a. Der Verwaltungstyp. Er gibt den Typ der automatischen Verwaltungsaktivität an, über die Informationen zurückgegeben werden sollen.
     b. Der Name einer Datei. In diese Datei gibt die Prozedur die Informationen zur Richtlinie für automatische Verwaltung aus.

   Gültige Werte für den Verwaltungstyp sind:
   - AUTO_BACKUP - automatisches Backup
   - AUTO_REORG - automatische Reorganisation von Tabellen und Indizes
   - AUTO_RUNSTATS - automatische RUNSTATS-Operationen für Tabellen
   - MAINTENANCE_WINDOW - Verwaltungsfenster

Konfigurieren einer Richtlinie für automatische Verwaltung mit SYSPROC.AUTOMAINT_SET_POLICY oder SYSPROC.AUTOMAINT_SET_POLICYFILE

Sie können die gespeicherten Systemprozeduren AUTOMAINT_SET_POLICY und AUTOMAINT_SET_POLICYFILE verwenden, um Ihre automatische Verwaltungsrichtlinie für eine Datenbank zu konfigurieren.

Führen Sie die folgenden Schritte aus, um eine Richtlinie für automatische Verwaltung für eine Datenbank zu konfigurieren:

1. Stellen Sie eine Verbindung zur Datenbank her
2. Rufen Sie AUTOMAINT_SET_POLICY oder AUTOMAINT_SET_POLICYFILE auf

- Die für AUTOMAINT_SET_POLICY erforderlichen Parameter sind die Folgenden:
  a. Der Verwaltungstyp. Er gibt den Typ der zu konfigurierenden automatischen Verwaltungsaktivität an.
  b. Verweis auf ein großes Binärobjekt (BLOB). Dieses Objekt gibt die Richtlinie für automatische Verwaltung im XML-Format an.
- Die für AUTOMAINT_SET_POLICYFILE erforderlichen Parameter sind die Folgenden:
  a. Der Verwaltungstyp. Er gibt den Typ der zu konfigurierenden automatischen Verwaltungsaktivität an.
  b. Der Name einer XML-Datei. Diese Datei gibt die Richtlinie für automatische Verwaltung an.

Gültige Werte für den Verwaltungstyp sind:

- AUTO_BACKUP - automatisches Backup
- AUTO_REORG - automatische Reorganisation von Tabellen und Indizes
- AUTO_RUNSTATS - automatische RUNSTATS-Operationen für Tabellen
- MAINTENANCE_WINDOW - Verwaltungsfenster

XML-Richtlinienspezifikation für automatische Verwaltung für AUTOMAINT_SET_POLICY oder AUTOMAINT_SET_POLICYFILE - Beispiel

Unabhängig davon, ob Sie für die Spezifikation Ihrer Richtlinie für die automatische Verwaltung AUTOMAINT_SET_POLICY oder AUTOMAINT_SET_POLICYFILE verwenden, muss dies im XML-Format geschehen. Im Verzeichnis SQLLIB/samples/automaintcfg sind Beispieldateien vorhanden, die veranschaulichen, wie Sie Ihre Richtlinie für die automatische Verwaltung in XML angeben müssen.

Der zweite Parameter, den Sie an die gespeicherte Systemprozedur AUTOMAINT_SET_POLICY übergeben, ist ein XML enthaltendes großes Binärobjekt (BLOB), das Ihre gewünschte Richtlinie für die automatische Verwaltung angibt. Der zweite Parameter, den Sie an die gespeicherte Systemprozedur AUTOMAINT_SET_POLICYFILE übergeben, ist der Name einer XML-Datei, die Ihre gewünschte Richtlinie für die automatische Verwaltung angibt. Die XML-Elemente, die im großen Binärobjekt gültig sind, welches Sie an AUTOMAINT_SET_POLICY übergeben, sind dieselben Elemente, die in der XML-Datei gültig sind, die Sie an AUTOMAINT_SET_POLICYFILE übergeben.

Im Beispielverzeichnis SQLLIB/samples/automaintcfg befinden sich vier XML-Dateien mit Beispielspezifikationen für Ihre Richtlinie für die automatische Verwaltung:

DB2MaintenanceWindowPolicySample.xml

Beschreibt die Spezifikation eines Verwaltungsfensters für die Zeitangabe, innerhalb deren der Datenbankmanager die automatische Verwaltung terminieren soll.

DB2AutoBackupPolicySample.xml

Beschreibt, wie die Spezifikation für den Datenbankmanager zum Ausführen eines automatischen Backups aussehen soll.

DB2AutoReorgPolicySample.xml

Beschreibt, wie die Spezifikation für den Datenbankmanager zum Ausführen der automatischen Tabellen- und Indexreorganisation aussehen soll.

DB2DefaultAutoRunstatsPolicySample.xml

Beschreibt, wie die Spezifikation für den Datenbankmanager zum Ausführen der automatischen RUNSTATS-Operationen für Tabellen aussehen soll.

Sie können Ihre eigene XML-Richtlinienspezifikation für die automatische Verwaltung erstellen, indem Sie das XML-Format aus den Beispieldateien kopieren und die XML gemäß den Anforderungen Ihres Systems modifizieren.

Konfigurieren der Optionen zur Datenbankprotokollierung

Mit den Konfigurationsparameter für die Datenbankprotokollierung können Sie Optionen zur Datenprotokollierung für Ihre Datenbank angeben, wie zum Beispiel den Typ der zu verwendenden Protokollierung, die Größe der Protokolldateien und die Position, an der die Protokolldateien gespeichert werden sollen.

Zur Konfiguration von Optionen der Datenbankprotokollierung müssen Sie über die Berechtigung SYSADM, SYSCTRL oder SYSMAINT verfügen.

Sie können Optionen für die Datenbankprotokollierung über den Befehlszeilenprozessor (CLP) mit dem Befehl UPDATE DATABASE CONFIGURATION, über die grafische Benutzerschnittstelle (GUI) des Assistenten Datenbankprotokollierung konfigurieren in der Steuerzentrale oder durch Aufrufen der API db2CfgSet konfigurieren.

- Gehen Sie wie folgt vor, um Optionen für die Datenbankprotokollierung über den Befehlszeilenprozessor mit dem Befehl UPDATE DATABASE CONFIGURATION zu konfigurieren:
  1. Geben Sie an, ob Sie eine Umlaufprotokollierung oder eine Archivprotokollierung wünschen. Wenn Sie eine Umlaufprotokollierung wünschen, müssen die Datenbankkonfigurationsparameter LOGARCHMETH1 und LOGARCHMETH2 auf den Wert OFF gesetzt werden. Dies ist die Standardeinstellung. Wenn Sie eine Archivprotokollierung nutzen möchten, müssen Sie mindestens einen dieser Datenbankkonfigurationsparameter auf einen anderen Wert als OFF setzen. Wenn Sie zum Beispiel die Archivprotokollierung verwenden und die archivierten Protokolle auf Platte speichern wollen, führen Sie den folgenden Befehl aus:

     ```
     db2 update db configuration for mydb using logarchmeth1
           disk:/u/dbuser/archived_logs
     ```

     Die archivierten Protokolle werden in einem Verzeichnis mit dem Namen '/u/dbuser/archived_logs' gespeichert.
  2. Geben Sie für die anderen Konfigurationsparameter zur Datenbankprotokollierung Werte nach Bedarf an. Weitere Konfigurationsparameter für die Datenbankprotokollierung sind folgende:
     - ARCHRETRYDELAY
     - BLK_LOG_DSK_FUL
     - FAILARCHPATH
     - LOGARCHOPT1
     - LOGARCHOPT2
     - LOGBUFSZ
     - LOGFILSIZ
     - LOGPRIMARY
     - LOGRETAIN

- LOGSECOND
- MAX_LOG
- MIRRORLOGPATH
- NEWLOGPATH
- MINCOMMIT
- NUMARCHRETRY
- NUM_LOG_SPAN
- OVERFLOWLOGPATH
- USEREXIT

Weitere Informationen zu diesen Konfigurationsparameter für die Datenbankprotokollierung finden Sie in „Konfigurationsparameter für die Datenbankprotokollierung".

- Gehen Sie wie folgt vor, um den Assistenten Datenbankprotokollierung konfigurieren zu öffnen:
  1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie die Datenbank finden, für die Sie die Protokollierung konfigurieren wollen.
  2. Klicken Sie die Datenbank mit der rechten Maustaste an, und wählen Sie **Datenbankprotokollierung konfigurieren** im Kontextmenü aus. Der Assistent **Datenbankprotokollierung konfigurieren** wird geöffnet.
- Detaillierte Informationen finden Sie in der Onlinehilfefunktion der Steuerzentrale.

Konfigurationsparameter für die Datenbankprotokollierung

Die Datenbankprotokollierung ist eins der Schlüsselelemente aller Strategien für die Hochverfügbarkeit. Sie können Datenbankprotokolle verwenden, um Transaktionsinformationen aufzuzeichnen, primäre und sekundäre bzw. Bereitschaftsdatenbanken zu synchronisieren und eine sekundäre Datenbank aktualisierend wiederherzustellen, die die Funktion für eine fehlgeschlagene Primärdatenbank übernommen hat. Sie müssen eine Reihe von Datenbankkonfigurationsparametern festlegen, um diese Datenbankprotokollierungsaktivitäten nach Ihren Bedürfnissen zu konfigurieren.

Wiederholungsintervall für Protokollarchivierung (archretrydelay)
Gibt die Zeit in Sekunden an, die jeweils zwischen einem fehlgeschlagenen Versuch, Protokolldateien zu archivieren, und dem nächsten Versuch gewartet werden soll. Der Standardwert ist 20.

Bei voller Protokollplatte blockieren (blk_log_dsk_ful)
Dieser Konfigurationsparameter kann so festgelegt werden, dass keine Fehler aufgrund erschöpfter Festplattenkapazität generiert werden, wenn DB2 keine neue Protokolldatei im aktiven Protokollpfad erstellen kann. Stattdessen wird DB2 alle fünf Minuten versuchen, die Protokolldatei zu erstellen, bis diese Datei erfolgreich erstellt wurde. Nach jedem Versuch schreibt DB2 eine Nachricht in das Protokoll mit den Benachrichtigungen für die Systemverwaltung. Sie können nur durch Überwachen des Protokolls mit den Benachrichtigungen für die Systemverwaltung feststellen, ob Ihre Anwendung deshalb blockiert ist, weil für Protokolldateien kein Plattenplatz mehr vorhanden ist. Bis zur erfolgreichen Erstellung der Protokolldatei kann keine Benutzeranwendung, die eine Aktualisierung von Tabellendaten ausführen will, eine Transaktion festschreiben. Abfragen mit Lesezugriff sind hiervon möglicherweise nicht direkt betroffen. Wenn eine Abfrage jedoch auf Daten zugreifen will, die aufgrund einer Aktualisierungsanforderung ge-

sperrt sind, oder auf eine Datenseite, die von der aktualisierenden Anwendung im Pufferpool fixiert wurde, scheinen Abfragen mit Lesezugriff ebenfalls zu blockieren.

Wenn Sie *blk_log_dsk_ful* auf YES setzen, blockieren Anwendungen, sobald DB2 einen Fehler wegen einer vollen Protokollplatte feststellt. Sie können diesen Fehler anschließend beheben, und die Transaktion kann fortgesetzt werden. Sie können auf einem vollen Datenträger Speicher freigeben, indem Sie alte Protokolldateien auf ein anderes Dateisystem verschieben, oder indem Sie die Größe des Dateisystems erhöhen, sodass blockierte Anwendungen beendet werden können.

Wenn blk_log_dsk_ful auf NO gesetzt wurde, schlägt eine Transaktion fehl, die einen Fehler wegen einer vollen Protokollplatte empfängt, und wird zurückgesetzt. In einigen Fällen wird die Datenbank abnormal beendet, wenn eine Transaktion einen Fehler wegen einer vollen Protokollplatte verursacht.

Alternativpfad für Archivprotokolldatei (failarchpath)
Gibt einen Alternativpfad für die Archivprotokolldateien an, wenn die angegebene Protokollarchivierungsmethode fehlschlägt. Dieses Verzeichnis ist ein Bereich zur temporären Speicherung der Protokolldateien, bis die fehlgeschlagene Protokollarchivierungsmethode wieder zur Verfügung steht, woraufhin die Protokolldateien aus diesem Verzeichnis in das Verzeichnis für die Protokollarchivierungsmethode versetzt werden. Durch Versetzen der Protokolldateien in dieses temporäre Verzeichnis können Situationen vermieden werden, in denen der Speicherplatz im Protokollverzeichnis erschöpft ist. Für diesen Parameter muss ein vollständig qualifiziertes, vorhandenes Verzeichnis angegeben werden.

Erste Protokollarchivierungsmethode (logarchmeth1), zweite Protokollarchivierungsmethode (logarchmeth2)
Diese Parameter weisen den Datenbankmanager an, Protokolldateien in einem anderen Pfad als dem Pfad für aktive Protokolldateien zu speichern. Wenn beide Parameter angegeben werden, wird jede Protokolldatei doppelt gespeichert. Dies bedeutet, dass Sie zwei Kopien der Archivprotokolldateien in zwei verschiedenen Verzeichnissen erhalten.

Zu den gültigen Werten für diese Parameter gehören ein Datenträgertyp und, in einigen Fällen, ein Zielfeld. Gültige Werte:

OFF Gibt an, dass die Protokollarchivierungsmethode nicht verwendet werden soll. Wenn sowohl *logarchmeth1* als auch *logarchmeth2* auf OFF gesetzt ist, heißt dies, dass die Datenbank Umlaufprotokolle verwendet. Sie kann in diesem Fall nicht aktualisierend wiederhergestellt werden. Dies ist der Standardwert.

LOGRETAIN
Dieser Wert kann nur für *logarchmeth1* verwendet werden und ist äquivalent zur Einstellung RECOVERY für den Konfigurationsparameter *logretain*. Wenn Sie diesen Wert angeben, werden die Konfigurationsparameter *logretain* automatisch aktualisiert.

USEREXIT
Dieser Wert kann nur für *logarchmeth1* verwendet werden und ist äquivalent zur Einstellung ON für den Konfigurationsparameter *userexit*. Wenn Sie diesen Wert angeben, wird der Konfigurationsparameter *userexit* automatisch aktualisiert.

DISK Auf diesen Wert muss ein Doppelpunkt (:) folgen und darauf der

vollständig qualifizierte Name des Pfads, in dem die Protokolldateien archiviert werden sollen. Wenn Sie beispielsweise *logarchmeth1* auf `DISK:/u/dbuser/archived_logs` setzen, werden die archivierten Protokolldateien in ein Verzeichnis mit dem Namen `/u/dbuser/archived_logs` gestellt.

Anmerkung: Wenn Sie auf Band archivieren, können Sie zum Speichern und Abrufen der Protokolldateien das Dienstprogramm 'db2tapemgr' verwenden.

TSM Wenn dieser Wert ohne weitere Konfigurationsparameter angegeben wird, werden Protokolldateien unter Verwendung der Standardmanagementklasse auf dem lokalen TSM-Server archiviert. Folgen auf diesen Wert ein Doppelpunkt (:) und eine TSM-Managementklasse, werden die Protokolldateien unter Verwendung der angegebenen Managementklasse archiviert.

VENDOR
Gibt an, dass zur Archivierung der Protokolldateien eine Bibliothek eines anderen Lieferanten verwendet wird. Auf diesen Wert müssen ein Doppelpunkt (:) und der Name der Bibliothek folgen. Die in der Bibliothek zur Verfügung gestellten APIs müssen die Backup- und Restore-APIs für Produkte anderer Lieferanten verwenden.

Anmerkung:

1. Wenn *logarchmeth1* oder *logarchmeth2* auf einen anderen Wert als `OFF` gesetzt ist, ist die Datenbank für die aktualisierende Recovery konfiguriert.
2. Wenn Sie die Konfigurationsparameter *userexit* oder *logretain* aktualisieren, wird *logarchmeth1* automatisch aktualisiert und umgekehrt. Wenn Sie *userexit* oder *logretain* verwenden, muss *logarchmeth2* auf `OFF` gesetzt werden.

Optionen für logarchmeth1 (logarchopt1), Optionen für logarchmeth2 (logarchopt2) Gibt eine Zeichenfolge an, die an die API von Tivoli Storage Manager (TSM) oder an APIs anderer Lieferanten übergeben wird.

In TSM-Umgebungen werden diese Parameter verwendet, um dem Datenbankmanager das Abrufen von Protokollen zu ermöglichen, die auf einem anderen TSM-Knoten, von einem anderen TSM-Benutzer oder in TSM-Umgebungen mit Proxy-Knoten erstellt wurden. Die Zeichenfolge muss in einem der folgenden Formate angegeben werden:

- Abrufen von Protokollen, die auf einem anderen TSM-Knoten generiert wurden, wenn der TSM-Server nicht für die Unterstützung von Proxy-Knotenclients konfiguriert ist:

  `"-fromnode=`*`knotenname`*`"`

- Abrufen von Protokollen, die von einem anderen TSM-Benutzer generiert wurden, wenn der TSM-Server nicht für die Unterstützung von Proxy-Knotenclients konfiguriert ist:

  `"-fromowner=`*`eignername`*`"`

- Abrufen von Protokollen, die auf einem anderen TSM-Knoten und von einem anderen TSM-Benutzer generiert wurden, wenn der TSM-Server nicht für die Unterstützung von Proxy-Knotenclients konfiguriert ist:

  `"-fromnode=`*`knotenname`* `-fromowner=`*`eignername`*`"`

- Abrufen von Protokollen, die von einem beliebigen Benutzer generiert wurden, wenn der TSM-Server für die Unterstützung von Proxy-Knotenclients konfiguriert ist:

  ```
  "-asnodename=proxy-knoten"
  ```

  Dabei ist *knotenname* der Name des TSM-Knotens, auf dem die Protokolldateien ursprünglich archiviert wurden, *eignername* ist der Name des TSM-Benutzers, von dem die Protokolldateien ursprünglich archiviert wurden, und *proxy-knoten* ist der Name des gemeinsam genutzten TSM-Proxy-Knotens. Jedes Protokollarchivierungsoptionsfeld entspricht einer der Protokollarchivierungsmethoden: *logarchopt1* wird mit *logarchmeth1* verwendet und *logarchopt2* mit *logarchmeth2*.

Anmerkung: Die Optionen `-fromnode` und `-fromowner` sind nicht kompatibel mit der Option `-asnodename` und sie können nicht zusammen verwenden werden. Verwenden Sie die Option `-asnodename` für TSM-Konfigurationen mit Proxy-Knoten und die beiden anderen Optionen für andere Arten von TSM-Konfigurationen. Weitere Informationen hierzu finden Sie im Abschnitt „Konfigurieren eines Tivoli Storage Manager-Clients" auf Seite 361.

Protokollpuffer (logbufsz)
Mit diesem Parameter kann die Speichermenge angegeben werden, die als Puffer für Protokollsätze verwendet werden soll, bevor diese Protokollsätze auf die Festplatte geschrieben werden. Die Protokollsätze werden auf die Platte geschrieben, sobald eines der folgenden Ereignisse eintritt:

- Eine Transaktion wird festgeschrieben.
- Die Größe des Protokollpuffers reicht nicht mehr aus.
- Ein anderes internes Datenbankmanagerereignis tritt ein.

Die Erhöhung der Protokollpuffergröße führt zu einer effizienteren E/A-Aktivität in Bezug auf die Protokollierung, da die Protokollsätze nun in größeren Abständen und dabei in größeren Mengen auf Platte geschrieben werden. Die Recovery kann bei einem höheren Wert für die Protokollpuffergröße jedoch länger dauern.

Protokolldateigröße (logfilsiz)
Dieser Parameter gibt die Größe jeder konfigurierten Protokolldatei an, und zwar als Anzahl 4-KB-Seiten.

Es besteht eine logische Begrenzung von 512 GB für den konfigurierbaren Gesamtspeicherplatz für aktive Protokolldateien. Diese Begrenzung ergibt sich aus dem oberen Grenzwert für jede Protokolldatei, der bei 2 GB liegt, und der maximalen Gesamtanzahl der primären und sekundären Protokolldateien, die bei 256 liegt. Ab Version 9.5 Fixpack 3 wurde der Gesamtspeicherplatz für aktive Protokolldateien auf 1024 GB erweitert und die Obergrenze für eine einzelne Protokolldatei beträgt nun 4 GB.

Die Größe der Protokolldatei hat eine direkte Auswirkung auf die Leistung. Wenn von einem Protokoll zu einem nächsten gewechselt werden muss, ist dies mit einem Leistungsaufwand verbunden. Aus reiner Leistungsperspektive betrachtet wäre das Protokoll daher umso besser, je größer es wäre. Dieser Parameter gibt auch die Protokolldateigröße für die Archivierung an. In diesem Fall ist eine größere Protokolldatei nicht unbedingt leistungsfördernd, da sie das Fehlerrisiko erhöht bzw. bei der Protokollübertragung zu Verzögerungen führen kann. Für den Speicherbereich für aktive Protokolldateien könnte es sich eher empfehlen, eine größere Anzahl kleinerer Protokolldateien zu verwenden. Wenn beispielsweise 2

sehr große Protokolldateien verwendet werden und eine Transaktion nahe am Ende einer der Protokolldateien startet, ist nur noch die Hälfte des Speicherbereichs verfügbar.

Jedes Mal, wenn eine Datenbank inaktiviert wird (d. h. alle Verbindungen zur Datenbank werden beendet), wird die gerade verwendete Protokolldatei abgeschnitten. Wenn eine Datenbank häufig inaktiviert wird, sollte eine eher kleine Protokolldateigröße gewählt werden, da DB2 andernfalls unnötigerweise eine große Datei generieren würde, die ohnehin abgeschnitten wird. Sie können diesen Aufwand vermeiden, indem Sie den Befehl ACTIVATE DATABASE verwenden. Wenn Sie den Pufferpool vorbereiten, wirkt sich dies ebenfalls positiv auf die Leistung aus.

Angenommen, Sie haben eine Anwendung, die die Datenbank geöffnet hält, um die Verarbeitungszeit beim Öffnen von Datenbanken zu minimieren. In diesem Fall sollte der Wert für die Protokollgröße durch den Zeitraum bestimmt werden, der erforderlich ist, um Kopien von Offlinearchivprotokolldateien anzulegen.

Die Verlustminimierung von Protokolldaten ist ebenfalls ein wichtiger Aspekt beim Definieren der Protokollgröße. Von der Archivierung ist stets das gesamte Protokoll betroffen. Wenn Sie ein einzelnes, umfangreiches Protokoll verwenden, vergrößern Sie den Zeitraum zwischen den Archivierungen. Bei einem Defekt des Datenträgers, auf dem sich das Protokoll befindet, gehen wahrscheinlich einige Transaktionsinformationen verloren. Das Verringern der Protokollgröße erhöht zwar die Häufigkeit der Archivierungen, kann aber den Informationsverlust bei einem Datenträgerfehler verringern, da die kleineren Protokolle, die vor dem verloren gegangenen Protokoll erstellt wurden, weiterhin verwendet werden können.

Beibehalten von Protokollen (logretain)
Dieser Konfigurationsparameter wurde durch *logarchmeth1* ersetzt. Er wird aus Gründen der Kompatibilität mit früheren Versionen von DB2 weiterhin unterstützt.

Wenn logretain auf `RECOVERY` gesetzt wurde, werden Archivprotokolldateien im Verzeichnis des Datenbankprotokollpfads aufbewahrt, und die Datenbank wird als wiederherstellbar betrachtet, was bedeutet, dass die aktualisierende Recovery aktiviert ist.

Anmerkung: Die Standardeinstellung für den Datenbankkonfigurationsparameter *logretain* unterstützt keine aktualisierende Recovery. Wenn Sie die aktualisierende Recovery verwenden wollen, müssen Sie den Wert dieses Parameters ändern.

Maximales Protokoll pro Transaktion (max_log)
Dieser Parameter zeigt den Prozentsatz des primären Protokollspeicherbereichs an, der von einer Transaktion verbraucht werden kann. Der Wert ist ein Prozentsatz des Werts, der für den Konfigurationsparameter *logprimary* angegeben ist.

Wird der Wert auf `0` gesetzt, besteht keine Begrenzung des Prozentsatzes des gesamten primären Protokollbereichs, den eine Transaktion verbrauchen kann. Verletzt eine Anwendung die Konfiguration *max_log*, wird die Anwendung gezwungen, die Verbindung zur Datenbank zu beenden. Die Transaktion wird rückgängig gemacht und der Fehler SQL1224N wird zurückgegeben.

Sie können dieses Verhalten außer Kraft setzen, indem Sie die Registrierdatenbankvariable DB2_FORCE_APP_ON_MAX_LOG auf *FALSE* setzen. Dies

bewirkt, dass Transaktionen, die die Konfiguration *max_log* verletzen, fehlschlagen und den Fehler SQL0964N zurückgeben. Die Anwendung kann weiterhin von vorherigen Anweisungen in der UOW fertig gestellte Arbeit festschreiben oder die fertig gestellte Arbeit wird rückgängig gemacht, um die UOW zu widerrufen.

Dieser Parameter kann zusammen mit dem Konfigurationsparameter num_log_span nützlich sein, wenn ein unbegrenzter aktiver Protokollspeicherbereich aktiviert ist. Wenn eine Endlosprotokollierung aktiv ist (d. h. *logsecond* hat den Wert -1), werden Transaktionen nicht auf die Obergrenze der Anzahl von Protokolldateien (*logprimary* + *logsecond*) beschränkt. Wenn der Wert von logprimary erreicht wurde, beginnt DB2 mit der Archivierung der aktiven Protokolldateien, statt die Transaktion fehlschlagen zu lassen. Dies kann zu Problemen führen, wenn zum Beispiel eine Transaktion mit langer Ausführungszeit ohne Commit zurückgelassen wurde (vielleicht aufgrund eines Anwendungsfehlers). In einen solchen Fall wächst der Protokollspeicherbereich immer weiter an, was sich negativ auf die Leistung bei einer Recovery nach einem Systemabsturz auswirken könnte. Um dies zu vermeiden, können Sie Werte für einen der Konfigurationsparameter max_log oder num_log_span (oder beide) angeben.

Anmerkung: Die folgenden DB2-Befehle sind von der Begrenzung ausgenommen, die vom Konfigurationsparameter *max_log* festgelegt wird: ARCHIVE LOG, BACKUP DATABASE, LOAD, REORG, RESTORE DATABASE und ROLLFORWARD DATABASE.

Pfad für Protokollspiegelung (mirrorlogpath)
Zum Schutz der Protokolle im primären Protokollpfad vor Plattenausfällen oder versehentlichem Löschen können Sie angeben, dass alle Protokolle zusätzlich in einem zweiten Protokollpfad, dem Spiegelprotokollpfad, gespeichert werden. Ändern Sie dazu den Wert dieses Konfigurationsparameters, sodass er auf ein anderes Verzeichnis zeigt. Aktive Protokolldateien, die momentan im Spiegelprotokollpfad gespeichert werden, werden nicht an die neue Position versetzt, wenn die Datenbank für die aktualisierende Recovery konfiguriert ist.

Da Sie den Protokollpfad ändern können, befinden sich die für die aktualisierende Recovery erforderlichen Protokolle möglicherweise in verschiedenen Verzeichnissen. Sie können diesen Konfigurationsparameter während der aktualisierenden Recovery ändern, um Zugriff auf Protokolle an mehreren Speicherpositionen zu ermöglichen.

Sie müssen die Speicherposition der Protokolle verfolgen.

Die Änderungen werden erst dann angewendet, wenn sich die Datenbank wieder in einem konsistenten Status befindet. Der Datenbankstatus wird durch den Konfigurationsparameter *database_consistent* zurückgegeben.

Wenn Sie diesen Konfigurationsparameter inaktivieren wollen, setzen Sie seinen Wert auf DEFAULT.

Anmerkung:

1. Dieser Konfigurationsparameter wird nicht unterstützt, wenn es sich bei dem primären Protokollpfad um eine Roheinheit handelt.
2. Der für diesen Parameter angegebene Wert darf keine Roheinheit sein.

Neuer Protokollpfad (newlogpath)
Anfangs werden die Datenbankprotokolle im Unterverzeichnis SQLOGDIR des Datenbankverzeichnisses erstellt. Sie können die Position, an der die

aktiven Protokolle sowie die künftigen Archivprotokolldateien gespeichert werden, ändern, indem Sie den Wert für diesen Konfigurationsparameter so ändern, dass er entweder auf ein anderes Verzeichnis oder auf eine andere Einheit zeigt. Aktive Protokolldateien, die momentan im Verzeichnispfad der Datenbankprotokolle gespeichert werden, werden nicht an die neue Position versetzt, wenn die Datenbank für die aktualisierende Recovery konfiguriert ist.

Da Sie den Protokollpfad ändern können, befinden sich die für die aktualisierende Recovery erforderlichen Protokolle möglicherweise in verschiedenen Verzeichnissen bzw. auf verschiedenen Einheiten. Sie können diesen Konfigurationsparameter während der aktualisierenden Recovery ändern, um Zugriff auf Protokolle an mehreren Speicherpositionen zu ermöglichen.

Sie müssen die Speicherposition der Protokolle verfolgen.

Die Änderungen werden erst dann angewendet, wenn sich die Datenbank wieder in einem konsistenten Status befindet. Der Datenbankstatus wird durch den Konfigurationsparameter *database_consistent* zurückgegeben.

Anzahl der Gruppencommits (mincommit)
Dieser Parameter ermöglicht Ihnen, das Schreiben von Protokollsätzen auf Platte zu verzögern, bis eine minimale Anzahl von Commitoperationen ausgeführt worden ist. Diese Verzögerung kann dazu beitragen, dass der Systemaufwand für den Datenbankmanager im Zusammenhang mit dem Schreiben von Protokollsätzen verringert und infolgedessen der Durchsatz erhöht wird, wenn mehrere Anwendungen für eine Datenbank ausgeführt werden und viele Commits von den Anwendungen innerhalb kurzer Zeit angefordert werden.

Diese Gruppierung von Commits wird nur dann ausgeführt, wenn der Wert dieses Parameters größer als eins und die Anzahl der Anwendungen, die mit der Datenbank verbunden sind, größer als der Wert dieses Parameters ist. Wenn die Gruppierung von Commits durchgeführt wird, werden Commitanforderungen von Anwendungen solange zurückgehalten, bis entweder eine Sekunde vergangen oder die Anzahl der Commitanforderungen gleich dem Wert dieses Parameters ist.

Anzahl Wiederholungen der Archivierung nach Fehler (numarchretry)
Gibt an, wie oft versucht wird, die Protokolldateien mit der angegebenen Protokollarchivierungsmethode zu archivieren, bevor sie in dem vom Konfigurationsparameter *failarchpath* angegebenen Pfad archiviert werden. Dieser Parameter kann nur verwendet werden, wenn der Konfigurationsparameter *failarchpath* gesetzt ist. Der Standardwert ist 5.

Anzahl aktiver Protokolldateien für UOW (num_log_span)
Dieser Parameter zeigt die Anzahl aktiver Protokolldateien an, die eine aktive Transaktion umfassen kann. Wird der Wert auf 0 gesetzt, gibt es keine Begrenzung für die Anzahl der Protokolldateien, die von einer einzigen Transaktion umfasst werden können.

Verletzt eine Anwendung die Konfiguration *num_log_span*, wird die Anwendung gezwungen, die Verbindung zur Datenbank zu beenden, und der Fehler SQL1224N wird zurückgegeben.

Dieser Parameter kann zusammen mit dem Konfigurationsparameter max_log nützlich sein, wenn ein unbegrenzter aktiver Protokollspeicherbereich aktiviert ist. Wenn eine Endlosprotokollierung aktiv ist (d. h. *logsecond* hat den Wert -1), werden Transaktionen nicht auf die Obergrenze der Anzahl von Protokolldateien (*logprimary* + *logsecond*) beschränkt. Wenn der Wert

von logprimary erreicht wurde, beginnt DB2 mit der Archivierung der aktiven Protokolldateien, statt die Transaktion fehlschlagen zu lassen. Dies kann zu Problemen führen, wenn zum Beispiel eine Transaktion mit langer Ausführungszeit ohne Commit zurückgelassen wurde (vielleicht aufgrund eines Anwendungsfehlers). In einen solchen Fall wächst der Protokollspeicherbereich immer weiter an, was sich negativ auf die Leistung bei einer Recovery nach einem Systemabsturz auswirken könnte. Um dies zu vermeiden, können Sie Werte für einen der Konfigurationsparameter max_log oder num_log_span (oder beide) angeben.

Anmerkung: Die folgenden DB2-Befehle sind von der Begrenzung ausgenommen, die vom Konfigurationsparameter *num_log_span* festgelegt wird: ARCHIVE LOG, BACKUP DATABASE, LOAD, REORG, RESTORE DATABASE und ROLLFORWARD DATABASE.

Überlaufprotokollpfad (overflowlogpath)

Je nach Ihren Protokollierungsanforderungen kann dieser Parameter für verschiedene Funktionen verwendet werden. Sie können eine Position angeben, an der DB2 nach Protokolldateien suchen soll, die für eine aktualisierende Recovery benötigt werden. Dies weist Ähnlichkeiten mit der Option OVERFLOW LOG PATH des Befehls ROLLFORWARD auf, mit dem Unterschied, dass die Option OVERFLOW LOG PATH mit jedem abgesetzten Befehl ROLLFORWARD angegeben werden muss und Sie diesen Parameter nur einmal angeben müssen. Wenn sowohl der Befehl als auch der Parameter verwendet werden, überschreibt die Option OVERFLOW LOG PATH den Konfigurationsparameter *overflowlogpath* für diese bestimmte ROLLFORWARD-Operation.

Wenn *logsecond* auf -1 gesetzt ist, können Sie ein Verzeichnis angeben, in dem DB2 die aus dem Archiv abgerufenen aktiven Protokolldateien speichern kann. (Aktive Protokolldateien müssen für Rollback-Operationen abgerufen werden, wenn sie sich nicht mehr im Pfad für aktive Protokolldateien befinden).

Wenn für *overflowlogpath* kein Wert angegeben ist, ruft DB2 die Protokolldateien in den Pfad für aktive Protokolldateien ab. Wenn Sie für diesen Parameter einen Wert angeben, können Sie DB2 dadurch zusätzliche Ressourcen zum Speichern der abgerufenen Protokolldateien bereitstellen. Dies bietet den Vorteil, den Ein-/Ausgabeaufwand auf verschiedene Datenträger zu verteilen und mehr Protokolldateien im Pfad für aktive Protokolldateien speichern zu können.

Wenn Sie beispielsweise die Anwendungsprogrammierschnittstelle (API) db2ReadLog zur Replikation verwenden, können Sie mit dem Parameter *overflowlogpath* eine Position angeben, an der DB2 nach Protokolldateien suchen soll, die für diese API benötigt werden. Wenn die Protokolldatei nicht gefunden wird (weder im Pfad für aktive Protokolldateien noch im Überlaufprotokollpfad) und die Datenbank mit dem aktivierten Parameter *userexit* konfiguriert ist, ruft DB2 die Protokolldatei ab. Sie können mit diesem Parameter auch ein Verzeichnis angeben, in dem DB2 die abgerufenen Protokolldateien speichern soll. Dadurch wird der Ein-/Ausgabeaufwand für den Pfad für aktive Protokolldateien reduziert, und es können mehr Protokolldateien im Pfad für aktive Protokolldateien gespeichert werden.

Wenn Sie für den Pfad für aktive Protokolldateien eine Roheinheit konfiguriert haben und *logsecond* auf -1 setzen oder die Anwendungsprogrammierschnittstelle db2ReadLog verwenden wollen, muss *overflowlogpath* konfiguriert werden.

Geben Sie eine Zeichenfolge von bis zu 242 Byte an, um den Parameter *overflowlogpath* festzulegen. Die Zeichenfolge muss auf einen Pfadnamen zeigen, bei dem es sich um einen vollständig qualifizierten Pfadnamen handeln muss, nicht um einen relativen Pfad. Der Pfadname muss ein Verzeichnis, keine Roheinheit sein.

Anmerkung: In einer Umgebung mit partitionierten Datenbanken wird die Datenbankpartitionsnummer automatisch an den Pfad angehängt. Dies geschieht, um die Eindeutigkeit des Pfads in mehreren logischen Knotenkonfigurationen zu bewahren.

Anzahl primärer Protokolle (logprimary)
Dieser Parameter gibt die Anzahl der primären Protokolle der Größe *logfilsiz* an, die erstellt werden.

Ein primäres Protokoll belegt im leeren sowie im vollen Zustand denselben Plattenspeicherplatz. Wenn Sie also mehr Protokolle als erforderlich konfigurieren, wird unnötigerweise Plattenspeicherplatz belegt. Konfigurieren Sie dagegen zu wenig Protokolle, kann der Fall eintreten, dass Ihre Protokolle vollständig mit Daten gefüllt sind. Beim Auswählen der Anzahl der zu konfigurierenden Protokolle müssen Sie daher die für jedes Protokoll vorgesehene Größe einkalkulieren und berücksichtigen, ob Ihre Anwendung mit vollen Protokollen umgehen kann. Die Protokolldateien im Speicherbereich aktiver Protokolldateien können insgesamt maximal 256 GB belegen.

Wenn Sie für eine existierende Datenbank die aktualisierende Recovery aktivieren, müssen Sie die Anzahl der primären Protokolle auf den Wert der Summe der primären und sekundären Protokolle plus 1 ändern. Zusätzliche Informationen werden für Felder der Datentypen LONG VARCHAR und LOB in einer Datenbank aufgezeichnet, die für die aktualisierende Recovery aktiviert ist.

Anzahl sekundärer Protokolle (logsecond)
Dieser Parameter gibt die Anzahl der sekundären Protokolldateien an, die erstellt und bei Bedarf für die Recovery verwendet werden.

Wenn die primären Protokolldateien voll sind, werden die sekundären Protokolldateien in der durch *logfilsiz* Größe definierten Größe bei Bedarf einzeln zugeordnet, und zwar im Höchstfall so viele, wie von diesem Parameter angegeben wird. Wenn der Parameter auf -1 gesetzt ist, wird die Datenbank mit unbegrenztem Speicherbereich für aktive Protokolldateien konfiguriert. Es gibt keine Begrenzung für die Größe oder Anzahl der unvollständigen Transaktionen, die in der Datenbank ausgeführt werden. Eine aktive Endlosprotokollierung ist in Umgebungen nützlich, die umfangreiche Jobs mit größerem Protokollspeicherbedarf verarbeiten müssen, als Sie normalerweise für primäre Protokolle zuordnen würden.

Anmerkung:

1. Die Protokollarchivierung muss aktiviert sein, damit *logsecond* auf -1 gesetzt werden kann.
2. Wenn dieser Parameter auf -1 gesetzt ist, wird der Zeitaufwand für die Recovery nach einem Systemabsturz möglicherweise größer, da DB2 vielleicht archivierte Protokolldateien abrufen muss.

Benutzerexit (userexit)

Dieser Konfigurationsparameter wurde durch *logarchmeth1* ersetzt. Er wird aus Gründen der Kompatibilität mit früheren Versionen von DB2 weiterhin unterstützt.

Der Konfigurationsparameter *userexit* weist den Datenbankmanager an, ein Benutzerexitprogramm zum Archivieren und Abrufen von Protokollen aufzurufen. Die Protokolldateien werden an einer anderen Position als dem aktiven Protokollpfad gespeichert. Wenn *userexit* auf `ON` gesetzt ist, ist die aktualisierende Recovery aktiviert.

Die Datenübertragungsgeschwindigkeit der Einheit, die Sie zum Speichern von Offlinearchivprotokolldateien verwenden, und die Software, mit der Sie die Kopien anlegen, müssen mindestens der Durchschnittsgeschwindigkeit entsprechen, mit der der Datenbankmanager Daten in die Protokolle schreibt. Wenn die Übertragungsgeschwindigkeit für die neuen Protokolldaten, die generiert werden, nicht ausreicht, kann der Plattenspeicherplatz knapp werden. Dieser Fall kann eintreten, wenn die Protokollierungsaktivitäten über einen genügend langen Zeitraum hinweg fortgesetzt werden. Die Zeit bis zum Knappwerden des Plattenspeicherplatzes ist abhängig von der Größe des freien Plattenspeicherplatzes. Reicht der verfügbare Speicherplatz nicht aus, wird die Datenbankverarbeitung gestoppt.

Die Datenübertragungsgeschwindigkeit spielt insbesondere bei der Verwendung von Bändern oder optischen Datenträgern eine bedeutende Rolle. Einige Bandeinheiten benötigen stets dieselbe Zeit zum Kopieren einer Datei, unabhängig davon, wie groß die betreffende Datei ist. Sie müssen die Leistungsfähigkeit Ihrer Archivierungseinheit bestimmen.

Für Bandeinheiten gelten andere Faktoren. Die Häufigkeit der Archivierungsanforderungen ist entscheidend. Wenn z. B. eine Kopieroperation fünf Minuten in Anspruch nimmt, muss das Protokoll groß genug sein, um bei einer hohen Arbeitsauslastung die Protokolldaten von fünf Minuten aufnehmen zu können. Bedingt durch die Eigenschaften Ihrer Bandeinheit kann zudem die Anzahl der pro Tag ausführbaren Operationen begrenzt sein. Sie müssen diese Faktoren berücksichtigen, wenn Sie die Protokollgröße definieren.

Anmerkung:

1. Dieser Parameter muss auf `ON` gesetzt werden, um unbegrenzten Speicherbereich für aktive Protokolldateien zu aktivieren.
2. Die Standardeinstellung für den Datenbankkonfigurationsparameter *userexit* unterstützt keine aktualisierende Recovery und muss geändert werden, falls Sie ihn verwenden wollen.

Verringern der Protokollieraktivitäten mit dem Parameter NOT LOGGED INITIALLY

Wenn Ihre Anwendung Arbeitstabellen aus Originaltabellen erstellt und füllt und Sie in Bezug auf die Wiederherstellbarkeit dieser Arbeitstabellen keine Bedenken haben, da sie unproblematisch aus den Originaltabellen wiederhergestellt werden können, haben Sie die Möglichkeit, die Arbeitstabellen unter Angabe des Parameters NOT LOGGED INITIALLY in der Anweisung CREATE TABLE zu erstellen. Dadurch wird die Protokollierung reduziert und die Leistung verbessert.

Ein Vorteil in der Verwendung des Parameters NOT LOGGED INITIALLY besteht darin, dass etwaige in der Tabelle durchgeführte Änderungen (wie Einfüge-, Lösch-, Aktualisierungs- oder Indexerstellungsoperationen) innerhalb derselben

UOW (Unit of Work), in der die Tabelle erstellt wird, nicht protokolliert werden. Dadurch werden nicht nur die Protokollieraktivitäten verringert, sondern es wird auch eine bessere Leistung für Ihre Anwendung erzielt. Sie können das gleiche Ergebnis für vorhandene Tabellen erreichen, indem Sie die Anweisung ALTER TABLE mit dem Parameter NOT LOGGED INITIALLY verwenden.

Anmerkung:

1. Sie können mehrere Tabellen mit dem Parameter NOT LOGGED INITIALLY in derselben UOW erstellen.
2. Änderungen an den Katalogtabellen und anderen Benutzertabellen werden weiterhin protokolliert.

Da die Änderungen der Tabelle nicht protokolliert werden, sollten Sie bei der Entscheidung, das Tabellenattribut NOT LOGGED INITIALLY zu verwenden, Folgendes beachten:

- *Alle* Änderungen an der Tabelle werden zum Commitzeitpunkt auf Platte geschrieben. Das bedeutet, dass die Commitoperation länger dauern kann.
- Wenn das Attribut NOT LOGGED INITIALLY aktiviert ist und eine nicht protokollierte Aktivität ausgeführt wird, wird beim Fehlschlagen einer Anweisung entweder die gesamte UOW rückgängig gemacht oder eine Operation ROLLBACK TO SAVEPOINT ausgeführt (SQL1476N).
- Wenn Sie HADR (High Availability Disaster Recovery) verwenden, sollten Sie das Tabellenattribut NOT LOGGED INITIALLY nicht verwenden. Tabellen, die mit der Option NOT LOGGED INITIALLY erstellt werden, werden nicht repliziert. Versuche, auf solche Tabellen zuzugreifen, nachdem eine HADR-Bereitschaftsdatenbank die Funktion als primäre Datenbank übernommen hat, enden mit einem Fehler.
- Diese Tabellen können Sie bei einer aktualisierenden Recovery nicht wiederherstellen. Wenn die aktualisierende Recovery auf eine Tabelle stößt, die mit der Option NOT LOGGED INITIALLY erstellt wurde, wird diese Tabelle als nicht verfügbar markiert. Nach der Recovery der Datenbank führt jeder Versuch, auf die Tabelle zuzugreifen, zur Rückgabe der Nachricht SQL1477N.

  Anmerkung: Bei der Erstellung einer Tabelle werden Zeilensperren für die Katalogtabellen aktiviert, bis ein Commit durchgeführt wird. Die Inaktivierung der Protokollfunktion ist nur dann von Vorteil, wenn Sie die Tabelle in derselben UOW mit Werten füllen, in der sie erstellt wird. Daraus ergeben sich Konsequenzen für den gemeinsamen Zugriff.

Verringern der Protokollieraktivitäten mit deklarierten temporären Tabellen

Wenn Sie planen, deklarierte temporäre Tabellen als Arbeitstabellen einzusetzen, ist Folgendes zu beachten:

- Deklarierte temporäre Tabellen werden nicht in den Katalogen erstellt. Daher werden keine Sperren aktiviert.
- Für deklarierte temporäre Tabellen findet keine Protokollierung statt, auch nicht nach der ersten Commitoperation.
- Geben Sie die Option ON COMMIT PRESERVE an, um die Zeilen in der Tabelle nach einer Commitoperation zu behalten. Ansonsten werden alle Zeilen gelöscht.
- Nur die Anwendung, die die deklarierte temporäre Tabelle erstellt, kann auf diese Instanz der Tabelle zugreifen.

- Die Tabelle wird implizit gelöscht, wenn die Verbindung der Anwendung zur Datenbank beendet wird.
- Betriebsfehler in einer UOW mit einer deklarierten temporären Tabelle bewirken nicht, dass die UOW vollständig rückgängig gemacht wird. Ein Fehler bei der Ausführung einer Anweisung, die den Inhalt einer deklarierten temporären Tabelle ändert, führt jedoch dazu, dass alle Zeilen in dieser Tabelle gelöscht werden. Eine Rollback-Operation der UOW (bzw. eines Sicherungspunkts) löscht alle Zeilen in deklarierten temporären Tabellen, die innerhalb dieser UOW (bzw. dieses Sicherungspunkts) geändert wurden.

Blockieren von Transaktionen bei vollem Protokollverzeichnis

Wenn der DB2-Datenbankmanager im Pfad für aktive Protokolldateien keine neue Protokolldatei erstellen kann, weil für diese nicht genügend Speicherplatz verfügbar ist, erhalten Sie Fehlerrückmeldungen, dass der Datenträger voll ist. Wenn Sie dagegen den Datenbankkonfigurationsparameter blk_log_dsk_ful einrichten, unternimmt der DB2-Datenbankmanager wiederholte Versuche, die neue Protokolldatei zu erstellen, bis er schließlich erfolgreich ist, anstatt den Fehler „Datenträger voll" zurückzugeben.

Wenn Sie den Datenbankkonfigurationsparameter blk_log_dsk_ful einrichten, versucht der DB2-Datenbankmanager alle fünf Minuten, die Protokolldatei zu erstellen, bis er erfolgreich ist. Wenn eine Protokollarchivierungsmethode angegeben wurde, prüft der DB2-Datenbankmanager außerdem den Abschluss der Protokolldateiarchivierung. Wenn eine archivierte Protokolldatei erfolgreich archiviert wurde, kann der DB2-Datenbankmanager die inaktive Protokolldatei in den neuen Protokolldateinamen umbenennen und fortfahren. Nach jedem Versuch schreibt der DB2-Datenbankmanager eine Nachricht in das Protokoll mit den Benachrichtigungen für die Systemverwaltung. Sie können nur durch Überwachen des Protokolls mit den Benachrichtigungen für die Systemverwaltung feststellen, ob Ihre Anwendung nur deshalb blockiert ist, weil für Protokolldateien kein Plattenplatz mehr vorhanden ist.

Bis zur erfolgreichen Erstellung der Protokolldatei kann keine Benutzeranwendung, die eine Aktualisierung von Tabellendaten ausführen will, eine Transaktion festschreiben. Auf Lesezugriff beschränkte Abfragen sind u. U. nicht direkt betroffen. Wenn jedoch eine Abfrage auf Daten zugreifen muss, die aufgrund einer Aktualisierungsanforderung gesperrt sind, oder auf eine Datenseite, die im Pufferpool von der aktualisierenden Anwendung korrigiert wird, erscheinen auch auf Lesezugriff beschränkte Abfragen blockiert.

Protokolldateiverwaltung durch Protokollarchivierung

Die Protokolldateiarchivierung von DB2 Data Server gestaltet sich durch verschiedene Dateiverwaltungs- und Terminierungsprobleme der Betriebssysteme als kompliziert. Beispiel: der DB2-Datenbankmanager könnte unter bestimmten Bedingungen versuchen, eine Protokolldatei abzurufen, während diese Datei gerade archiviert wird. Oder es könnten im Falle eines Datenträgerfehlers, während der DB2-Datenbankmanager eine Warteschlange von Protokolldateien archiviert, einige der Protokolldateien (und der in ihnen enthaltenen Transaktionsdaten) verloren gehen. Ein ordnungsgemäßes Konfigurieren der Datenbankprotokollierung kann verhindern, dass derartige potenzielle Probleme Ihre Verfügbarkeits- und Recoverystrategie untergraben.

Im Folgenden werden allgemeine Aspekte erläutert, die für alle Methoden der Protokollarchivierung gelten:

- Wenn Sie einen Wert für den Datenbankkonfigurationsparameter *logarchmeth1* festlegen, geben Sie dadurch an, dass der Datenbankmanager Dateien archivieren bzw. Protokolldateien bei der aktualisierenden Recovery von Datenbanken mithilfe der angegebenen Methode abrufen soll. Es erfolgt eine Anforderung zum Abrufen einer Protokolldatei, wenn das Dienstprogramm zur aktualisierenden Recovery eine Protokolldatei benötigt, die nicht im Verzeichnis für den Protokollpfad zu finden ist.
- Lokal angeschlossene Bandlaufwerke sollten nicht zum Speichern von Protokolldateien verwendet werden, wenn Sie eines der folgenden Verfahren verwenden:
  - Endlosprotokollierung
  - Online-Recovery auf Tabellenbereichsebene
  - Replikation
  - Die API zum asynchronen Lesen von Protokolldaten (db2ReadLog)
  - HADR (High Availability Disaster Recovery)

  Bei jedem dieser Verfahren kann eine Protokolldatei abgerufen werden, was mit den Protokollarchivierungsoperationen kollidieren kann.
- Wenn Sie mit Protokollarchivierung arbeiten, versucht der Protokollmanager aktive Protokolle zu archivieren, wenn sie gefüllt sind. Wenn eine Datenbank inaktiviert wird, bevor der Protokollmanager das Archiv erfolgreich aufzeichnen kann, versucht der Protokollmanager unter Umständen, das Protokoll erneut zu archivieren, wenn die Datenbank aktiviert wird. Auf diese Weise kann es vorkommen, dass eine Protokolldatei mehrmals archiviert wird.
- Beim Archivieren wird eine Protokolldatei an den Protokollmanager übergeben, wenn sie voll ist, auch wenn die Protokolldatei noch aktiv ist und für die normale Verarbeitung benötigt wird. Dadurch können Kopien der Daten so schnell wie möglich aus den flüchtigen Speichern entfernt werden. Die an den Protokollmanager übergebene Protokolldatei wird so lange im Verzeichnis für den Protokollpfad behalten, bis sie für die normale Verarbeitung nicht mehr benötigt wird. Dann wird der Plattenspeicherplatz wiederverwendet.
- Wenn eine Protokolldatei archiviert wurde und keine offenen Transaktion enthält, löscht der DB2-Datenbankmanager die Datei nicht, sondern benennt sie in die nächste Protokolldatei um, falls eine solche benötigt wird. Dies führt zu einem Leistungsvorteil, da bei der Erstellung einer neuen Protokolldatei (statt der Umbenennung) alle Seiten neu geschrieben werden müssen, um den Plattenspeicherplatz zu reservieren. Es ist effizienter, die erforderlichen Seiten auf der Platte wiederzuverwenden, als sie freizugeben und neu zuzuordnen.
- Der DB2-Datenbankmanager ruft während einer Recovery nach einem Systemabsturz oder eines Rollbacks *keine* Protokolldateien ab, es sei denn, der Datenbankkonfigurationsparameter *logsecond* ist auf -1 gesetzt.
- Das Konfigurieren der Protokollarchivierung garantiert nicht die aktualisierende Recovery bis zum Fehlerpunkt, sondern ist lediglich ein Versuch, das Fehlerfenster zu verkleinern. Wenn Protokolldateien voll sind, werden sie vom Protokollmanager asynchron archiviert. Sollte bei der Platte, auf der das Protokoll gespeichert wird, ein Fehler auftreten, bevor eine Protokolldatei voll ist, gehen die Daten in der betreffenden Protokolldatei verloren. Da die Dateien für die Archivierung in eine Warteschlange gestellt werden, kann außerdem ein Fehler auf der Platte auftreten, bevor alle Dateien kopiert werden.

  Im Fall eines Ausfalls der Platte oder Einheit, auf der sich der Protokollpfad befindet, können Sie mithilfe des Datenbankkonfigurationsparameters MIRRORLOGPATH sicherstellen, dass Ihre Protokolle in den sekundären Pfad geschrieben werden, sofern die Platte bzw. die Einheit, auf der sich der Spiegelprotokollpfad befindet, nicht ebenfalls ausgefallen ist.

- Die konfigurierte Größe der einzelnen Protokolldateien hat eine direkte Auswirkung auf die Protokollarchivierung. Wenn die einzelnen Protokolldateien sehr groß sind, kann eine große Menge von Daten verlorengehen, wenn ein Fehler auf der Platte auftritt. Wenn Sie Ihre Datenbank zur Verwendung kleiner Protokolldateien konfigurieren, führt der Protokollmanager häufig Protokollarchivierungen aus.

  Wenn die Daten allerdings auf eine langsamere Einheit wie ein Band übertragen werden, ist es oft sinnvoller, größere Protokolldateien zu konfigurieren, um eine zu große Warteschlange zu vermeiden. Sie sollten ebenfalls größere Protokolldateien verwenden, wenn durch das Archivieren der einzelnen Dateien jeweils zusätzlicher Systemaufwand entsteht, z. B. Zurückspulen der Bandeinheit oder Herstellen einer Verbindung zum Archivierungsmedium.
- Wenn Sie mit Protokollarchivierung arbeiten, versucht der Protokollmanager aktive Protokolle zu archivieren, wenn sie gefüllt sind. In einigen Fällen archiviert der Protokollmanager ein Protokoll, bevor es voll ist. Dies geschieht, wenn die Protokolldatei entweder durch die Inaktivierung der Datenbank, durch die Ausführung des Befehls ARCHIVE LOG am Ende eines Online-Backups oder durch die Ausführung des Befehls SET WRITE SUSPEND abgeschnitten wird.

  Anmerkung: Zur Freigabe nicht genutzten Protokollspeichers wird die Protokolldatei abgeschnitten, bevor sie archiviert wird.
- Wenn Sie Protokolle und Backup-Images auf einem Bandlaufwerk als Speichereinheit für Protokolle und Backup-Images archivieren, müssen Sie sicherstellen, dass das als Ziel für die Backup-Images und die archivierten Protokolldateien nicht dasselbe Bandlaufwerk verwendet wird. Da während einer Backup-Operation eine Protokollarchivierung stattfinden kann, kann es zu einem Fehler kommen, wenn beide Prozesse gleichzeitig versuchen, auf dasselbe Bandlaufwerk zu schreiben.

Folgende Hinweise und Informationen sind für das Aufrufen eines Benutzerexitprogramms bzw. eines von einem anderen Anbieter gelieferten Programms zum Archivieren und Abrufen von Protokolldateien zu beachten:

- Der DB2-Datenbankmanager öffnet eine Protokolldatei im Lesemodus, wenn er ein Benutzerexitprogramm zum Archivieren der Datei startet. Auf einigen Plattformen nimmt dies dem Benutzerexitprogramm die Möglichkeit, die Protokolldatei zu löschen. Andere Plattformen wie AIX ermöglichen es Prozessen, einschließlich des Benutzerexitprogramms, Protokolldateien zu löschen. Ein Benutzerexitprogramm sollte eine Protokolldatei nie nach ihrer Archivierung löschen, da die Datei weiterhin aktiv und für die Recovery nach einem Systemabsturz erforderlich sein könnte. Der DB2-Datenbankmanager verwaltet die Wiederverwendung des Speicherplatzes auf Datenträgern, wenn Protokolldateien archiviert werden.
- Wenn ein Benutzerexitprogramm bzw. ein Programm eines anderen Anbieters eine Anforderung zum Archivieren einer nicht existierenden Datei empfängt (weil mehrere Anforderungen zum Archivieren ausgeführt wurden und die Datei nach der ersten erfolgreichen Archivierung gelöscht wurde) oder zum Abrufen einer nicht existierenden Datei (weil sie sich in einem anderen Verzeichnis befindet oder das Ende der Protokolle erreicht wurde), sollte es diese Anforderung ignorieren und einen erfolgreichen Rückkehrcode übergeben.
- Unter Windows-Betriebssystemen können Sie keinen REXX-Benutzerexit zum Archivieren von Protokollen verwenden.

- Das Benutzerexitprogramm bzw. das Programm eines anderen Anbieters sollte das Vorhandensein verschiedener gleichnamiger Protokolldateien nach einer Recovery bis zu einem bestimmten Zeitpunkt zulassen. Es sollte so konzipiert sein, dass beide Protokolldateien beibehalten werden und diese Protokolldateien dem korrekten Recoverypfad zugeordnet werden.
- Wenn ein Benutzerexitprogramm bzw. ein Programm eines anderen Anbieters für zwei oder mehr Datenbanken eingerichtet ist, die zur Archivierung von Protokolldateien dieselbe Bandeinheit verwenden, und für eine der Datenbanken eine aktualisierende Recovery ausgeführt wird, sollten keine anderen Datenbanken aktiv sein. Wenn eine der anderen Datenbanken während der Operation zur aktualisierenden Recovery versucht, eine Protokolldatei zu archivieren, ist es möglich, dass die zur aktualisierenden Recovery erforderlichen Protokolle nicht gefunden werden oder dass die neue, auf der Bandeinheit archivierte Protokolldatei die zuvor auf dieser Bandeinheit gespeicherten Protokolldateien überschreibt.

  Damit keine der beiden Situationen auftritt, müssen Sie entweder sicherstellen, dass keine anderen Datenbanken in der Datenbankpartition, die das Benutzerexitprogramm aufruft, während der aktualisierenden Recovery geöffnet sind, oder Sie müssen ein Benutzerexitprogramm zur Behandlung dieser Situation schreiben.

Konfigurieren der Datenbankprotokollierung ohne Dateisystemcaching

Für die Verwaltung von Datenbankrecoveryprotokollen in Dateisystemen unter AIX, Linux, Solaris, HP-UX und Windows können Sie ungepufferte E/A (auch: direkte E/A, Direct I/O oder DIO) verwenden. Dadurch entsteht für das Betriebssystem kein Systemaufwand für das Caching von Datenbankrecoveryprotokollen.

Vorgehensweise

Aktivieren Sie die direkte E/A in Ihrem Dateisystem, indem Sie die Registrierdatenbankvariable **DB2_LOGGER_NON_BUFFERED_IO** auf ON setzen. Es wird empfohlen, das Dateisystem nicht mit Optionen anzuhängen, durch die die betriebssystembasierte Cachepufferung für das Dateisystem inaktiviert wird.

Ergebnisse

Mit dieser Konfiguration wird vermieden, dass im Zuge des Cachings von Datenbankrecoveryprotokollen Systemaufwand für das Betriebssystem entsteht. In bestimmten Situationen kann das Fehlen von betriebssystembasiertem Dateicaching zu einer Leistungsbeeinträchtigung bei den Platten-E/A-Anwortzeiten im Zusammenhang mit Protokollen führen, was längere Commitzeiten zur Folge hat. Darüber hinaus kann dies die Leistung bei Protokollarchivierungs- oder -rollbackoperationen beeinflussen. Den Leistungseinbußen durch längere Commitzeiten kann dadurch entgegengewirkt werden, dass sichergestellt wird, dass die Anzahl der physischen Plattenspindeln für Protokolldateisysteme dem gewünschten Leistungsniveau entspricht. Darüber hinaus wird empfohlen, Speichercontrollermechanismen für das Caching von Schreiboperationen zu aktivieren, um die Leistung zu verbessern, vorausgesetzt, diese Caching-Mechanismen bietet umfassende Stabilität. Die Leistungsprobleme beim Rollback können Sie beheben, indem Sie den Datenbankkonfigurationsparameter **logbufsz** so optimieren, dass sichergestellt wird, dass die für die aktualisierende Recovery erforderlichen Protokolldaten im Protokollpuffer selbst bereitgestellt werden, sodass keine physische Lese-E/A auf der Platte durchgeführt werden muss.

Konfigurieren einer Clusterumgebung für hohe Verfügbarkeit

Eine Strategie zum Entwerfen einer hoch verfügbaren Lösung besteht im Erstellen eines Clusters von Maschinen und in der Verwendung von Cluster-Management-Software, um die Auslastung auf diesen Maschinen gleichmäßig zu verteilen. Wenn Sie IBM Data Server auf einer oder auf mehreren Maschinen eines Clusters installieren, müssen Sie den Cluster-Manager so konfigurieren, dass er ordnungsgemäß auf Fehler oder Störungen reagiert, die die Datenbank(en) beeinträchtigen. Sie müssen außerdem die Datenbankmanagerinstanzen so konfigurieren, dass sie in einer Clusterumgebung ordnungsgemäß funktionieren.

Informationen zu dieser Task

Die manuelle Konfiguration und Verwaltung der Datenbankinstanzen und des Cluster-Managers ist komplex, zeitintensiv und fehleranfällig. DB2 High Availability Feature stellt eine Infrastruktur bereit, die es dem Datenbankmanager ermöglicht, mit dem Cluster-Manager zu kommunizieren, wenn Konfigurationsänderungen bei Instanzen, wie beispielsweise das Stoppen einer Datenbankmanagerinstanz, Clusteränderungen erforderlich machen.

Vorgehensweise

1. Installieren Sie die Cluster-Management-Software.

   IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist in IBM Data Server unter AIX- und Linux-Betriebssystemen als Teil von DB2 High Availability Feature integriert. Verwenden Sie für eine Installation, ein Upgrade oder eine Deinstallation von SA MP Base Component entweder das DB2-Installationsprogramm oder die Scripts installSAM und uninstallSAM, die sich auf dem Installationsdatenträger von IBM Data Server befinden. Unter Windows-Betriebssystemen ist SA MP Base Component Teil des Produktpakets von DB2 High Availability Feature, es ist jedoch nicht in das DB2-Installationsprogramm integriert.
2. Konfigurieren Sie IBM Data Server-Datenbankmanagerinstanzen für den Cluster-Manager, und konfigurieren Sie den Cluster-Manager für IBM Data Server.

   DB2 High Availability Instance Configuration Utility (db2haicu) ist ein textbasiertes Dienstprogramm zur Konfiguration und Verwaltung hoch verfügbarer Datenbanken in einer Clusterumgebung. db2haicu stellt Abfragen an das System, um Informationen zur Datenbankinstanz, zur Clusterumgebung und zum Cluster-Manager zu sammeln. Zur Angabe weiterer Informationen über Parameter für den Aufruf db2haicu, eine Eingabedatei oder während der Laufzeit können Sie die Eingabeaufforderungen von db2haicu verwenden.
3. Wenn sich die Anforderungen an die Datenbank im Laufe der Zeit ändern und die Datenbankkonfiguration in der Clusterumgebung modifiziert werden muss, fahren Sie damit fort, die Konfiguration der Datenbankmanagerinstanzen und die Konfiguration des Cluster-Managers zu synchronisieren.

   DB2 High Availability Feature stellt eine Infrastruktur bereit, die es dem Datenbankmanager ermöglicht, mit dem Cluster-Manager zu kommunizieren, wenn Konfigurationsänderungen bei Instanzen, wie beispielsweise das Stoppen einer Datenbankmanagerinstanz, Clusteränderungen erforderlich machen.

   Es spielt keine Rolle, ob Sie db2haicu mit SA MP Base Component oder einen weiteren Cluster-Manager verwenden, der die API des DB2-Cluster-Managers unterstützt - die Verwaltung der Clusterumgebung mit DB2 High Availability Feature ist einfacher als eine separate Verwaltung von Datenbankmanager- und Clusterkonfiguration.

Cluster-Manager-Integration mit DB2 High Availability Feature

DB2 High Availability Feature ermöglicht die Integration zwischen IBM Data Server und Cluster-Management-Software.

Wenn Sie eine Datenbankmanagerinstanz in einer Clusterumgebung stoppen, müssen Sie den Cluster-Manager darüber informieren, dass die Instanz gestoppt wurde. Wenn der Cluster-Manager über das Stoppen der Instanz nicht informiert wird, versucht er möglicherweise, eine Operation wie beispielsweise eine Funktionsübernahme für die gestoppte Instanz auszuführen. DB2 High Availability Feature stellt eine Infrastruktur bereit, die es dem Datenbankmanager ermöglicht, mit dem Cluster-Manager zu kommunizieren, wenn Konfigurationsänderungen bei Instanzen, wie beispielsweise das Stoppen einer Datenbankmanagerinstanz, Clusteränderungen erforderlich machen.

DB2 High Availability Feature besteht aus den folgenden Elementen:

- IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist in IBM Data Server unter AIX- und Linux-Betriebssystemen als Teil von DB2 High Availability Feature integriert. Verwenden Sie für eine Installation, ein Upgrade oder eine Deinstallation von SA MP Base Component entweder das DB2-Installationsprogramm oder die Scripts installSAM und uninstallSAM, die sich auf dem Installationsdatenträger von IBM Data Server befinden. Unter Windows-Betriebssystemen ist SA MP Base Component Teil des Produktpakets von DB2 High Availability Feature, es ist jedoch nicht in das DB2-Installationsprogramm integriert. Siehe: „Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm"
- In einer Clusterumgebung erfordern bestimmte Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz entsprechende Änderungen an der Clusterkonfiguration. Mit DB2 High Availability Feature kann der Datenbankmanager automatisch Änderungen an der Konfiguration des Cluster-Managers anfordern, wenn bestimmte Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz ausgeführt werden. Siehe: „Automatisches Konfigurieren eines Clusters mit DB2 High Availability Feature" auf Seite 96
- DB2 High Availability Instance Configuration Utility (db2haicu) ist ein textbasiertes Dienstprogramm zur Konfiguration und Verwaltung hoch verfügbarer Datenbanken in einer Clusterumgebung. db2haicu stellt Abfragen an das System, um Informationen zur Datenbankinstanz, zur Clusterumgebung und zum Cluster-Manager zu sammeln. Zur Angabe weiterer Informationen über Parameter für den Aufruf db2haicu, eine Eingabedatei oder während der Laufzeit können Sie die Eingabeaufforderungen von db2haicu verwenden. Siehe: „db2haicu" auf Seite 104
- Die DB2-Cluster-Manager-API definiert eine Reihe von Funktionen, mit deren Hilfe der Datenbankmanager dem Cluster-Manager Konfigurationsänderungen mitteilen kann. Siehe: „DB2-Cluster-Manager-API" auf Seite 140

Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm

IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist in IBM Data Server unter AIX- und Linux-Betriebssystemen als Teil von DB2 High Availability Feature integriert. Verwenden Sie für eine Installation, ein Upgrade oder eine Deinstallation von SA MP Base Component entweder das DB2-Installationsprogramm oder die Scripts installSAM und uninstallSAM, die sich auf dem Installationsdatenträger von IBM Data Server befinden. Unter Windows-Betriebssystemen ist SA MP Base Component Teil des Produktpakets von DB2 High Availability Feature, es ist jedoch nicht in das DB2-Installationsprogramm integriert.

Vorbereitung

- Zur Installation und Verwendung von SA MP Base Component muss Ihre Systemkonfiguration und die vorgesehene Nutzung von SA MP Base Component mit den Lizenzbedingungen für das in IBM Data Server integrierte Produkt SA MP Base Component übereinstimmen.

  Ausführliche Informationen zu den Lizenzbedingungen für das in IBM Data Server integrierte Produkt SA MP Base Component finden Sie in „Lizenzbedingungen für die Verwendung des in IBM Data Server integrierten Produkts IBM Tivoli SA MP Base Component" auf Seite 94.
- Zur Installation oder für ein Upgrade von SA MP Base Component muss Ihre Systemarchitektur von SA MP Base Component unterstützt werden, das in IBM Data Server integriert ist.

  Weitere Informationen zur Software und Hardware, die von SA MP Base Component unterstützt wird, finden Sie in „Unterstützte Software und Hardware für IBM Tivoli SA MP Base Component" auf Seite 94.
- Zur Installation von SA MP Base Component ist eine Rootberechtigung erforderlich.

  Bei einer Installation, die nicht als Root ausgeführt wird, können Sie SA MP Base Component separat vom Installationsdatenträger von IBM Data Server installieren. Wenn Sie SA MP Base Component separat installieren, benötigen Sie ebenfalls eine Rootberechtigung.
- Version 9.5 Fixpack 6 und spätere Fixpacks enthalten eine aktualisierte Version von SA MP Base Component, eine höhere Version als die Version, die im ursprünglichen Lieferumfang von DB2 Version 9.5 enthalten war. Für einige Umgebungen mit neueren Betriebssystemen oder Hardwarekomponenten ist diese aktualisierte Version von SA MP Base Component zur Unterstützung der Hochverfügbarkeitsfunktion erforderlich.

IBM Tivoli SA MP Base Component

IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component stellt für AIX-, Linux- und Windows-Betriebssysteme HADR-Funktionalität (HADR = High Availability Disaster Recovery) bereit. SA MP Base Component ist der standardmäßig verwendete Cluster-Manager in einer IBM Data Server-Clusterumgebung auf AIX- und Linux-Betriebssystemen.

SA MP Base Component ist in DB2 Enterprise Server Edition, DB2 Workgroup Server Edition, DB2 Connect Enterprise Edition und DB2 Connect Application Server Edition unter AIX und Linux integriert. Darüber hinaus besteht eine Integration mit Express Edition für DB2 Express-C Fixed Term License (FTL) und DB2 High Availability Feature.

Auf AIX- und Linux-Betriebssystemen hängt die Version von SA MP Base Component, die installiert werden soll, von der installierten DB2-Version ab.

- Vor DB2 Version 9.5 Fixpack 6: SA MP Base Component 2.2
- DB2 Version 9.5 Fixpack 6 und spätere Fixpacks: SA MP Base Component 3.2 (AIX-Betriebssysteme) oder SA MP Base Component 3.1 (Linux-Betriebssysteme)

Unter Windows-Betriebssystemen ist SA MP Base Component 2.2 Teil des jeweiligen Produktpakets für diese DB2-Datenbankprodukte und -komponenten, es ist jedoch nicht in das DB2-Installationsprogramm integriert.

Mithilfe dieser Kopie von SA MP Base Component können Sie die Hochverfügbarkeit Ihres DB2-Datenbanksystems verwalten. Es ist nicht möglich, diese Kopie zur

Verwaltung von Nicht-DB2-Systemen zu verwenden; hierfür ist ein Upgrade für die Lizenz von SA MP Base Component erforderlich.

In einigen Umgebungen ist eine höhere Version von SA MP Base Component als die Version erforderlich, die im ursprünglichen Lieferumfang von DB2 Version 9.5 enthalten war. Version 9.5 Fixpack 6 und spätere Fixpacks enthalten höhere Versionen von SA MP Base Component, die in Umgebungen mit SUSE Linux Enterprise Server (SLES) 11-, AIX 7.1- oder POWER7-Systemen verwendet werden können. Informationen zur Vorgehensweise bei der Installation bestimmter Versionen der SA MP Base Component-Software oder zum Durchführen eines Upgrades für die SA MP Base Component-Software auf eine neue Version finden Sie unter „Installieren von IBM Tivoli SA MP Base Component" oder „Durchführen eines Upgrades für IBM Tivoli SA MP Base Component" auf Seite 82.

Weitere Informationen zu den SA MP Base Component-Versionen, die in DB2-Produktimages integriert sind, finden Sie im Information Center für Tivoli-Software. Die Liste der unterstützten Betriebssysteme ist auch auf der folgenden Website verfügbar: http://www.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html.

Installieren von IBM Tivoli SA MP Base Component

Sie können IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component entweder mit dem DB2-Installationsprogramm oder dem Installationsscript installSAM installieren; dieses befindet sich auf dem Installationsdatenträger von IBM Data Server. Sie können auch Versionen der SA MP Base Component-Software installieren, die nicht Teil des Installationsdatenträgers von IBM Data Server sind, indem Sie sie zuerst direkt von der Produktwebsite herunterladen.

Vorbereitungen

Unabhängig davon, ob Sie das DB2-Installationsprogramm, installSAM oder uninstallSAM verwenden, müssen Sie sicherstellen, dass die grundlegenden Voraussetzungen für die Installation, das Upgrade oder die Deinstallation von SA MP Base Component erfüllt sind. Siehe hierzu „Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm" auf Seite 75.

Wenn Sie SA MP Base Component bereits installiert haben, können Sie für diese Version mithilfe des DB2-Installationsprogramms oder des Installationsscripts installSAM ein Upgrade durchführen. Weitere Informationen zu einem Upgrade von SA MP Base Component finden Sie in „Durchführen eines Upgrades für IBM Tivoli SA MP Base Component" auf Seite 82.

In einigen Umgebungen ist eine höhere Version von SA MP Base Component als die Version erforderlich, die im ursprünglichen Lieferumfang von DB2 Version 9.5 enthalten war. Version 9.5 Fixpack 6 und spätere Fixpacks enthalten höhere Versionen von SA MP Base Component, die in Umgebungen mit SUSE Linux Enterprise Server (SLES) 11-, AIX 7.1- oder POWER7-Systemen verwendet werden können. Weitere Informationen zur Einschränkung für SLES 11 finden Sie unter http://www.ibm.com/developerworks/wikis/display/im/SUSE+Linux+Enterprise+Server+(SLES)+11+-+DB2+9.5.

Wenn Sie SA MP Base Component mit einem Image von Version 9.5 Fixpack 6 oder einem höheren Fixpack installieren wollen, müssen Sie die in dem Fixpackimage enthaltene Testlizenzdatei für SA MP durch eine permanente SA MP-Lizenz ersetzen, welche auf der Website von Passport Advantage verfügbar ist. Wenn Sie die

Lizenzdatei gerade nicht ersetzen können, haben Sie die Möglichkeit, mit der Option -f NOTSAMP das Upgrade von SA MP Base Component zu überspringen.

Vorgehensweise

Für eine Installation oder ein Upgrade von SA MP Base Component gibt es drei Möglichkeiten:

- Verwenden des DB2-Installationsprogramms
- Verwenden des Installationsscripts installSAM; dieses befindet sich auf dem Installationsdatenträger von IBM Data Server
- Herunterladen und Installieren neuerer Versionen

Weitere Schritte

Ziehen Sie die Diagnoseinformationen im Installationsprotokoll von SA MP Base Component zu Warnungen oder Fehlern zurate, die das DB2-Installationsprogramm oder das Installationsscript installSAM zurückgegeben hat. Weitere Informationen zum Installationsprotokoll für SA MP Base Component finden Sie in „IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle" auf Seite 93.

Installieren von IBM Tivoli SA MP Base Component mit dem DB2-Installationsprogramm:

Sie können IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component mit dem DB2-Installationsprogramm installieren.

Vorbereitung

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder dem Installationsscript installSAM installieren, müssen die grundlegenden Voraussetzungen für die Installation von SA MP Base Component erfüllt werden. Siehe hierzu „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Führen Sie die folgenden Schritte aus, wenn Sie vorhaben, die Installation mit einem Installationsimage von Version 9.5 Fixpack 6 oder einem höheren Fixpack auszuführen:

1. Wechseln Sie zur Website von Passport Advantage und rufen Sie von einer Ihrer DB2 Version 9.5-Aktivierungs-CDs eine permanente SA MP-Lizenzdatei ab. Unter AIX benötigen Sie die Datei `sam32.lic` (in Fixpack 6 oder höher verfügbar). Bei anderen Betriebssystemen benötigen Sie die Datei `sam31.lic` (in Fixpack 5 oder höher verfügbar).
2. Kopieren Sie die permanente Lizenzdatei in das Verzeichnis *`fixpackpfad`*`/db2/`*`plattform`*`/tsamp/license`. Dabei steht *fixpackpfad* für den Pfad, in dem sich das Fixpackimage befindet, und *plattform* für das verwendete Betriebssystem.
3. Entfernen Sie die Datei `sam31tb.lic` oder `sam32tb.lic` aus dem Fixpackimage. Wenn Sie diese zusätzlichen Lizenzdateien nicht löschen, schlägt die Installation fehl.
4. Setzen Sie den Installationsprozess fort.

Informationen zu dieser Task

Es gibt drei Möglichkeiten zur Verwendung des DB2-Installationsprogramms:

- DB2-Installationsassistent (Installation, Upgrade oder Deinstallation)
- Automatische Installation unter Verwendung einer Antwortdatei mit db2setup (Installation oder Upgrade) bzw. db2unins (Deinstallation)
- Befehl db2_install (Installation), Befehl installFixPack (Upgrade) oder Befehl db2_deinstall (Deinstallation)

Vor der Installation von SA MP Base Component auf einer bestimmten Maschine benötigt das DB2-Installationsprogramm vom System die folgenden Informationen:

- Ist SA MP Base Component auf dem Installationsdatenträger von IBM Data Server vorhanden?
- Wurde SA MP Base Component bereits installiert?

Das DB2-Installationsprogramm ruft das Installationsscript installSAM auf, das einige Aufgaben der SA MP Base Component-Installation übernimmt. Alternativ zur Installation von SA MP Base Component mit dem DB2-Installationsprogramm können Sie installSAM auch direkt aufrufen. Weitere Informationen zur Verwendung des Installationsscripts installSAM für die Installation von SA MP Base Component finden Sie in „Installieren von IBM Tivoli SA MP Base Component mit dem Installationsscript 'installSAM'" auf Seite 80.

Sie können die Option -l mit db2setup, db2_install oder installFixPack verwenden, um die Position anzugeben, in der das Dienstprogramm installSAM das Installationsprotokoll für SA MP Base Component ablegen soll. Weitere Informationen zum Installationsprotokoll für SA MP Base Component finden Sie in „IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle" auf Seite 93.

Vorgehensweise

- Zum Installieren von SA MP Base Component mit dem DB2-Installationsassistenten müssen Sie den DB2-Installationsassistenten ausführen und die angezeigten Anweisungen befolgen.

  Von den Informationen zum System, die vom DB2-Installationsprogramm erfasst werden, hängt ab, welche Fenster in der grafischen Oberfläche des DB2-Installationsassistenten während der Installation angezeigt werden. Beispiel: Wenn SA MP Base Component bereits installiert ist, zeigt der DB2-Installationsassistent kein Fenster zum Installieren von SA MP Base Component an.
- Wenn Sie SA MP Base Component unter Verwendung einer Antwortdatei installieren möchten, setzen Sie das Antwortdateischlüsselwort INSTALL_TSAMP auf 'Ja' (YES).

  Bei einer Installation mit einer Antwortdatei wird SA MP Base Component vom DB2-Installationsprogramm standardmäßig installiert. Wenn INSTALL_TSAMP auf 'Ja' (YES) oder auf Kommentar gesetzt bzw. in der Antwortdatei nicht enthalten ist, versucht das DB2-Installationsprogramm, SA MP Base Component zu installieren.

  Setzen Sie INSTALL_TSAMP auf 'Nein' (NO), um zu verhindern, dass SA MP Base Component vom DB2-Installationsprogramm unter Verwendung einer Antwortdatei installiert wird.
- Zur Installation von SA MP Base Component mit dem Befehl db2_install oder installFixPack können Sie die Befehle ohne SA MP Base Component-spezifische Parameter ausführen.

  Das Standardverhalten ist, SA MP Base Component zu installieren.

  Zum Verhindern der Installation von SA MP Base Component müssen Sie die Option -f NOTSAMP verwenden.

Weitere Schritte

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Installationsscript installSAM installieren, müssen nach Abschluss der Installation dieselbe Schritte befolgt werden. Weitere Informationen zu allgemeinen Schritten nach Abschluss der Installation finden Sie in „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Installieren von IBM Tivoli SA MP Base Component mit dem Installationsscript 'installSAM':

Sie können IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component mit dem Installationsscript installSAM installieren, das sich auf dem Installationsdatenträger von IBM Data Server befindet.

Vorbereitung

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder dem Installationsscript installSAM installieren, müssen die grundlegenden Voraussetzungen für die Installation von SA MP Base Component erfüllt werden. Siehe hierzu „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Führen Sie die folgenden Schritte aus, wenn Sie vorhaben, die Installation mit einem Installationsimage von Version 9.5 Fixpack 6 oder einem höheren Fixpack auszuführen:

1. Wechseln Sie zur Website von Passport Advantage und rufen Sie von einer Ihrer DB2 Version 9.5-Aktivierungs-CDs eine permanente SA MP-Lizenzdatei ab. Unter AIX benötigen Sie die Datei `sam32.lic` (in Fixpack 6 oder höher verfügbar). Bei anderen Betriebssystemen benötigen Sie die Datei `sam31.lic` (in Fixpack 5 oder höher verfügbar).
2. Kopieren Sie die permanente Lizenzdatei in das Verzeichnis *fixpackpfad*`/db2/`*plattform*`/tsamp/license`. Dabei steht *fixpackpfad* für den Pfad, in dem sich das Fixpackimage befindet, und *plattform* für das verwendete Betriebssystem.
3. Entfernen Sie die Datei `sam31tb.lic` oder `sam32tb.lic` aus dem Fixpackimage. Wenn Sie diese zusätzlichen Lizenzdateien nicht löschen, schlägt die Installation fehl.
4. Setzen Sie den Installationsprozess fort.

Führen Sie das Installationsscript installSAM aus.
Das Installationsscript installSAM befindet sich auf dem Installationsdatenträger von IBM Data Server im folgenden Verzeichnis:

`db2/<`*plattform*`>/tsamp`

Dabei steht <*plattform*> für die entsprechende Hardwareplattform.
Informationen zur Verwendung von installSAM finden Sie in Information Center für Tivoli-Software.

Weitere Schritte

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Installationsscript installSAM installieren, müssen nach Abschluss der Installation dieselbe Schritte befolgt werden. Weitere Informationen zu allgemeinen Schritten nach Abschluss der Installation finden Sie in „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Bei Verwendung von DB2 High Availability Feature mit IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component als Cluster-Manager verwendet der Datenbankmanager Scripts zur Unterstützung der DB2 HADR-Funktionalität. Diese HADR-Scripts werden automatisch installiert oder aktualisiert, wenn Sie SA MP Base Component mit dem DB2-Installationsprogramm installieren oder aktualisieren. Bei einer Installation oder Aktualisierung von SA MP Base Component mit dem Dienstprogramm installSam müssen die HADR-Scripts manuell installiert oder aktualisiert werden. Weitere Informationen zu einer manuellen Installation oder einem manuellen Upgrade der HADR-Scripts finden Sie in „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

Installieren neuerer Versionen von IBM Tivoli SA MP Base Component:

In einigen Umgebungen ist eine höhere Version von SA MP Base Component als die Version erforderlich, die im ursprünglichen Lieferumfang von DB2 Version 9.5 enthalten war. Version 9.5 Fixpack 6 und spätere Fixpacks enthalten eine höhere Version von SA MP Base Component, die in Umgebungen mit SUSE Linux Enterprise Server (SLES) 11 oder POWER7-Systemen verwendet werden kann.

Vorbereitung

Wenn Sie bereits eine Kopie von SA MP Base Component besitzen und Version 9.5 Fixpack 6 und spätere Fixpacks anwenden, wird für SA MP Base Component unter AIX- und Linux-Betriebssystemen ein Upgrade auf eine spätere Version durchgeführt. Sie müssen jedoch vor der Installation die Lizenzdatei für SA MP Base Component in den Fixpackimages separat aktualisieren. Andernfalls schlägt das Upgrade fehl.

Die Liste der unterstützten Betriebssysteme für die verschiedenen Versionen von SA MP Base Component ist auf der folgenden Website verfügbar: http://www.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html. Stellen Sie sicher, dass Sie eine kompatible Version von SA MP Base Component für das jeweilige Betriebssystem herunterladen und installieren.

Vorgehensweise

Gehen Sie wie folgt vor, um eine höhere Version von SA MP Base Component zu installieren, indem Sie diese Version herunterladen und separat installieren (anstelle der Verwendung von Version 9.5 Fixpack 6 oder eines späteren Fixpacks):

1. Rufen Sie die Seite http://www-01.ibm.com/software/tivoli/support/sys-auto-multi/ auf und klicken auf auf den Link für Downloads.
2. Wählen Sie die erforderliche Version aus und laden Sie sie herunter. Für SLES 11 ist dies beispielsweise SA MP Base Component 3.1.
3. Installieren Sie das Produkt mithilfe des Readme-Dokuments, das dem heruntergeladenen SA MP Base Component-Image beigefügt ist.
4. Ist eine frühere Version von SA MP Base Component installiert, muss die vorhandene Domäne migriert werden. Anweisungen hierzu finden Sie im Handbuch IBM Tivoli System Automation for Multiplatforms Installation and Configuration Guide Version 3.1 (IBM Form SC33-8416-01) in Kapitel 1 „Installing and upgrading System Automation for Multiplatforms", Abschnitt „Migrating System Automation for Multiplatforms". Führen Sie die Schritte im Abschnitt 'Migrating an entire domain' aus.

5. Registrieren Sie die SA MP Base Component-Lizenz wie folgt:
   ```
   samlicm -i pfad/sam31.lic
   ```

   Dabei steht *pfad* für die Position der Lizenzdatei, z. B. `db2/license`.

Gehen Sie wie folgt vor, um eine Lizenz für SA MP Base Component 3.1 zu erhalten:

1. Rufen Sie die Passport Advantage-Website unter der folgenden Adresse auf: http://www-01.ibm.com/software/howtobuy/passportadvantage/
2. Laden Sie ein DB2-Produktimage oder ein High Availability Feature-Image mit SA MP Base Component herunter.

   Ab Version 9.5 Fixpack 5 enthält jedes dieser Images Testlizenzschlüsseldateien für SA MP Base Component.

Wenn für die SA MP Base Component-Lizenzdateien auf den einzelnen Servern kein Upgrade durchgeführt wird, können einige Tivoli SA MP Base Component-Befehle nicht fehlerfrei ausgeführt werden.

Durchführen eines Upgrades für IBM Tivoli SA MP Base Component

Für ein Upgrade von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component können Sie entweder das DB2-Installationsprogramm oder das Installationsscript installSAM verwenden, das sich auf dem Installationsdatenträger IBM Data Server befindet. Sie können auch ein Upgrade auf spätere Versionen der SA MP Base Component-Software durchführen, die nicht Teil des IBM Data Server-Installationsdatenträgers sind, indem Sie sie zuerst direkt von der Produktwebsite herunterladen.

Vorbereitungen

Unabhängig davon, ob Sie das DB2-Installationsprogramm, installSAM oder uninstallSAM verwenden, müssen Sie sicherstellen, dass die grundlegenden Voraussetzungen für die Installation, das Upgrade oder die Deinstallation von SA MP Base Component erfüllt sind. Siehe hierzu „Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm” auf Seite 75.

Wenn Sie SA MP Base Component bereits installiert haben, können Sie für diese Version mithilfe des DB2-Installationsprogramms oder des Installationsscripts installSAM ein Upgrade durchführen. Weitere Informationen zu einem Upgrade von SA MP Base Component finden Sie in „Durchführen eines Upgrades für IBM Tivoli SA MP Base Component”.

In einigen Umgebungen ist eine höhere Version von SA MP Base Component als die Version erforderlich, die im ursprünglichen Lieferumfang von DB2 Version 9.5 enthalten war. Version 9.5 Fixpack 6 und spätere Fixpacks enthalten höhere Versionen von SA MP Base Component, die in Umgebungen mit SUSE Linux Enterprise Server (SLES) 11-, AIX 7.1- oder POWER7-Systemen verwendet werden können. Weitere Informationen zur Einschränkung für SLES 11 finden Sie unter http://www.ibm.com/developerworks/wikis/display/im/SUSE+Linux+Enterprise+Server+(SLES)+11+-+DB2+9.5.

Wenn Sie SA MP Base Component mit einem Image von Version 9.5 Fixpack 6 oder einem höheren Fixpack installieren wollen, müssen Sie die in dem Fixpackimage enthaltene Testlizenzdatei für SA MP durch eine permanente SA MP-Lizenz ersetzen, welche auf der Website von Passport Advantage verfügbar ist. Wenn Sie die

Lizenzdatei gerade nicht ersetzen können, haben Sie die Möglichkeit, mit der Option -f NOTSAMP das Upgrade von SA MP Base Component zu überspringen.

Einschränkungen

- Auf dem Installationsdatenträger von DB2 Version 9.5 ist SA MP Base Component Version 2.2 enthalten. Version 9.5 Fixpack 6 und spätere Fixpacks enthalten den Code für neuere Versionen von SA MP Base Component, aber nicht die permanenten Lizenzdateien für diese Versionen. Wenn Sie vor der Installation die Lizenzdateien in den Fixpackimages nicht ersetzen, schlägt das Upgrade von SA MP Base Component fehl. Weitere Informationen hierzu finden Sie im Abschnitt „Durchführen eines Upgrades für IBM Tivoli SA MP Base Component mit dem DB2-Installationsprogramm".
- SA MP Base Component unterstützt kein direktes Upgrade von Version 1 auf Version 2.2. Wenn auf Ihrem System SA MP Base Component Version 1 installiert ist, müssen Sie vor einem Upgrade auf Version 2.2 ein Upgrade von Version 1 auf Version 2.1 durchführen.
- Sie können das Upgrade von SA MP Base Component nicht mit dem DB2-Installationsprogramm durchführen, wenn auf Ihrem System mindestens eine IBM RSCT-Peerdomäne (RSCT = Reliable Scalable Cluster Technology) definiert ist.
- Die knotenweise Migration wird für Upgrades von Version 2.2 auf Version 3.1 nicht unterstützt. Die gesamte Domäne muss migriert werden. Weitere Informationen hierzu finden Sie im Handbuch IBM Tivoli System Automation for Multiplatforms Installation and Configuration Guide Version 3.1 (IBM Form SC33-8416-01) in Kapitel 1 „Installing and upgrading System Automation for Multiplatforms", Abschnitt „Migrating System Automation for Multiplatforms". Führen Sie die Schritte im Abschnitt 'Migrating an entire domain' aus.

Vorgehensweise

Für eine Installation oder ein Upgrade von SA MP Base Component gibt es drei Möglichkeiten:

- Verwenden des DB2-Installationsprogramms
- Verwenden des Installationsscripts installSAM; dieses befindet sich auf dem Installationsdatenträger von IBM Data Server
- Herunterladen und Installieren neuerer Versionen

Weitere Schritte

Ziehen Sie die Diagnoseinformationen im Installationsprotokoll von SA MP Base Component zu Warnungen oder Fehlern zurate, die das DB2-Installationsprogramm oder das Installationsscript installSAM zurückgegeben hat. Weitere Informationen zum Installationsprotokoll für SA MP Base Component finden Sie in „IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle" auf Seite 93.

Durchführen eines Upgrades für IBM Tivoli SA MP Base Component mit dem DB2-Installationsprogramm:

Sie können mit dem DB2-Installationsprogramm ein Upgrade für IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component durchführen.

Vorbereitung

Unabhängig davon, ob Sie für das Upgrade von SA MP Base Component das DB2-Installationsprogramm oder das Installationsscript installSAM verwenden, das sich

auf dem Installationsdatenträger von IBM Data Server befindet, müssen die grundlegenden Voraussetzungen für ein Upgrade von SA MP Base Component erfüllt werden. Siehe hierzu „Durchführen eines Upgrades für IBM Tivoli SA MP Base Component" auf Seite 82.

Um mit dem Befehl installFixPack ein Upgrade von SA MP Base Component durchzuführen, müssen Sie vor der Installation die folgenden Schritte ausführen, wenn Sie Fixpack 6 oder ein höheres Fixpack verwenden:

1. Wechseln Sie zur Website von Passport Advantage und rufen Sie von einer Ihrer DB2 Version 9.5-Aktivierungs-CDs eine permanente SA MP-Lizenzdatei ab. Unter AIX benötigen Sie die Datei `sam32.lic` (in Fixpack 6 oder höher verfügbar). Bei anderen Betriebssystemen benötigen Sie die Datei `sam31.lic` (in Fixpack 5 oder höher verfügbar).
2. Kopieren Sie die permanente Lizenzdatei in das Verzeichnis *fixpackpfad*`/db2/`*plattform*`/tsamp/license`. Dabei steht *fixpackpfad* für den Pfad, in dem sich das Fixpackimage befindet, und *plattform* für das verwendete Betriebssystem.
3. Entfernen Sie die Datei `sam31tb.lic` oder `sam32tb.lic` aus dem Fixpackimage. Wenn Sie diese zusätzlichen Lizenzdateien nicht löschen, schlägt die Installation fehl.
4. Setzen Sie den Upgradeprozess fort.

Informationen zu dieser Task

Es gibt drei Möglichkeiten zur Verwendung des DB2-Installationsprogramms:

- DB2-Installationsassistent (Installation, Upgrade oder Deinstallation)
- Automatische Installation unter Verwendung einer Antwortdatei mit db2setup (Installation oder Upgrade) bzw. db2unins (Deinstallation)
- Befehl db2_install (Installation), Befehl installFixPack (Upgrade) oder Befehl db2_deinstall (Deinstallation)

Bevor ein Upgrade von SA MP Base Component auf einer bestimmten Maschine durchgeführt werden kann, benötigt das DB2-Installationsprogramm vom System die folgenden Informationen:

- Wenn SA MP Base Component bereits installiert ist: Ist die bereits installierte Version von SA MP Base Component älter als die Version von SA MP Base Component, die sich auf dem Installationsdatenträger von IBM Data Server befindet?

Das DB2-Installationsprogramm ruft das Installationsscript installSAM auf, das einige Aufgaben der Upgrade-Operation für SA MP Base Component übernimmt. Sie können installSAM auch direkt aufrufen. Weitere Informationen zur Verwendung des Installationsscripts installSAM für ein Upgrade von SA MP Base Component finden Sie in „Upgrade von IBM Tivoli SA MP Base Component mit dem Installationsscript installSAM" auf Seite 86.

Sie können die Option -l mit db2setup, db2_install oder installFixPack verwenden, um die Position anzugeben, in der das Dienstprogramm installSAM das Installationsprotokoll für SA MP Base Component ablegen soll. Weitere Informationen zum Installationsprotokoll für SA MP Base Component finden Sie in „IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle" auf Seite 93.

Vorgehensweise

- Für ein Upgrade von SA MP Base Component mit dem DB2-Installationsassistenten müssen Sie den DB2-Installationsassistenten ausführen und die angezeigten Anweisungen befolgen.

Von den Informationen zum System, die vom DB2-Installationsprogramm erfasst werden, hängt ab, welche Fenster in der grafischen Oberfläche des DB2-Installationsassistenten während des Upgrades angezeigt werden. Beispiel: Wenn die bereits installierte Version von SA MP Base Component mit der Version von SA MP Base Component identisch ist, die sich auf dem Installationsdatenträger von IBM Data Server befindet, oder wenn es sich um eine höhere Version handelt, zeigt der DB2-Installationsassistent kein Fenster für ein Upgrade von SA MP Base Component an.

- Für ein Upgrade von SA MP Base Component mit einer Antwortdatei müssen Sie das Antwortdateischlüsselwort INSTALL_TSAMP auf 'Ja' (YES) setzen.

  Bei einer Installation mit einer Antwortdatei führt das DB2-Installationsprogramm standardmäßig ein Upgrade von SA MP Base Component durch, wenn die bereits installierte Version von SA MP Base Component älter ist als die Version, die sich auf dem Installationsdatenträger von IBM Data Server befindet. Wenn INSTALL_TSAMP auf 'Ja' (YES) oder auf Kommentar gesetzt bzw. in der Antwortdatei nicht enthalten ist, versucht das DB2-Installationsprogramm, ein Upgrade für SA MP Base Component durchzuführen.

  Setzen Sie INSTALL_TSAMP auf 'Nein' (NO), um zu verhindern, dass das DB2-Installationsprogramm im Rahmen einer Installation mit einer Antwortdatei ein Upgrade für SA MP Base Component durchführt.
- Für ein Upgrade von SA MP Base Component mit dem Befehl db2_install oder installFixPack können Sie die Befehle ohne SA MP Base Component-spezifische Parameter ausführen.

  db2_install führt standardmäßig ein Upgrade von SA MP Base Component durch, wenn die bereits installierte Version von SA MP Base Component älter ist als die Version, die sich auf dem Installationsdatenträger von IBM Data Server befindet.

  Zum Verhindern des Upgrades von SA MP Base Component müssen Sie die Option -f NOTSAMP verwenden.

Weitere Schritte

Wenn das Upgrade von SA MP Base Component aufgrund von Diskrepanzen zwischen der Lizenzdatei des Fixpackimages und der Lizenzdatei auf Ihrem Computer fehlgeschlagen ist, führen Sie die folgenden Schritte aus:

1. Aktualisieren Sie die vorhandene Lizenz mit der korrekten permanenten SA MP-Lizenz, die sich auf einer der DB2 Version 9.5-Aktivierungs-CDs befindet.
2. Führen Sie mit einer der folgenden Methoden eine Neuinstallation von SA MP Base Component aus:
   a. Verwenden Sie das Script installSAM.
   b. Wenden Sie das Fixpack mit dem folgenden Befehl erneut an:
      ```
      fixpackimagepfad/installFixPack -f level -b basisinstallationspfad
      ```

      Dabei steht *fixpackimagepfad* für den Pfad, in dem sich das Fixpackimage befindet, und *basisinstallationspfad* für den Pfad, in dem SA MP Base Component installiert werden soll.

      Anmerkung: Wenn Sie das DB2-Installationsprogramm zwingen wollen, das Fixpack unabhängig von der gerade installierten DB2-Version anzuwenden, müssen Sie die Option **-f level** verwenden.

Unabhängig davon, ob das Upgrade für SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Installationsscript installSAM durchgeführt

wird, müssen nach Abschluss des Upgrades dieselbe Schritte befolgt werden. Weitere Informationen zu allgemeinen Schritten nach Abschluss des Upgrades finden Sie im Abschnitt „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Upgrade von IBM Tivoli SA MP Base Component mit dem Installationsscript installSAM:

Für ein Upgrade von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component können Sie das Installationsscript installSAM verwenden, das sich auf dem Installationsdatenträger von IBM Data Server befindet.

Vorbereitung

Unabhängig davon, ob Sie für das Upgrade von SA MP Base Component das DB2-Installationsprogramm oder das Installationsscript installSAM verwenden, das sich auf dem Installationsdatenträger von IBM Data Server befindet, müssen die grundlegenden Voraussetzungen für ein Upgrade von SA MP Base Component erfüllt werden. Siehe hierzu „Durchführen eines Upgrades für IBM Tivoli SA MP Base Component" auf Seite 82.

Um mit dem Befehl installFixPack ein Upgrade von SA MP Base Component durchzuführen, müssen Sie vor der Installation die folgenden Schritte ausführen, wenn Sie Fixpack 6 oder ein höheres Fixpack verwenden:

1. Wechseln Sie zur Website von Passport Advantage und rufen Sie von einer Ihrer DB2 Version 9.5-Aktivierungs-CDs eine permanente SA MP-Lizenzdatei ab. Unter AIX benötigen Sie die Datei `sam32.lic` (in Fixpack 6 oder höher verfügbar). Bei anderen Betriebssystemen benötigen Sie die Datei `sam31.lic` (in Fixpack 5 oder höher verfügbar).
2. Kopieren Sie die permanente Lizenzdatei in das Verzeichnis *fixpackpfad*`/db2/`*plattform*`/tsamp/license`. Dabei steht *fixpackpfad* für den Pfad, in dem sich das Fixpackimage befindet, und *plattform* für das verwendete Betriebssystem.
3. Entfernen Sie die Datei `sam31tb.lic` oder `sam32tb.lic` aus dem Fixpackimage. Wenn Sie diese zusätzlichen Lizenzdateien nicht löschen, schlägt die Installation fehl.
4. Setzen Sie den Upgradeprozess fort.

Führen Sie das Installationsscript installSAM aus.
Das Installationsscript installSAM befindet sich auf dem Installationsdatenträger von IBM Data Server im folgenden Verzeichnis:

`db2/<`*plattform*`>/tsamp`

Dabei steht <*plattform*> für die entsprechende Hardwareplattform.
Informationen zur Verwendung von installSAM finden Sie in Information Center für Tivoli-Software.

Weitere Schritte

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Installationsscript installSAM installieren, müssen nach Abschluss der Installation dieselbe Schritte befolgt werden. Weitere Informationen zu allgemeinen Schritten nach Abschluss der Installation finden Sie in „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Bei Verwendung von DB2 High Availability Feature mit IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component als Cluster-Manager verwendet der Datenbankmanager Scripts zur Unterstützung der DB2 HADR-Funktionalität. Diese HADR-Scripts werden automatisch installiert oder aktualisiert, wenn Sie SA MP Base Component mit dem DB2-Installationsprogramm installieren oder aktualisieren. Bei einer Installation oder Aktualisierung von SA MP Base Component mit dem Dienstprogramm installSam müssen die HADR-Scripts manuell installiert oder aktualisiert werden. Weitere Informationen zu einer manuellen Installation oder einem manuellen Upgrade der HADR-Scripts finden Sie in „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

Installieren neuerer Versionen von IBM Tivoli SA MP Base Component:

In einigen Umgebungen ist eine höhere Version von SA MP Base Component als die Version erforderlich, die im ursprünglichen Lieferumfang von DB2 Version 9.5 enthalten war. Version 9.5 Fixpack 6 und spätere Fixpacks enthalten eine höhere Version von SA MP Base Component, die in Umgebungen mit SUSE Linux Enterprise Server (SLES) 11 oder POWER7-Systemen verwendet werden kann.

Vorbereitung

Wenn Sie bereits eine Kopie von SA MP Base Component besitzen und Version 9.5 Fixpack 6 und spätere Fixpacks anwenden, wird für SA MP Base Component unter AIX- und Linux-Betriebssystemen ein Upgrade auf eine spätere Version durchgeführt. Sie müssen jedoch vor der Installation die Lizenzdatei für SA MP Base Component in den Fixpackimages separat aktualisieren. Andernfalls schlägt das Upgrade fehl.

Die Liste der unterstützten Betriebssysteme für die verschiedenen Versionen von SA MP Base Component ist auf der folgenden Website verfügbar: http://www.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html. Stellen Sie sicher, dass Sie eine kompatible Version von SA MP Base Component für das jeweilige Betriebssystem herunterladen und installieren.

Vorgehensweise

Gehen Sie wie folgt vor, um eine höhere Version von SA MP Base Component zu installieren, indem Sie diese Version herunterladen und separat installieren (anstelle der Verwendung von Version 9.5 Fixpack 6 oder eines späteren Fixpacks):

1. Rufen Sie die Seite http://www-01.ibm.com/software/tivoli/support/sys-auto-multi/ auf und klicken auf auf den Link für Downloads.
2. Wählen Sie die erforderliche Version aus und laden Sie sie herunter. Für SLES 11 ist dies beispielsweise SA MP Base Component 3.1.
3. Installieren Sie das Produkt mithilfe des Readme-Dokuments, das dem heruntergeladenen SA MP Base Component-Image beigefügt ist.
4. Ist eine frühere Version von SA MP Base Component installiert, muss die vorhandene Domäne migriert werden. Anweisungen hierzu finden Sie im Handbuch IBM Tivoli System Automation for Multiplatforms Installation and Configuration Guide Version 3.1 (IBM Form SC33-8416-01) in Kapitel 1 „Installing and upgrading System Automation for Multiplatforms", Abschnitt „Migrating System Automation for Multiplatforms". Führen Sie die Schritte im Abschnitt 'Migrating an entire domain' aus.
5. Registrieren Sie die SA MP Base Component-Lizenz wie folgt:
   ```
   samlicm -i pfad/sam31.lic
   ```

Dabei steht *pfad* für die Position der Lizenzdatei, z. B. `db2/license`.

Gehen Sie wie folgt vor, um eine Lizenz für SA MP Base Component 3.1 zu erhalten:

1. Rufen Sie die Passport Advantage-Website unter der folgenden Adresse auf: http://www-01.ibm.com/software/howtobuy/passportadvantage/
2. Laden Sie ein DB2-Produktimage oder ein High Availability Feature-Image mit SA MP Base Component herunter.

   Ab Version 9.5 Fixpack 5 enthält jedes dieser Images Testlizenzschlüsseldateien für SA MP Base Component.

Wenn für die SA MP Base Component-Lizenzdateien auf den einzelnen Servern kein Upgrade durchgeführt wird, können einige Tivoli SA MP Base Component-Befehle nicht fehlerfrei ausgeführt werden.

Deinstallieren von IBM Tivoli SA MP Base Component

Für eine Deinstallation von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component können Sie entweder das DB2-Installationsprogramm oder das Installationsscript uninstallSAM verwenden, das sich auf dem Installationsdatenträger von IBM Data Server befindet.

Vorbereitungen

Unabhängig davon, ob Sie das DB2-Installationsprogramm, installSAM oder uninstallSAM verwenden, müssen Sie sicherstellen, dass die grundlegenden Voraussetzungen für die Installation, das Upgrade oder die Deinstallation von SA MP Base Component erfüllt sind. Siehe hierzu „Installieren und Durchführen eines Upgrades von SA MP Base Component mit dem DB2-Installationsprogramm" auf Seite 75.

Vorgehensweise

Es gibt zwei Möglichkeiten zum Deinstallieren von SA MP Base Component:

- Verwenden des DB2-Installationsprogramms
- Verwenden des Deinstallationsscripts uninstallSAM, das sich auf dem Installationsdatenträger von IBM Data Server befindet

Weitere Schritte

Diagnoseinformationen zu Warnungen oder Fehlern, die vom DB2-Installationsprogramm oder vom Deinstallationsscript uninstallSAM zurückgegeben wurden, finden Sie im Deinstallationsprotokoll für SA MP Base Component. Weitere Informationen zum Deinstallationsprotokoll für SA MP Base Component finden Sie in „IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle" auf Seite 93.

Deinstallieren von IBM Tivoli SA MP Base Component mit dem DB2-Installationsprogramm:

Sie können IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component mit dem DB2-Installationsprogramm deinstallieren.

Vorbereitung

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Deinstallationsscript uninstallSAM deinstallieren, das sich auf dem Installationsdatenträger von IBM Data Server befindet, müssen die grundlegenden Voraussetzungen für eine Deinstallation von SA MP Base Component erfüllt werden. Siehe hierzu „Deinstallieren von IBM Tivoli SA MP Base Component" auf Seite 88.

Informationen zu dieser Task

Es gibt drei Möglichkeiten zur Verwendung des DB2-Installationsprogramms:
- DB2-Installationsassistent (Installation, Upgrade oder Deinstallation)
- Automatische Installation unter Verwendung einer Antwortdatei mit db2setup (Installation oder Upgrade) bzw. db2unins (Deinstallation)
- Befehl db2_install (Installation), Befehl installFixPack (Upgrade) oder Befehl db2_deinstall (Deinstallation)

Das DB2-Installationsprogramm ruft das Deinstallationsscript uninstallSAM auf, das einige Aufgaben der Deinstallation von SA MP Base Component übernimmt. Sie können uninstallSAM auch direkt aufrufen. Weitere Informationen zur Verwendung des Scripts uninstallSAM für die Deinstallation von SA MP Base Component finden Sie in „Installieren von IBM Tivoli SA MP Base Component mit dem Installationsscript 'installSAM'" auf Seite 80.

Sie können die Option -l mit db2setup, db2_install oder installFixPack verwenden, um die Position anzugeben, in der das Dienstprogramm installSAM das Installationsprotokoll für SA MP Base Component ablegen soll. Weitere Informationen zum Installationsprotokoll für SA MP Base Component finden Sie in „IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle" auf Seite 93.

Vorgehensweise

- Zum Deinstallieren von SA MP Base Component mit dem DB2-Installationsassistenten müssen Sie den DB2-Installationsassistenten ausführen und die angezeigten Anweisungen befolgen.

  Von den Informationen zum System, die vom DB2-Installationsprogramm erfasst werden, hängt ab, welche Fenster in der grafischen Oberfläche des DB2-Installationsassistenten während der Deinstallation angezeigt werden. Beispiel: Wenn SA MP Base Component nicht installiert ist, zeigt der DB2-Installationsassistent kein Fenster zum Deinstallieren von SA MP Base Component an.
- Wenn Sie SA MP Base Component unter Verwendung einer Antwortdatei deinstallieren möchten, setzen Sie das Antwortdateischlüsselwort INSTALL_TSAMP auf 'Ja' (YES).

  Bei einer Deinstallationsoperation mit einer Antwortdatei wird SA MP Base Component vom DB2-Installationsprogramm nicht standardmäßig deinstalliert. Wenn INSTALL_TSAMP auf 'Nein' (NO) oder auf Kommentar gesetzt bzw. in der Antwortdatei nicht enthalten ist, versucht das DB2-Installationsprogramm nicht, SA MP Base Component zu deinstallieren.
- Zum Deinstallieren von SA MP Base Component mit db2_deinstall können Sie db2_deinstall mit der Option -a -F TSAMP ausführen.

  Bei der Ausführung von db2_deinstall wird SA MP Base Component vom DB2-Installationsprogramm nicht standardmäßig deinstalliert.

Weitere Schritte

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder dem Deinstallationsscript uninstallSAM deinstallieren, das sich auf dem Installationsdatenträger von IBM Data Server befindet, müssen nach Abschluss der Deinstallation dieselben Schritte befolgt werden. Weitere Informationen zu allgemeinen Schritten nach Abschluss der Deinstallation finden Sie in „Deinstallieren von IBM Tivoli SA MP Base Component" auf Seite 88.

Deinstallieren von IBM Tivoli SA MP Base Component mit dem Deinstallationsscript uninstallSAM:

Sie können IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component mit dem Deinstallationsscript uninstallSAM deinstallieren, das sich auf dem Installationsdatenträger von IBM Data Server befindet.

Vorbereitung

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Deinstallationsscript uninstallSAM deinstallieren, das sich auf dem Installationsdatenträger von IBM Data Server befindet, müssen die grundlegenden Voraussetzungen für eine Deinstallation von SA MP Base Component erfüllt werden. Siehe hierzu „Deinstallieren von IBM Tivoli SA MP Base Component" auf Seite 88.

Führen Sie das Deinstallationsscript uninstallSAM aus.
Das Deinstallationsscript uninstallSAM befindet sich auf dem Installationsdatenträger von IBM Data Server im folgenden Verzeichnis:

```
db2/<plattform>/tsamp
```

Dabei steht *<plattform>* für die entsprechende Hardwareplattform.
Informationen zur Verwendung von uninstallSAM finden Sie in Information Center für Tivoli-Software.

Weitere Schritte

Unabhängig davon, ob Sie SA MP Base Component mit dem DB2-Installationsprogramm oder mit dem Installationsscript installSAM installieren, müssen nach Abschluss der Installation dieselbe Schritte befolgt werden. Weitere Informationen zu allgemeinen Schritten nach Abschluss der Installation finden Sie in „Installieren von IBM Tivoli SA MP Base Component" auf Seite 77.

Bei Verwendung von DB2 High Availability Feature mit IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component als Cluster-Manager verwendet der Datenbankmanager Scripts zur Unterstützung der DB2 HADR-Funktionalität. Diese HADR-Scripts werden automatisch deinstalliert, wenn Sie db2_deinstall zum Deinstallieren von SA MP Base Component ausführen. Bei einer Deinstallation von SA MP Base Component mit dem Dienstprogramm uninstallSam müssen die HADR-Scripts manuell deinstalliert werden. Weitere Informationen zur manuellen Deinstallation der HADR-Scripts finden Sie in „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component

Bei Verwendung von DB2 High Availability Feature mit IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component als Cluster-Manager verwendet der Datenbankmanager Scripts zur Unterstützung der DB2 HADR-Funktionalität. Für eine Installation, ein Upgrade oder eine Deinstallation dieser HADR-Scripts für SA MP Base Component können Sie das DB2-Installationsprogramm oder die Scripts installSAM und uninstallSAM verwenden, die sich auf dem Installationsdatenträger von IBM Data Server befinden.

Vorbereitung

- Für eine Installation, ein Upgrade oder eine Deinstallation der HADR-Scripts für SA MP Base Component mit dem DB2-Installationsprogramm oder den Scripts installSAM und uninstallSAM, die sich auf dem Installationsdatenträger von IBM Data Server befinden, müssen Sie DB2 High Availability Feature erwerben.
- Für eine Installation, ein Upgrade oder eine Deinstallation der HADR-Scripts für SA MP Base Component müssen Sie über eine Rootberechtigung verfügen.

  Bei einer Installation von IBM Data Server, die nicht als Root ausgeführt wird, können Sie die HADR-Scripts für SA MP Base Component separat vom Installationsdatenträger von IBM Data Server installieren. Wenn Sie die HADR-Scripts für SA MP Base Component separat installieren, benötigen Sie ebenfalls eine Rootberechtigung.

Vorgehensweise

Für eine Installation, ein Upgrade oder eine Deinstallation der HADR-Scripts für SA MP Base Component gibt es zwei Möglichkeiten :

- Verwenden des DB2-Installationsprogramms
- Manuelles Installieren vom Installationsdatenträger von IBM Data Server

Ergebnisse

Wenn Sie die HADR-Scripts für SA MP Base Component installieren, befinden sich die Scripts im folgenden Verzeichnis:

```
/usr/sbin/rsct/sapolicies/db2
```

Wenn Sie die HADR-Scripts für SA MP Base Component deinstallieren, können Sie die HADR-Funktionalität in einem von SA MP Base Component verwalteten Cluster nicht länger verwenden.

Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component mit dem DB2-Installationsprogramm:

Für eine Installation, ein Upgrade oder eine Deinstallation von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component DB2 HADR-Scripts kann das DB2-Installationsprogramm verwendet werden.

Vorbereitung

Unabhängig davon, ob Sie das DB2-Installationsprogramm verwenden oder die HADR-Scripts für SA MP Base Component manuell installieren, deinstallieren bzw. ein Upgrade durchführen, müssen die grundlegenden Voraussetzungen für eine Installation, ein Upgrade oder eine Deinstallation der HADR-Scripts für SA MP Base

Component erfüllt werden. Siehe hierzu „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

Informationen zu dieser Task

Es gibt drei Möglichkeiten zur Verwendung des DB2-Installationsprogramms:

- DB2-Installationsassistent (Installation, Upgrade oder Deinstallation)
- Automatische Installation unter Verwendung einer Antwortdatei mit db2setup (Installation oder Upgrade) bzw. db2unins (Deinstallation)
- Befehl db2_install (Installation), Befehl installFixPack (Upgrade) oder Befehl db2_deinstall (Deinstallation)

Vorgehensweise

1. Führen Sie zum Installieren der HADR-Scripts für SA MP Base Component das DB2-Installationsprogramm aus.

   Die HADR-Scripts für SA MP Base Component werden vom DB2-Installationsprogramm standardmäßig installiert, wenn SA MP Base Component installiert ist bzw. installiert wird und die Scripts nicht bereits installiert sind.
2. Führen Sie das DB2-Installationsprogramm aus, um ein Upgrade für die HADR-Scripts für SA MP Base Component durchzuführen.

   Das DB2-Installationsprogramm führt standardmäßig ein Upgrade für die HADR-Scripts für SA MP Base Component durch, wenn SA MP Base Component installiert ist bzw. installiert wird und die bereits installierten Scripts eine niedrigere Version aufweisen als die Scripts, die sich auf dem Installationsdatenträger von IBM Data Server befinden.
3. Führen Sie zum Deinstallieren der HADR-Scripts für SA MP Base Component das DB2-Installationsprogramm aus.

Ergebnisse

Unabhängig davon, ob Sie das DB2-Installationsprogramm verwenden oder die HADR-Scripts für SA MP Base Component manuell installieren, deinstallieren bzw. ein Upgrade durchführen, die Resultate sind im Allgemeinen identisch. Siehe hierzu „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

Manuelles Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component:

Sie können die DB2 HADR-Scripts für IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component über den Installationsdatenträger von IBM Data Server manuell installieren oder deinstallieren bzw. ein manuelles Upgrade für diese Scripts durchführen.

Vorbereitung

Unabhängig davon, ob Sie das DB2-Installationsprogramm verwenden oder die HADR-Scripts für SA MP Base Component manuell installieren, deinstallieren bzw. ein Upgrade durchführen, müssen die grundlegenden Voraussetzungen für eine Installation, ein Upgrade oder eine Deinstallation der HADR-Scripts für SA MP Base Component erfüllt werden. Siehe hierzu „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

Informationen zu dieser Task

Bei der Installation von SA MP Base Component werden die HADR-Scripts für SA MP Base Component vom DB2-Installationsprogramm automatisch installiert. Bei einer manuellen Installation oder einem manuellen Upgrade von SA MP Base Component muss auch eine manuelle Installation bzw. ein manuelles Upgrade der HADR-Scripts für SA MP Base Component erfolgen. Das DB2-Installationsprogramm deinstalliert die HADR-Scripts für SA MP Base Component nicht, deshalb müssen diese Scripts manuell deinstalliert werden.

Vorgehensweise

Verwenden Sie zum manuellen Installieren, Aktualisieren oder Deinstallieren der HADR-Scripts für SA MP Base Component das Dienstprogramm db2cptsa.

Ergebnisse

Unabhängig davon, ob Sie das DB2-Installationsprogramm verwenden oder die HADR-Scripts für SA MP Base Component manuell installieren, deinstallieren bzw. ein Upgrade durchführen, die Resultate sind im Allgemeinen identisch. Siehe hierzu „Installieren, Aktualisieren und Deinstallieren von HADR-Scripts für IBM Tivoli SA MP Base Component" auf Seite 91.

IBM Tivoli SA MP Base Component - Installations- und Deinstallationsprotokolle

Diagnoseinformationen, Warnungen und Fehlernachrichten, die sich auf eine Installation, ein Upgrade oder eine Deinstallation von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component beziehen, werden in SA MP Base Component-spezifischen Installations- und Deinstallationsprotokollen gespeichert.

Für eine Installation, ein Upgrade oder eine Deinstallation von SA MP Base Component können Sie das DB2-Installationsprogramm oder die Scripts installSAM und uninstallSAM verwenden, die sich auf dem Installationsdatenträger von IBM Data Server befinden. Das DB2-Installationsprogramm verwendet die Dienstprogramme installSAM und uninstallSAM ebenfalls, um einen Teil der Installations-, Upgrade- und Deinstallationsoperationen auszuführen. Das Dienstprogramm installSAM generiert eine Reihe von Protokolldateien, die sequenziell benannt werden:

`/tmp/installSAM.<protokollnummer>.log`

Dabei steht *protokollnummer* für die jeweilige Protokolldatei in der Sequenz.

Sie können die Option -l mit db2setup, db2_install oder installFixPack verwenden, um anzugeben, in welcher Position das Dienstprogramm installSAM das Installationsprotokoll für SA MP Base Component speichern soll.

Das Dienstprogramm uninstallSAM generiert eine Reihe von Protokolldateien, die sequenziell benannt werden:

`/tmp/uninstallSAM.<protokollnummer>.log`

Dabei steht *protokollnummer* für die jeweilige Protokolldatei in der Sequenz.

Sie können die Option -l mit db2unins oder db2_deinstall verwenden, um anzugeben, in welcher Position das Dienstprogramm uninstallSAM das Deinstallationsprotokoll für SA MP Base Component speichern soll.

Lizenzbedingungen für die Verwendung des in IBM Data Server integrierten Produkts IBM Tivoli SA MP Base Component

Die Verwendung des in IBM Data Server integrierten Produkts IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component unterliegt bestimmten Bedingungen.

Sie können das in IBM Data Server mit DB2 HADR-Funktionalität integrierte Produkt SA MP Base Component verwenden, wenn Sie eine Lizenz für eines der folgenden Produkte erworben haben:

- DB2 Enterprise Server Edition
- DB2 Connect Enterprise Edition
- DB2 Workgroup Server Edition

Sie können das in IBM Data Server mit HADR-Funktionalität integrierte Produkt SA MP Base Component verwenden, wenn Sie eine Lizenz für DB2 High Availability Feature und das folgende Produkt erworben haben:

- DB2 Express Edition

Sie können eine Probelizenz des in IBM Data Server mit HADR-Funktionalität integrierten Produkts SA MP Base Component verwenden, wenn Sie eine Probelizenz für eines der folgenden Produkte erworben haben:

- DB2 Enterprise Server Edition
- DB2 Connect Enterprise Edition
- DB2 Workgroup Server Edition
- DB2 Express Edition

Unterstützte Software und Hardware für IBM Tivoli SA MP Base Component

IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist in IBM Data Server unter AIX- und Linux-Betriebssystemen integriert. Darüber hinaus ist das Produkt Teil des Produktpakets von IBM Data Server unter Windows.

SA MP Base Component ist in die folgenden DB2-Datenbankprodukte und -komponenten integriert bzw. Teil des entsprechenden Produktpakets:

- DB2 Enterprise Server Edition
- DB2 Connect Enterprise Edition
- DB2 Workgroup Server Edition
- DB2 Express-C mit Fixed Term License (FTL)
- DB2 Express Edition mit DB2 High Availability Feature

Auf AIX- und Linux-Betriebssystemen hängt die Version von SA MP Base Component, die installiert werden soll, von der installierten DB2-Version ab.

- Vor DB2 Version 9.5 Fixpack 6: SA MP Base Component 2.2
- DB2 Version 9.5 Fixpack 6 und spätere Fixpacks: SA MP Base Component 3.2 (AIX-Betriebssysteme) oder SA MP Base Component 3.1 (Linux-Betriebssysteme)

Die folgenden Betriebssysteme und Hardwarekomponenten werden von SA MP Base Component 2.2 unterstützt:

- AIX Version 5.3 und 6.1 auf der folgenden Hardware:
  - eServer pSeries
  - IBM System p

  - IBM System p5
- Linux-Variante:
  - Red Hat Enterprise Linux (RHEL) 4 Aktualisierung 4
  - RHEL 5
  - SUSE Linux Enterprise Server (SLES) 9 Service-Pack 3
  - SUSE SLES 10 Service-Pack 1

  Auf der folgenden Hardware:
  - x86-32-Bit-Intel®- und -AMD-Prozessoren (Intel Pentium®, Intel® Xeon® und AMD)
  - x64 (64-Bit-AMD64- und Intel EM64T-Prozessoren)
  - POWER (IBM eServer OpenPower-, System i- oder pSeries-Systeme, die Linux unterstützen)
  - eServer System z oder System z9
- Microsoft® Windows Server 2003 R2 Standard Edition (32-Bit)
- Microsoft Windows Server 2003 R2 Enterprise Edition (32-Bit)

Die folgenden Betriebssysteme und Hardwarekomponenten werden von der in Version 9.5 Fixpack 6 und spätere Fixpacks integrierten SA MP Base Component 3.1-Version unterstützt:

- IBM AIX Version 5.3 und AIX 6.1 auf der folgenden Hardware:
  - eServer pSeries
  - IBM Power Systems
- Linux-Variante:
  - Red Hat Enterprise Linux (RHEL) 5 Aktualisierung 2
  - SUSE SLES 10 Service-Pack 2
  - SUSE SLES 11

  Auf der folgenden Hardware:
  - x86-32-Bit-Intel- und -AMD-Prozessoren (Intel Pentium, Intel Xeon® und AMD)
  - x64 (64-Bit-AMD64- und Intel EM64T-Prozessoren)
  - POWER (IBM eServer OpenPower-, System i- oder pSeries-Systeme, die Linux unterstützen)
  - eServer System z oder System z9

Die folgenden Betriebssysteme und Hardwarekomponenten werden von der in Version 9.5 Fixpack 6 und spätere Fixpacks integrierten SA MP Base Component 3.2-Version unterstützt:

- IBM AIX Version 5.3, AIX 6.1 und AIX 7.1 auf der folgenden Hardware:
  - eServer pSeries
  - IBM Power Systems

Für einige Umgebungen mit neueren Betriebssystemen oder Hardwarekomponenten ist SA MP Base Component Version 3.1 oder Version 3.2 zur Unterstützung der Hochverfügbarkeitsfunktion erforderlich. Wenn Sie die Hochverfügbarkeitsfunktionalität nutzen möchten, müssen Sie sicherstellen, dass das verwendete System die Voraussetzungen für SA MP Base Component erfüllt. Einzelheiten hierzu finden Sie in den Handbüchern zur Installation und Konfiguration im Information Center für Tivoli-Software.

Wenn Sie die integrierte oder im Produktpaket enthaltene Kopie von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component nicht verwenden, können Sie über die folgende Website eine vollständige Liste der unterstützten Betriebssysteme abrufen: http://www.ibm.com/software/tivoli/products/sys-auto-linux/platforms.html.

Automatisches Konfigurieren eines Clusters mit DB2 High Availability Feature

In einer Clusterumgebung erfordern bestimmte Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz entsprechende Änderungen an der Clusterkonfiguration. Mit DB2 High Availability Feature kann der Datenbankmanager automatisch Änderungen an der Konfiguration des Cluster-Managers anfordern, wenn bestimmte Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz ausgeführt werden.

Vorbereitung

Damit der Datenbankmanager die erforderliche Clusterkonfiguration für Datenbankverwaltungstasks vornehmen kann, müssen Sie *die Instanz für hohe Verfügbarkeit konfigurieren*, indem Sie mit db2haicu eine *Clusterdomäne* für die Instanz erstellen. Weitere Informationen hierzu finden Sie in „Konfigurieren einer Clusterumgebung mit db2haicu" auf Seite 97.

Vorgehensweise

Bei Ausführung der folgenden Konfigurations- und Verwaltungsoperationen der Datenbankmanagerinstanz nimmt der Datenbankmanager automatisch die entsprechende Cluster-Manager-Konfiguration für Sie vor:

- Starten einer Datenbank mit START DATABASE oder db2start
- Stoppen einer Datenbank mit STOP DATABASE oder db2stop
- Erstellen einer Datenbank mit CREATE DATABASE
- Hinzufügen von Speicher mit CREATE TABLESPACE
- Entfernen von Speicher mit ALTER TABLESPACE DROP oder DROP TABLESPACE
- Löschen einer Datenbank mit DROP TABLESPACE
- Wiederherstellen einer Datenbank mit RESTORE DATABASE oder db2Restore
- Angeben der Tabellenbereichscontainer für einen umgeleiteten Restore mit SET TABLESPACE CONTAINERS
- Aktualisierendes Wiederherstellen einer Datenbank mit ROLLFORWARD DATABASE oder db2Rollforward
- Recovery einer Datenbank mit RECOVER DATABASE oder db2Recover
- Erstellen von Ereignismonitoren mit CREATE EVENT MONITOR
- Löschen von Ereignismonitoren mit DROP EVENT MONITOR
- Erstellen und Ändern von externen Routinen unter Verwendung von:
  - CREATE PROCEDURE
  - CREATE FUNCTION
  - CREATE FUNCTION
  - CREATE METHOD
  - ALTER PROCEDURE
  - ALTER FUNCTION

  - ALTER METHOD
- Löschen von externen Routinen unter Verwendung von:
  - DROP PROCEDURE
  - DROP FUNCTION
  - DROP METHOD
- Starten von DB2 HADR-Operationen (High Availability Disaster Recovery) für eine Datenbank mit START HADR
- Stoppen von HADR-Operationen für eine Datenbank mit STOP HADR
- Veranlassen, dass eine HADR-Bereitschaftsdatenbank die Funktion einer HADR-Primärdatenbank übernimmt (mit TAKEOVER HADR)
- Festlegen des Konfigurationsparameters **DIAGPATH** oder **SPM_LOG_PATH** des Datenbankmanagers
- Festlegen des Datenbankkonfigurationsparameters **NEWLOGPATH**, **OVERFLOWLOGPATH**, **MIRRORLOGPATH** oder **FAILARCHPATH**
- Löschen einer Datenbankmanagerinstanz mit db2idrop

Ergebnisse

Wenn der Datenbankmanager die Änderungen an der Clusterkonfiguration für die aufgelisteten Datenbankverwaltungstasks koordiniert, müssen Sie keine separaten Cluster-Manager-Operationen ausführen.

Konfigurieren einer Clusterumgebung mit db2haicu

Mit DB2 High Availability Instance Configuration Utility (db2haicu) können Sie die Datenbanken in einer Clusterumgebung konfigurieren und verwalten. Wenn Sie db2haicu die Konfigurationsdetails für eine Datenbankmanagerinstanz mitteilen, leitet db2haicu die erforderlichen Details der Clusterkonfiguration an die Cluster-Management-Software weiter.

Vorbereitung

Vor der Verwendung von DB2 High Availability Instance Configuration Utility (db2haicu) müssen bestimmte Tasks ausgeführt werden. Weitere Informationen hierzu finden Sie in „db2haicu - Voraussetzungen" auf Seite 135.

Einschränkungen

Für die Verwendung von DB2 High Availability Instance Configuration Utility (db2haicu) gelten bestimmte Einschränkungen. Weitere Informationen hierzu finden Sie in „db2haicu - Einschränkungen" auf Seite 138.

Informationen zu dieser Task

Sie können db2haicu im interaktiven Modus ausführen oder eine XML-Eingabedatei verwenden:

Interaktiver Modus
Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) mit dem Befehl db2haicu aufrufen und keine XML-Eingabedatei mit dem Parameter **-f** angeben, wird das Dienstprogramm im interaktiven Modus ausgeführt. Im interaktiven Modus von db2haicu werden Informationen in

einem textbasierten Format angezeigt und abgefragt. Weitere Informationen hierzu finden Sie in „Ausführen von db2haicu im interaktiven Modus" auf Seite 106.

Stapelmodus mit einer XML-Eingabedatei
Sie können den Parameter `-f <name-der-eingabedatei>` mit dem Befehl db2haicu verwenden, um DB2 High Availability Instance Configuration Utility (db2haicu) mit einer XML-Eingabedatei auszuführen, in der die Konfigurationsdetails angegeben sind. Die Ausführung von db2haicu mit einer XML-Eingabedatei ist nützlich, wenn Sie bestimmte Konfigurationstasks mehrmals ausführen müssen, z. B. bei der Konfiguration mehrerer Datenbankpartitionen für hohe Verfügbarkeit. Weitere Informationen hierzu finden Sie in „Ausführen von db2haicu mit einer XML-Eingabedatei" auf Seite 107.

Vorgehensweise

Führen Sie die folgenden Schritte für jede Datenbankmanagerinstanz aus.

1. Erstellen Sie eine neue Clusterdomäne.

   Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) zum ersten Mal für eine Datenbankmanagerinstanz ausführen, erstellt db2haicu ein Modell des Clusters, das als *Clusterdomäne* bezeichnet wird. Weitere Informationen hierzu finden Sie in „Erstellen einer Clusterdomäne mit db2haicu" auf Seite 136.
2. Fahren Sie fort, die Konfiguration der Clusterdomäne weiter einzugrenzen, und führen Sie Verwaltungs- und Wartungsaufgaben für die Clusterdomäne aus.

   Wenn Sie das Clusterdomänenmodell der Clusterumgebung mit db2haicu modifizieren, gibt der Datenbankmanager die entsprechenden Änderungen an die Datenbankmanagerinstanz und die Clusterkonfiguration weiter. Weitere Informationen hierzu finden Sie in „Verwalten einer Clusterdomäne mit db2haicu" auf Seite 136.

Weitere Schritte

DB2 High Availability Instance Configuration Utility (db2haicu) verfügt über kein separates Diagnoseprotokoll. Fehler bei db2haicu können mit dem Diagnoseprotokoll `db2diag.log` des Datenbankmanagers und dem Tool db2pd untersucht und diagnostiziert werden. Weitere Informationen hierzu finden Sie in „Fehler bei db2haicu beheben" auf Seite 138.

Clusterdomäne

Eine Clusterdomäne ist ein Modell, das Informationen zu Clusterelementen wie Datenbanken, Mountpunkten und Richtlinien für Funktionsübernahme enthält. Clusterdomänen werden mit DB2 High Availability Instance Configuration Utility (db2haicu) erstellt. db2haicu (DB2 High Availability Instance Configuration Utility) verwendet die Informationen in der Clusterdomäne, um Konfigurations- und Verwaltungstasks für die Clusterverwaltung zu ermöglichen. Darüber hinaus verwendet der Datenbankmanager als Bestandteil von DB2 High Availability Feature die Informationen in der Clusterdomäne, um automatisierte Clusterverwaltungstasks auszuführen.

Wenn der Clusterdomäne ein Clusterelement hinzugefügt wird, dann wird dieses Element bei allen nachfolgenden db2haicu-Konfigurationsoperationen bzw. automatischen Operationen für die Clusterverwaltung berücksichtigt, die als Bestandteil von DB2 High Availability Feature vom Datenbankmanager ausgeführt werden. Wenn Sie ein Clusterelement aus der Clusterdomäne entfernen, wird dieses Ele-

ment bei db2haicu-Operationen bzw. automatischen Operationen für die Clusterverwaltung des Datenbankmanagers nicht mehr berücksichtigt. db2haicu und der Datenbankmanager können nur Operationen für Clusterelemente mit dem Clustermanager koordinieren, die in der von Ihnen mit db2haicu erstellten Clusterdomäne enthalten sind.

Sie können mit db2haicu folgende Clusterdomänenelemente erstellen und konfigurieren:

- Computer oder Maschinen (im Zusammenhang mit Clusterdomänen *Clusterdomänenknoten* genannt)
- Netzschnittstellenkarten (in db2haicu *Netzschnittstellen*, *Schnittstellen*, *Netzadapter* oder *Adapter* genannt)
- IP-Adressen
- Datenbanken, inklusive von Datenbankpaaren mit einer HADR-Primärdatenbank und einer HADR-Bereitschaftsdatenbank
- Datenbankpartitionen
- Mountpunkte und -pfade, inklusive von Pfaden, die im Falle einer Störung nicht für eine Funktionsübernahme relevant sind
- Richtlinien für Funktionsübernahme
- Quorumeinheiten

Cluster-Management-Software:

Cluster-Management-Software maximiert die Workload, die ein Cluster aus Computern ausführen kann. Ein Cluster-Manager verteilt die Workload möglichst gleichmäßig, um Engpässe zu vermeiden, überwacht den Status der Clusterelemente und führt eine Funktionsübernahme durch, wenn ein Element ausfallen sollte. Darüber hinaus kann ein Cluster-Manager Systemadministratoren die Ausführung von Verwaltungstasks für Elemente des Clusters erleichtern (z. B. durch das Umleiten von Verarbeitungsprozessen im Falle von Wartungsarbeiten am Computer).

Elemente eines Clusters

Der Cluster-Manager kann nur reibungslos ausgeführt werden, wenn er über die erforderlichen Angaben zu den Elementen des Clusters sowie zu den Beziehungen zwischen den Clusterelementen verfügt.

Der Cluster-Manager muss z. B. über Informationen zu folgenden Clusterelementen verfügen:

- Physische oder virtuelle Computer, Maschinen oder Einheiten im Cluster (im Zusammenhang mit Clustern *Clusterknoten* genannt)
- Die Clusterknoten verbindende Netze
- Netzschnittstellenkarten, die die Clusterknoten mit den Netzen verbinden
- IP-Adressen der Clusterknoten
- Virtuelle IP-Adressen oder Service-IP-Adressen

In Bezug auf die Beziehungen muss der Cluster-Manager z. B. über folgende Informationen verfügen:

- Clusterknotenpaare mit derselben Software, die für eine gegenseitige Funktionsübernahme verfügbar sind
- Netze mit denselben Eigenschaften, die für eine gegenseitige Funktionsübernahme verfügbar sind

- Clusterknoten, dem zurzeit eine virtuelle IP-Adresse zugeordnet ist

Hinzufügen oder Ändern von Clusterelementen

Die für die Clusterelemente und die Beziehungen zwischen diesen Elementen erforderlichen Angaben erhält der Cluster-Manager, indem die Elemente von einem Systemadministrator für den Cluster-Manager registriert werden. Nimmt ein Systemadministrator Änderungen an den Elementen eines Clusters vor, müssen diese Änderungen vom Administrator an den Cluster-Manager weitergegeben werden. Cluster-Manager verfügen über Schnittstellen, die für diese Tasks hilfreich sind.

Die Clusterverwaltung ist komplex, da es eine Vielzahl möglicher Clusterelemente gibt. Administratoren müssen über umfassende Kenntnisse in Bezug auf die Hardware und die Betriebssysteme der Clusterknoten, Netzprotokolle und -konfigurationen sowie die auf den Clusterknoten installierte Software verfügen (z. B. Datenbanksoftware). Die Registrierung der Clusterelemente mit der Cluster-Management-Software und das Aktualisieren des Cluster-Managers nach einer Systemänderung kann komplex und zeitaufwändig sein.

Hinzufügen und Ändern von Clusterelementen mit db2haicu

In einer DB2-Datenbanklösung können Sie mit DB2 High Availability Instance Configuration Utility (db2haicu) Elemente für den Cluster-Manager registrieren und den Cluster-Manager nach Verwaltungsänderungen am Cluster aktualisieren. db2haicu vereinfacht diese Tasks, da Sie kein Experte in Bezug auf die Besonderheiten der Hardware, der Betriebssysteme und der Schnittstelle des Cluster-Managers sein müssen, um diese Tasks auszuführen, sobald Sie mit dem Modell vertraut sind, das db2haicu verwendet, um die Clusterelemente und die Beziehungen zwischen diesen Elementen einzubinden.

Ressourcen und Ressourcengruppen:

Bei einer *Ressource* handelt es sich um ein Clusterelement (z. B. einen Clusterknoten, eine Datenbank, einen Mountpunkt oder eine Netzschnittstellenkarte), das mit dem Clustermanager registriert wurde. Elemente, die nicht mit dem Clustermanager registriert wurden, sind dem Clustermanager nicht bekannt und werden bei Operationen des Clustermanagers nicht berücksichtigt. Bei einer *Ressourcengruppe* handelt es sich um eine logische Sammlung von Ressourcen. Das Konstrukt der Ressourcengruppen ist sehr effizient, weil dieses Konstrukt die Definition von Beziehungen und Einschränkungen für Ressourcengruppen ermöglicht und so komplexe Verwaltungstasks für die in den Gruppen enthaltenen Ressourcen vereinfacht.

Wenn Ressourcen vom Clustermanager in Gruppen zusammengefasst werden, kann der Clustermanager diese Ressourcen anschließend gemeinsam bearbeiten. Angenommen z. B. die beiden Datenbanken `database-1` und `database-2` gehören zu der Ressourcengruppe `resource-group-A`. Führt der Clustermanager eine Startoperation für `resource-group-A` durch, werden `database-1` und `database-2` durch diese einzelne Operation des Clustermanagers gestartet.

Einschränkungen

- Ressourcengruppen können kein *Ersatzsystem* enthalten, und ein Ersatzsystem kann keine Ressourcengruppe enthalten (Ein *Ersatzsystem* besteht aus einer Gruppe von Ressourcen, die jeweils dieselbe Funktionalität bereitstellen und für eine Funktionsübernahme verwendet werden können.).
- Ressourcen können nur jeweils einer Ressourcengruppe angehören.

- Ressourcen können nicht in einer Ressourcengruppe und einem Ersatzsystem enthalten sein.
- Ressourcengruppen können weitere Ressourcengruppen enthalten. Maximale Verschachtelungsebene: 50.
- Es können bis zu 100 Ressourcen zu einer Ressourcengruppe zusammengefasst werden.

Quorumeinheiten:

Eine *Quorumeinheit* erleichtert es dem Cluster-Manager Entscheidungen in Bezug auf die Clusterverwaltung zu treffen, wenn der Standardentscheidungsprozess des Cluster-Managers nicht zu einem eindeutigen Ergebnis führt. Wenn ein Cluster-Manager zwischen mehreren möglichen Aktionen wählen muss, zählt der Cluster-Manager die Clusterdomänenknoten, die die einzelnen möglichen Aktionen unterstützen, und wählt anschließend die Aktion aus, die von der Mehrheit der Clusterdomänenknoten unterstützt wird. Werden mehrere Aktionen von derselben Anzahl von Clusterdomänenknoten unterstützt, greift der Cluster-Manager bei seiner Entscheidung auf eine Quorumeinheit zurück.

db2haicu unterstützt die in der folgenden Tabelle aufgeführten Quorumeinheiten.

Tabelle 2. Von db2haicu unterstützte Typen von Quorumeinheiten

Quorumeinheit	Beschreibung
network	Eine Netzquorumeinheit des Typs 'network' ist eine IP-Adresse, zu der jeder einzelne Clusterdomänenknoten zu jeder beliebigen Zeit eine Verbindung herstellen kann.

Netze in einer Clusterdomäne:

Für die Konfiguration netzspezifischer Clusterdomänenelemente können Sie mit DB2 High Availability Instance Configuration Utility (db2haicu) ein *physisches Netz* zur Clusterdomäne hinzufügen. Ein physisches Netz besteht aus Netzschnittstellenkarten, IP-Adressen und Teilnetzmasken.

Netzschnittstellenkarten

Eine *Netzschnittstellenkarte* (NIC, Network Interface Card) ist eine Hardwarekomponente, die einen Computer (auch *Clusterknoten* genannt) mit einem Netz verbindet. Netzschnittstellenkarten werden teilweise auch kurz *Schnittstellen*, *Netzadapter* oder *Adapter* genannt. Wenn Sie mit db2haicu ein physisches Netz zu Ihrer Clusterdomäne hinzufügen, müssen Sie mindestens eine Netzschnittstellenkarte sowie den Hostnamen des Computers, zu dem die Netzschnittstellenkarte gehört, den Namen der Netzschnittstellenkarte auf dem Hostcomputer sowie die IP-Adresse der Netzschnittstellenkarte angeben.

IP-Adressen

Eine *IP-Adresse* (IP, Internet Protocol) ist eine innerhalb eines Netzes eindeutige Adresse. Bei IP Version 4 besteht eine IP-Adresse aus 32 Bit und wird normalerweise in Dezimalschreibweise mit einem Punkt als Trennzeichen angegeben (z. B. `129.30.180.16`). Eine IP-Adresse setzt sich aus einem netzspezifischen Teil und einem hostspezifischen Teil zusammen.

db2haicu unterstützt IP-Version 6 nicht.

Teilnetzmasken

Ein Netz kann mithilfe von *Teilnetzmasken* in mehrere logische Teilnetze unterteilt werden. Eine Teilnetzmaske ermöglicht es, einige Bits des hostspezifischen Teils der IP-Adresse auf den netzspezifischen Teil zu verschieben. Wenn Sie mit db2haicu eine IP-Adresse in Ihrer Clusterdomäne hinzufügen, müssen Sie in manchen Fällen die Teilnetzmaske für die IP-Adresse angeben. Sie müssen z. B. die Teilnetzmaske der IP-Adresse der jeweiligen Netzschnittstellenkarte angeben, wenn Sie eine Netzschnittstellenkarte mit db2haicu hinzufügen.

Netzersatzsysteme

Ein *Ersatzsystem* besteht aus einer Gruppe von Ressourcen, die jeweils dieselbe Funktionalität bereitstellen und für eine Funktionsübernahme verwendet werden können. Wenn Sie ein Netz mit db2haicu erstellen, können die Netzschnittstellenkarten in diesem Netz für eine gegenseitige Funktionsübernahme verwendet werden. Ein derartiges Netz wird auch als *Netzersatzsystem* bezeichnet.

Netzprotokolle

Wenn Sie mit db2haicu ein Netz zu Ihrer Clusterdomäne hinzufügen, müssen Sie den Typ des verwendeten Netzprotokolls angeben. Zurzeit wird nur das Netzprotokoll TCP/IP unterstützt.

Richtlinien für Funktionsübernahme in Clusterdomänen:

Eine *Richtlinie für Funktionsübernahme* gibt an, wie ein Cluster-Manager sich verhalten soll, wenn ein Clusterelement, wie z. B. eine Netzschnittstellenkarte oder ein Datenbankserver, ausfällt. Im Allgemeinen wird ein Cluster-Manager die Workload in diesem Fall von dem ausgefallenen Element auf ein alternatives Element übertragen, das zuvor für den Cluster-Manager als geeigneter Ersatz für das ausgefallene Element definiert wurde. Diese Übertragung der Workload von einem ausgefallenen Element auf ein anderes Element wird als *Funktionsübernahme* bezeichnet.

Funktionsübernahmerichtlinie 'Umlauf'

Die *Funktionsübernahmerichtlinie 'Umlauf'* (RoundRobin) sieht vor, dass der Datenbankmanager bei einem Ausfall eines einzigen Computers (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne die Workload des ausgefallenen Clusterdomänenknotens auf einem beliebigen anderen Knoten in der Clusterdomäne erneut startet.

Funktionsübernahmerichtlinie 'Gegenseitige Übernahme'

Für die Konfiguration der *Funktionsübernahmerichtlinie 'Gegenseitige Übernahme'* (Mutual) müssen Sie zwei Computer (auch *Clusterdomänenknoten* oder einfach *Knoten* genannt) in der Clusterdomäne als ein Systempaar zuordnen. Tritt bei einem dieser beiden Knoten ein Ausfall auf, wird die Workload der Datenbankpartitionen des ausgefallenen Knotens von dem anderen Knoten übernommen. Eine gegenseitige Übernahme ist nur möglich, wenn Sie über mehrere Datenbankpartitionen verfügen.

Funktionsübernahmerichtlinie 'M+N'

Wenn Sie die *Funktionsübernahmerichtlinie 'M+N'* (NPlusM) verwenden, wird bei einem Ausfall eines Computers (auch *Clusterdomänenknoten* oder kurz *Knoten* ge-

nannt) in der Clusterdomäne die Workload der Datenbankpartitionen auf dem ausgefallenen Knoten von einem beliebigen anderen Knoten in der Clusterdomäne übernommen. Eine M+N-Übernahme ist nur möglich, wenn Sie über mehrere Datenbankpartitionen verfügen.

Funktionsübernahmerichtlinie 'Lokaler Neustart'

Wenn Sie die *Funktionsübernahmerichtlinie 'Lokaler Neustart'* (LocalRestart) verwenden, startet der Datenbankmanager bei einem Ausfall auf einem Computer (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne die (lokal) vorhandene Datenbank auf dem Knoten, der ausgefallen ist, neu.

Funktionsübernahmerichtlinie 'HADR'

Wenn Sie die *Funktionsübernahmerichtlinie 'HADR'* (HADRFailover) konfigurieren, ermöglichen Sie der DB2 HADR-Funktion (High Availability Disaster Recovery) die Verwaltung der Funktionsübernahme. Fällt eine HADR-Primärdatenbank aus, versetzt der Datenbankmanager die Workload von der ausgefallenen Datenbank auf eine HADR-Bereitschaftsdatenbank.

Funktionsübernahmerichtlinie 'Benutzerdefiniert'

Wenn Sie die *Funktionsübernahmerichtlinie 'Benutzerdefiniert'* (Custom) konfigurieren, erstellen Sie eine Liste von Computern (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne, die der Datenbankmanager für eine Funktionsübernahme nutzen kann. Fällt ein Knoten in der Clusterdomäne aus, versetzt der Datenbankmanager die Workload vom ausgefallenen Knoten auf einen in der von Ihnen angegebenen Liste enthaltenen Knoten.

Mountpunkte in einer Clusterdomäne:

Nach dem Anhängen eines Dateisystems können Sie den jeweiligen Mountpunkt mit DB2 High Availability Instance Configuration Utility (db2haicu) in der Clusterdomäne hinzufügen.

Mountpunkte

Bei UNIX-, Linux- und AIX-Betriebssystemen werden Dateisysteme über einen *Mountvorgang* angehängt, um sie dem Betriebssystem zur Verfügung zu stellen. Während des Mountvorgangs führt das Betriebssystems Tasks wie das Lesen der Index- oder Navigationsdatenstrukturen durch und ordnet dem angehängten Dateisystem einen Verzeichnispfad zu. Dieser zugeordnete Verzeichnispfad, über den Sie auf das angehängte Dateisystem zugreifen können, wird als *Mountpunkt* bezeichnet.

Nicht kritische Mountpunkte oder -pfade

Der Cluster enthält möglicherweise Mountpunkte oder -pfade, für die im Falle einer Störung keine Funktionsübernahme erforderlich ist. Mit db2haicu können Sie eine Liste dieser Mountpunkte oder - pfade zu Ihrer Clusterdomäne hinzufügen. Mountpunkte oder -pfade in dieser Liste werden vom Cluster-Manager nicht in Funktionsübernahmen einbezogen.

Angenommen z. B. ein Festplattenlaufwerk ist über `/mnt/driveA` an den Computer `node1` im Cluster angehängt. Wenn Sie entscheiden, dass `/mnt/driveA` von kritischer Bedeutung ist und verfügbar sein muss, führt der Cluster-Manager im Falle

einer Störung eine Funktionsübernahme durch, damit /mnt/driveA verfügbar bleibt, wenn node1 ausfällt. Wenn Sie jedoch entscheiden, dass /mnt/driveA nicht unbedingt zur Verfügung stehen muss, wenn node1 ausfällt, können Sie dem Cluster-Manager angeben, dass /mnt/driveA nicht von kritischer Bedeutung ist, indem Sie /mnt/driveA in die Liste der nicht kritischen Pfade aufnehmen. Ist /mnt/driveA in Bezug auf Funktionsübernahmen als nicht kritisch eingestuft, führt dies dazu, dass das betreffende Laufwerk möglicherweise nicht verfügbar ist, wenn node1 ausfällt.

db2haicu

DB2 High Availability Instance Configuration Utility (db2haicu) ist ein textbasiertes Dienstprogramm zur Konfiguration und Verwaltung hoch verfügbarer Datenbanken in einer Clusterumgebung. db2haicu stellt Abfragen an das System, um Informationen zur Datenbankinstanz, zur Clusterumgebung und zum Cluster-Manager zu sammeln. Zur Angabe weiterer Informationen über Parameter für den Aufruf db2haicu, eine Eingabedatei oder während der Laufzeit können Sie die Eingabeaufforderungen von db2haicu verwenden.

Syntax

```
db2haicu [ -f <name-der-XML-eingabedatei> ]
         [ -disable ]
         [ -delete [ dbpartitionnum <liste-der-datenbankpartitionen> |
                     hadrdb <datenbankname> ] ]
```

Parameter

Bei der Übergabe von Parametern an den Befehl db2haicu muss die Groß-/Kleinschreibung beachtet werden; die Eingabe muss in Kleinbuchstaben erfolgen.

-f <*name-der-XML-eingabedatei***>**
Unter Verwendung des Parameters **-f** können Sie Details zur Clusterdomäne in der XML-Eingabedatei <*name-der-XML-eingabedatei*> angeben.Weitere Informationen hierzu finden Sie in „Ausführen von db2haicu mit einer XML-Eingabedatei" auf Seite 107.

-disable
Eine Datenbankmanagerinstanz ist *für hohe Verfügbarkeit konfiguriert*, sobald Sie mithilfe von db2haicu eine Clusterdomäne für diese Instanz erstellt haben. Wenn eine Datenbankmanagerinstanz für hohe Verfügbarkeit konfiguriert wurde und bestimmte Verwaltungsoperationen des Datenbankmanagers entsprechende Änderungen an der Clusterkonfiguration erforderlich machen, teilt der Datenbankmanager dem Cluster-Manager diese Änderungen mit. Wenn der Datenbankmanager diese Clusterverwaltungstasks mit dem Cluster-Manager für Sie koordiniert, müssen Sie keine separate Cluster-Manager-Operation für diese Verwaltungstasks ausführen. Diese Integration zwischen Datenbankmanager und Cluster-Manager ist eine Funktion von DB2 High Availability Feature.

Mit dem Parameter **-disable** können Sie angeben, dass eine Datenbankmanagerinstanz nicht mehr für hohe Verfügbarkeit konfiguriert sein soll. Wenn die Datenbankmanagerinstanz nicht mehr für hohe Verfügbarkeit konfiguriert ist, nimmt der Datenbankmanager keine Koordination mit dem Cluster-Manager vor, wenn die Ausführung von Verwaltungsoperationen des Datenbankmanagers entsprechende Änderungen an der Clusterkonfiguration erforderlich machen.

Führen Sie db2haicu erneut aus, um eine Datenbankmanagerinstanz für hohe Verfügbarkeit zu rekonfigurieren.

-delete

Mit dem Parameter **-delete** können Sie Ressourcengruppen für die aktuelle Datenbankmanagerinstanz löschen.

Wenn Sie weder den Parameter **dbpartitionnum** noch den Parameter **hadrdb** verwenden, entfernt db2haicu alle Ressourcengruppen, die der aktuellen Datenbankmanagerinstanz zugeordnet sind.

dbpartitionnum <*liste-der-datenbankpartitionen*>
Mit dem Parameter **dbpartitionnum** können Sie Ressourcengruppen löschen, die den in <*liste-der-datenbankpartitionen*> aufgelisteten Datenbankpartitionen zugeordnet sind.
<*liste-der-datenbankpartitionen*> ist eine durch Kommas getrennte Liste mit Datenbankpartitionsnummern.

hadrdb <*datenbankname*>
Mit dem Parameter **hadrdb** können Sie Ressourcengruppen löschen, die der DB2 HADR-Datenbank <*datenbankname*> zugeordnet sind.

Wenn in der Clusterdomäne keine Ressourcengruppen mehr vorhanden sind, nachdem db2haicu die Ressourcengruppen entfernt hat, entfernt db2haicu auch die Clusterdomäne.

Die Ausführung von db2haicu mit dem Parameter **-delete** hat zur Folge, dass die aktuelle Datenbankmanagerinstanz nicht mehr für hohe Verfügbarkeit konfiguriert ist. Wenn die Datenbankmanagerinstanz nicht mehr für hohe Verfügbarkeit konfiguriert ist, nimmt der Datenbankmanager keine Koordination mit dem Cluster-Manager vor, wenn die Ausführung von Verwaltungsoperationen des Datenbankmanagers entsprechende Änderungen an der Clusterkonfiguration erforderlich machen.

Führen Sie db2haicu erneut aus, um eine Datenbankmanagerinstanz für hohe Verfügbarkeit zu rekonfigurieren.

db2haicu - Startmodus:

Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) zum ersten Mal für eine Datenbankmanagerinstanz ausführen, wird db2haicu im Startmodus ausgeführt.

Bei der Ausführung von db2haicu überprüft db2haicu die Datenbankmanagerinstanz und die Systemkonfiguration und sucht nach einer vorhandenen *Clusterdomäne*. Eine Clusterdomäne ist ein Modell, das Informationen zu Clusterelementen wie Datenbanken, Mountpunkten und Richtlinien für Funktionsübernahme enthält. Clusterdomänen werden mit DB2 High Availability Instance Configuration Utility (db2haicu) erstellt. db2haicu (DB2 High Availability Instance Configuration Utility) verwendet die Informationen in der Clusterdomäne, um Konfigurations- und Verwaltungstasks für die Clusterverwaltung zu ermöglichen. Darüber hinaus verwendet der Datenbankmanager als Bestandteil von DB2 High Availability Feature die Informationen in der Clusterdomäne, um automatisierte Clusterverwaltungstasks auszuführen.

Wenn Sie db2haicu für eine Datenbankmanagerinstanz ausführen und für diese Instanz noch keine Clusterdomäne erstellt und konfiguriert wurde, beginnt db2haicu umgehend mit der Erstellung und Konfiguration einer neuen Clusterdomäne. Bei der Erstellung einer neuen Clusterdomäne werden Sie von db2haicu zur Eingabe von Informationen wie beispielsweise einem Namen für die neue Clusterdomäne oder dem Hostnamen der aktuellen Maschine aufgefordert.

Wenn Sie eine Clusterdomäne erstellen, diese jedoch nicht vollständig konfigurieren, setzt db2haicu bei der nächsten Ausführung die Konfiguration der Clusterdomäne fort.

Nachdem Sie eine Clusterdomäne für eine Datenbankmanagerinstanz erstellt und konfiguriert haben, wird db2haicu im Wartungsmodus ausgeführt.

db2haicu - Wartungsmodus:

Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) ausführen und für die aktuelle Datenbankmanagerinstanz bereits eine Clusterdomäne erstellt wurde, wird db2haicu im Wartungsmodus ausgeführt.

Wenn db2haicu im Wartungsmodus ausgeführt wird, zeigt db2haicu eine Liste der Konfigurations- und Verwaltungstasks an, die Sie ausführen können.

Zu den Wartungsaufgaben von db2haicu gehört das Hinzufügen von Datenbank- oder Clusterelementen wie beispielsweise Datenbanken oder Clusterknoten zur Clusterdomäne sowie das Entfernen von Elementen aus der Clusterdomäne. Ferner gehört zu den Wartungsaufgaben von db2haicu das Modifizieren der Details von Clusterdomänenelementen wie der Richtlinie für Funktionsübernahme für die Datenbankmanagerinstanz.

Bei der Ausführung von db2haicu im Wartungsmodus zeigt db2haicu eine Liste der Operationen an, die Sie für die Clusterdomäne ausführen können:

- Fügen Sie Clusterknoten hinzu, oder entfernen Sie sie. (Die Maschine wird durch den Hostnamen identifiziert.)
- Fügen Sie eine Netzschnittstelle (Netzschnittstellenkarte) hinzu, oder entfernen Sie sie.
- Fügen Sie DB2-Datenbankpartitionen hinzu, oder entfernen Sie sie. (Nur in Umgebungen mit partitionierten Datenbanken.)
- Fügen Sie HADR-Datenbanken hinzu, oder entfernen Sie sie.
- Fügen Sie eine Hochverfügbarkeitsdatenbank hinzu, oder entfernen Sie sie.
- Fügen Sie einen Mountpunkt hinzu, oder entfernen Sie ihn.
- Fügen Sie eine IP-Adresse hinzu, oder entfernen Sie sie.
- Fügen Sie einen nicht kritischen Pfad hinzu, oder entfernen Sie ihn.
- Versetzen Sie DB2-Datenbankpartitionen und HADR-Datenbanken für die planmäßige Wartung.
- Ändern Sie die Richtlinie für Funktionsübernahme für diese Instanz.
- Erstellen Sie eine neue Quorumeinheit für die Domäne.
- Löschen Sie die Domäne.

Ausführen von db2haicu im interaktiven Modus:

Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) mit dem Befehl db2haicu aufrufen und keine XML-Eingabedatei mit dem Parameter **-f** angeben, wird das Dienstprogramm im interaktiven Modus ausgeführt. Im interaktiven Modus von db2haicu werden Informationen in einem textbasierten Format angezeigt und abgefragt.

Vorbereitung

Vor der Verwendung von DB2 High Availability Instance Configuration Utility (db2haicu) müssen bestimmte Tasks ausgeführt werden. Weitere Informationen hierzu finden Sie in „db2haicu - Voraussetzungen" auf Seite 135.

Informationen zu dieser Task

Bei der Ausführung von db2haicu im interaktiven Modus werden Informationen und Fragen im Textformat auf dem Bildschirm angezeigt. Sie können die von db2haicu angeforderten Informationen in einer Eingabeaufforderung eingeben, die sich am unteren Rand des Bildschirms befindet.

Vorgehensweise

Rufen Sie zur Ausführung von db2haicu im interaktiven Modus den Befehl db2haicu ohne den Parameter `-f <name-der-eingabedatei>` auf.

Weitere Schritte

DB2 High Availability Instance Configuration Utility (db2haicu) verfügt über kein separates Diagnoseprotokoll. Fehler bei db2haicu können mit dem Diagnoseprotokoll `db2diag.log` des Datenbankmanagers und dem Tool db2pd untersucht und diagnostiziert werden. Weitere Informationen hierzu finden Sie in „Fehler bei db2haicu beheben" auf Seite 138.

Ausführen von db2haicu mit einer XML-Eingabedatei:

Sie können den Parameter `-f <name-der-eingabedatei>` mit dem Befehl db2haicu verwenden, um DB2 High Availability Instance Configuration Utility (db2haicu) mit einer XML-Eingabedatei auszuführen, in der die Konfigurationsdetails angegeben sind. Die Ausführung von db2haicu mit einer XML-Eingabedatei ist nützlich, wenn Sie bestimmte Konfigurationstasks mehrmals ausführen müssen, z. B. bei der Konfiguration mehrerer Datenbankpartitionen für hohe Verfügbarkeit.

Vorbereitung

Vor der Verwendung von DB2 High Availability Instance Configuration Utility (db2haicu) müssen bestimmte Tasks ausgeführt werden. Weitere Informationen hierzu finden Sie in „db2haicu - Voraussetzungen" auf Seite 135.

Informationen zu dieser Task

Im Verzeichnis `sqllib` befindet sich im Unterverzeichnis `samples` eine Gruppe von XML-Beispieleingabedateien, die Sie modifizieren und mit db2haicu zum Konfigurieren der Clusterumgebung verwenden können. Weitere Informationen hierzu finden Sie in „XML-Beispieleingabedateien für db2haicu" auf Seite 126.

Vorgehensweise

1. Erstellen Sie eine XML-Eingabedatei.
2. Rufen Sie db2haicu mit dem Parameter `-f <name-der-eingabedatei>` auf.

   Verwenden Sie den folgenden Befehl, um die Clusterumgebung mit db2haicu und der erstellten Eingabedatei `db2haicu-input.xml` für die aktuelle Datenbankmanagerinstanz zu konfigurieren:

   ```
   db2haicu -f db2haicu-input.xml
   ```

Weitere Schritte

DB2 High Availability Instance Configuration Utility (db2haicu) verfügt über kein separates Diagnoseprotokoll. Fehler bei db2haicu können mit dem Diagnoseprotokoll db2diag.log des Datenbankmanagers und dem Tool db2pd untersucht und diagnostiziert werden. Weitere Informationen hierzu finden Sie in „Fehler bei db2haicu beheben" auf Seite 138.

db2haicu - XML-Schemadefinition für Eingabedateien:

Die DB2 High Availability Instance Configuration Utility (db2haicu)-XML-Schemadefinition (XSD) für Eingabedateien definiert die Clusterdomänenobjekte, die Sie in einer db2haicu-XML-Eingabedatei angeben können. Diese db2haicu-XSD befindet sich in der Datei db2ha.xsd im Verzeichnis sqllib/samples/ha/xml.

DB2ClusterType

Das Stammelement der db2haicu-XSD ist das Element DB2Cluster, das den Typ DB2ClusterType aufweist. Eine db2haicu-XML-Eingabedatei muss mit dem Element DB2Cluster beginnen.

„XML-Schemadefinition "
„Subelemente"
„Attribute" auf Seite 109
„Hinweise zur Verwendung" auf Seite 110

XML-Schemadefinition

```
<xs:complexType name='DB2ClusterType'>
 <xs:sequence>
   <xs:element name='DB2ClusterTemplate' type='DB2ClusterTemplateType' minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='ClusterDomain'      type='ClusterDomainType'                    maxOccurs='unbounded'/>
   <xs:element name='FailoverPolicy'     type='FailoverPolicyType'     minOccurs='0'                        />
   <xs:element name='DB2PartitionSet'    type='DB2PartitionSetType'    minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='HADRDBSet'          type='HADRDBType'             minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='HADBSet'            type='HADBType'               minOccurs='0' maxOccurs='unbounded'/>
 </xs:sequence>
 <xs:attribute name='clusterManagerName' type='xs:string' use='optional'/>
</xs:complexType>
```

Subelemente

DB2ClusterTemplate

Typ: DB2ClusterTemplateType

Hinweise zur Verwendung:
Fügen Sie das Element DB2ClusterTemplateType nicht in der db2haicu-XML-Eingabedatei ein. Das Element DB2ClusterTemplateType ist für neuere Releases reserviert.

ClusterDomain

Typ: ClusterDomainType

Das Element ClusterDomainType enthält Spezifikationen zu den Maschinen oder Computern (auch *Clusterdomänenknoten* genannt) in der Clusterdomäne, zu den *Netzersatzsystemen* (Gruppen von Netzen mit gegenseitiger Funktionsübernahme) und zur *Quorumeinheit* (Zuteilungsroutinenmechanismus).

Regeln zur Häufigkeit:
Sie müssen mindestens ein Element ClusterDomain in das Element DB2ClusterType einfügen.

FailoverPolicy

Typ: `FailoverPolicyType`

Das Element `FailoverPolicyType` gibt die *Richtlinie für Funktionsübernahme* an, die der Cluster-Manager in der Clusterdomäne anwenden soll.

Regeln zur Häufigkeit:
Sie können kein oder ein Element `FailoverPolicy` in das Element `DB2ClusterType` einfügen.

DB2PartitionSet

Typ: `DB2PartitionSetType`

Das Element `DB2PartitionSetType` enthält Informationen zu Datenbankpartitionen. Eine Verwendung des Elements `DB2PartitionSetType` ist nur in Umgebungen mit partitionierten Datenbanken möglich.

Regeln zur Häufigkeit:
Sie können entsprechend der db2haicu-XML-Schemadefinition kein oder beliebig viele Elemente `DB2PartitionSet` in das Element `DB2ClusterType` einfügen.

HADRDBSet

Typ: `HADRDBType`

Das Element `HADRDBType` enthält eine Liste mit Datenbankpaaren, die aus einer HADR-Primärdatenbank (High Availability Disaster Recovery) und einer HADR-Bereitschaftsdatenbank bestehen.

Regeln zur Häufigkeit:
Sie können entsprechend der db2haicu-XML-Schemadefinition kein oder beliebig viele Elemente `HADRDBSet` in das Element `DB2ClusterType` einfügen.

Hinweise zur Verwendung:

- Sie dürfen das Element `HADRDBSet` nicht in einer Umgebung mit partitionierten Datenbanken einfügen.
- Wenn Sie das Element `HADRDBSet` einfügen, müssen Sie als Richtlinie für Übernahme `HADRFailover` im Element `FailoverPolicy` angeben.

HADBSet

Typ: `HADBType`

Das Element `HADBType` enthält eine Liste der Datenbanken, die zu der Clusterdomäne gehören und eine hohe Verfügbarkeit aufweisen sollen.

Regeln zur Häufigkeit:
Sie können entsprechend der db2haicu-XML-Schemadefinition kein oder beliebig viele Elemente `HADBSet` in das Element `DB2ClusterType` einfügen.

Attribute

clusterManagerName (optional)
Das Attribut `clusterManagerName` gibt den Cluster-Manager an.

Gültige Werte für dieses Attribut enthält die folgende Tabelle:

Tabelle 3. Gültige Werte für das Attribut `clusterManager`

Wert für `clusterManagerName`	Cluster-Managerprodukt
TSA	IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component

Hinweise zur Verwendung

In einer Umgebung mit Einzelpartitionsdatenbank wird in der Regel nur eine einzige Clusterdomäne pro Datenbankmanagerinstanz erstellt.

Für eine Umgebung mit partitionierten Datenbanken ist z. B. die folgende Konfiguration möglich:

- Definieren Sie das Element `FailoverPolicy` mit `Mutual`.
- Verwenden Sie im Subelement `DB2Partition` von `DB2PartitionSet` das Element `MutualPair`, um zwei Clusterdomänenknoten anzugeben, die in einer einzigen Clusterdomäne enthalten sind.

ClusterDomainType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `ClusterDomainType` enthält Spezifikationen zu den Maschinen oder Computern (auch *Clusterdomänenknoten* genannt) in der Clusterdomäne, zu den *Netzersatzsystemen* (Gruppen von Netzen mit gegenseitiger Funktionsübernahme) und zur *Quorumeinheit* (Zuteilungsroutinenmechanismus).

„Superelemente"
„XML-Schemadefinition "
„Subelemente"
„Attribute" auf Seite 111

Superelemente

Die folgenden Elementtypen enthalten `ClusterDomainType`-Subelemente:

- `DB2ClusterType`

XML-Schemadefinition

```
<xs:complexType name='ClusterDomainType'>
 <xs:sequence>
   <xs:element name='Quorum'          type='QuorumType'          minOccurs='0'                    />
   <xs:element name='PhysicalNetwork' type='PhysicalNetworkType' minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='ClusterNode'     type='ClusterNodeType'                   maxOccurs='unbounded'/>
 </xs:sequence>
 <xs:attribute name='domainName'      type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Quorum

Typ: `QuorumType`

Das Element `QuorumType` gibt die *Quorumeinheit* für die Clusterdomäne an.

Regeln zur Häufigkeit:
Sie können kein oder ein Element `Quorum` in das Element `ClusterDomainType` einfügen.

PhysicalNetwork

Typ: `PhysicalNetworkType`

Das Element `PhysicalNetworkType` enthält Netzschnittstellenkarten, die für eine gegenseitige Funktionsübernahme verfügbar sind. Dies wird auch als *Netzersatzsystem* bezeichnet.

Regeln zur Häufigkeit:
Sie können kein oder beliebig viele Elemente `PhysicalNetwork` in das Element `ClusterDomainType` einfügen.

ClusterNode

Typ: `ClusterNodeType`

Das Element `ClusterNodeType` enthält Informationen zu einem bestimmten Computer oder einer bestimmten Maschine (auch *Clusterdomänenknoten* genannt) des Clusters.

Regeln zur Häufigkeit:
Sie müssen mindestens ein Element `ClusterNode` im Element `ClusterDomainType` angeben.

Hinweise zur Verwendung
IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component unterstützt bis zu 32 Clusterdomänenknoten. Wenn es sich beim Clustermanager um SA MP Base Component handelt, können Sie bis zu 32 Elemente `ClusterNode` in das Element `ClusterDomainType` einfügen.

Attribute

`domainName` (erforderlich)

Geben Sie einen eindeutigen Namen für das Element `ClusterDomainType` an.

Bei der Verwendung von RSCT (Reliable Scalable Cluster Technology) zur Clusterverwaltung gelten für 'domainName' die folgenden Einschränkungen:

- `domainName` darf nur die Zeichen A bis Z, a bis z, die Ziffern 0 bis 9, Punkt (.) und Unterstreichungszeichen (_) enthalten.
- `domainName` darf nicht "IW" lauten.

QuorumType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `QuorumType` gibt die *Quorumeinheit* für die Clusterdomäne an.

„Superelemente"
„XML-Schemadefinition " auf Seite 112
„Subelemente" auf Seite 112
„Attribute" auf Seite 112

Superelemente

Die folgenden Elementtypen enthalten `QuorumType`-Subelement:

- `ClusterDomainType`

XML-Schemadefinition

```
<xs:complexType name='QuorumType'>
  <xs:attribute name='quorumDeviceProtocol' type='QuorumDeviceProtocolType' use='required'/>
  <xs:attribute name='quorumDeviceName'     type='xs:string'                use='required'/>
</xs:complexType>
```

Subelemente

Keine.

Attribute

quorumDeviceProtocol (erforderlich)

`quorumDeviceProtocol` gibt den Typ des zu verwendenden Quorums an.

Eine *Quorumeinheit* erleichtert es dem Cluster-Manager Entscheidungen in Bezug auf die Clusterverwaltung zu treffen, wenn der Standardentscheidungsprozess des Cluster-Managers nicht zu einem eindeutigen Ergebnis führt. Wenn ein Cluster-Manager zwischen mehreren möglichen Aktionen wählen muss, zählt der Cluster-Manager die Clusterdomänenknoten, die die einzelnen möglichen Aktionen unterstützen, und wählt anschließend die Aktion aus, die von der Mehrheit der Clusterdomänenknoten unterstützt wird. Werden mehrere Aktionen von derselben Anzahl von Clusterdomänenknoten unterstützt, greift der Cluster-Manager bei seiner Entscheidung auf eine Quorumeinheit zurück.

Der Typ des Attributs `quorumDeviceProtocol` ist `QuorumDeviceProtocolType`.

Die XML-Schemadefinition für das Element `QuorumDeviceProtocolType` sieht wie folgt aus:

```
<xs:simpleType name='QuorumDeviceProtocolType'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='disk'/>
    <xs:enumeration value='scsi'/>
    <xs:enumeration value='network'/>
    <xs:enumeration value='eckd'/>
    <xs:enumeration value='mns'/>
  </xs:restriction>
</xs:simpleType>
```

Zurzeit werden die in der folgenden Tabelle enthaltenen Werte für dieses Attribut unterstützt:

Tabelle 4. Gültige Werte für das Attribut `quorumDeviceProtocol`

Wert für **quorumDeviceProtocol**	Bedeutung
network	Eine Netzquorumeinheit des Typs 'network' ist eine IP-Adresse, zu der jeder einzelne Clusterdomänenknoten zu jeder beliebigen Zeit eine Verbindung herstellen kann.

quorumDeviceName (erforderlich)

Der Wert für `quorumDeviceName` richtet sich nach dem in `quorumDeviceProtocol` angegebenen Quorumeinheitentyp.

Gültige Werte für dieses Attribut enthält die folgende Tabelle:

Tabelle 5. Gültige Werte für das Attribut quorumDeviceName

Wert für **quorumDeviceProtocol**	Gültige Werte für **quorumDeviceName**
network	Eine Zeichenfolge mit einer korrekt formatierten IP-Adresse. Beispiel: `12.126.4.5` Die angegebene IP-Adresse ist nur für eine Netzquorumeinheit gültig, wenn jeder einzelne Clusterdomänenknoten auf die betreffende IP-Adresse zugreifen kann (z. B. mithilfe des Dienstprogramms `ping`).

PhysicalNetworkType - XML-Schemadefinition für Eingabedateien für DB2 High Availability Instance Configuration Utility (db2haicu):

Das Element `PhysicalNetworkType` enthält Netzschnittstellenkarten, die für eine gegenseitige Funktionsübernahme verfügbar sind. Dies wird auch als *Netzersatzsystem* bezeichnet.

„Superelemente"
„XML-Schemadefinition "
„Subelemente "
„Attribute" auf Seite 114

Superelemente

Die folgenden Elementtypen enthalten `PhysicalNetworkType`-Subelemente:

- `ClusterDomainType`

XML-Schemadefinition

```
<xs:complexType name='PhysicalNetworkType'>
  <xs:sequence>
    <xs:element name='Interface'              type='InterfaceType' minOccurs='1' maxOccurs='unbounded'/>
    <xs:element name='LogicalSubnet'          type='IPAddressType' minOccurs='0' maxOccurs='unbounded'/>
  </xs:sequence>
  <xs:attribute name='physicalNetworkName'     type='xs:string'                   use='required'/>
  <xs:attribute name='physicalNetworkProtocol' type='PhysicalNetworkProtocolType' use='required'/>
</xs:complexType>
```

Subelemente

Interface

Typ: `InterfaceType`

Das Element `InterfaceType` besteht aus einer IP-Adresse, dem Namen eines Computers oder einer Maschine im Netz (auch *Clusterdomänenknoten* genannt) und dem Namen einer *Netzschnittstellenkarte* (NIC) des Clusterdomänenknotens.

Regeln zur Häufigkeit:
Sie müssen mindestens ein Element `Interface` im Element `PhysicalNetworkType` angeben.

LogicalSubnet

Typ: `IPAddressType`

Das Element `IPAddressType` enthält alle Einzelangaben zu einer IP-Adresse, z. B. die *Basisadresse*, die *Teilnetzmaske* und den Namen des Netzes, zu dem die IP-Adresse gehört.

Regeln zur Häufigkeit:
Sie können kein oder beliebig viele Elemente `LogicalSubnet` im Element `PhysicalNetworkType` angeben.

Attribute

physicalNetworkName (erforderlich)
Geben Sie einen eindeutigen Namen des zugehörigen physischen Netzes für `physicalNetworkName` für jedes einzelne Element `PhysicalNetworkType` an.

physicalNetworkProtocol (erforderlich)
Der Typ des Attributs `physicalNetworkProtocol` ist `PhysicalNetworkProtocolType`.

Die XML-Schemadefinition für das Element `PhysicalNetworkProtocolType` sieht wie folgt aus:

```
<xs:simpleType name='PhysicalNetworkProtocolType'>
  <xs:restriction base='xs:string'>
    <xs:enumeration value='ip'/>
    <xs:enumeration value='rs232'/>
    <xs:enumeration value='scsi'/>
    <xs:enumeration value='ssa'/>
    <xs:enumeration value='disk'/>
  </xs:restriction>
</xs:simpleType>
```

Zurzeit werden die in der folgenden Tabelle enthaltenen Werte für dieses Attribut unterstützt:

Tabelle 6. Gültige Werte für das Attribut `physicalNetworkProtocol`

Wert für `physicalNetworkProtocol`	Bedeutung
ip	TCP/IP-Protokoll

InterfaceType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `InterfaceType` besteht aus einer IP-Adresse, dem Namen eines Computers oder einer Maschine im Netz (auch *Clusterdomänenknoten* genannt) und dem Namen einer *Netzschnittstellenkarte* (NIC) des Clusterdomänenknotens.

„Superelemente"
„XML-Schemadefinition "
„Subelemente " auf Seite 115
„Attribute" auf Seite 115

Superelemente

Die folgenden Elementtypen enthalten `InterfaceType`-Subelemente:

- `PhysicalNetworkType`

XML-Schemadefinition

```
<xs:complexType name='InterfaceType'>
  <xs:sequence>
    <xs:element name='IPAddress'       type='IPAddressType'/>
  </xs:sequence>
  <xs:attribute name='interfaceName'   type='xs:string' use='required'/>
  <xs:attribute name='clusterNodeName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

IPAddress

Typ: IPAddressType

Das Element IPAddressType enthält alle Einzelangaben zu einer IP-Adresse, z. B. die *Basisadresse*, die *Teilnetzmaske* und den Namen des Netzes, zu dem die IP-Adresse gehört.

Regeln zur Häufigkeit:
Geben Sie eine einzige IP-Adresse für IPAddress im Element InterfaceType an.

Attribute

interfaceName (erforderlich)
Geben Sie den Namen einer Netzschnittstellenkarte im Attribut interfaceName an. Die für interfaceName angegebene Netzschnittstellenkarte muss in dem Clusterdomänenknoten vorhanden sein, den Sie für das Attribut clusterNodeName angeben.

clusterNodeName (erforderlich)
Geben Sie den Namen des Clusterdomänenknotens an, der der IP-Adresse zugeordnet ist, die Sie für das Element IPAddress angeben.

IPAddressType - XML-Schemaelement für Eingabedateien für db2haicu:

Das Element IPAddressType enthält alle Einzelangaben zu einer IP-Adresse, z. B. die *Basisadresse*, die *Teilnetzmaske* und den Namen des Netzes, zu dem die IP-Adresse gehört.

„Superelemente"
„XML-Schemadefinition "
„Subelemente "
„Attribute"

Superelemente

Die folgenden Elementtypen enthalten IPAddressType-Subelemente:

- PhysicalNetworkType
- InterfaceType
- DB2PartitionType

XML-Schemadefinition

```
<xs:complexType name='IPAddressType'>
  <xs:attribute name='baseAddress' type='xs:string' use='required'/>
  <xs:attribute name='subnetMask'  type='xs:string' use='required'/>
  <xs:attribute name='networkName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

baseAddress (erforderlich)
Geben Sie die IP-Basisadresse in Form einer Zeichenfolge an, die ein gülti-

ges IP-Adressformat aufweist: vier durch einen Punkt getrennte Zifferngruppen im Bereich von 0 bis 255. Beispiel:

```
162.148.31.101
```

subnetMask (erforderlich)
Geben Sie die IP-Basisadresse in Form einer Zeichenfolge an, die ein gültiges IP-Adressformat aufweist.

networkName (erforderlich)
Geben Sie für `networkName` denselben Wert an wie für das Attribut `physicalNetworkName` des Elements PhysicalNetworkType, in dem das Element `IPAddress` enthalten ist.

ClusterNodeType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `ClusterNodeType` enthält Informationen zu einem bestimmten Computer oder einer bestimmten Maschine (auch *Clusterdomänenknoten* genannt) des Clusters.

„Superelemente"
„XML-Schemadefinition "
„Subelemente "
„Attribute"

Superelemente

Die folgenden Elementtypen enthalten `ClusterNodeType`-Subelemente:

- `ClusterDomainType`

XML-Schemadefinition

```
<xs:complexType name='ClusterNodeType'>
  <xs:attribute name='clusterNodeName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

clusterNodeName (erforderlich)
Geben Sie den Namen des Clusterdomänenknotens an.

FailoverPolicyType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `FailoverPolicyType` gibt die *Richtlinie für Funktionsübernahme* an, die der Cluster-Manager in der Clusterdomäne anwenden soll.

„Superelemente"
„XML-Schemadefinition " auf Seite 117
„Subelemente" auf Seite 117
„Gültige Werte " auf Seite 117

Superelemente

Die folgenden Elementtypen enthalten `InterfaceType`-Subelemente:

- `DB2ClusterType`

XML-Schemadefinition

```
<xs:complexType name='FailoverPolicyType'>
 <xs:choice>
   <xs:element name='RoundRobin'   type='xs:string' minOccurs='0' />
   <xs:element name='Mutual'       type='xs:string' minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='NPlusM'       type='xs:string' minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='LocalRestart' type='xs:string' fixed=''/>
   <xs:element name='HADRFailover' type='xs:string' fixed=''/>
   <xs:element name='Custom'       type='xs:string' minOccurs='0' />
 </xs:choice>
</xs:complexType>
```

Subelemente

Keine

Gültige Werte

Geben Sie dem Cluster-Manager eine der folgenden Optionen als den Funktionsübernahmetyp an, der im Falle von Störungen in der Clusterdomäne angewendet werden soll:

Eine *Richtlinie für Funktionsübernahme* gibt an, wie ein Cluster-Manager sich verhalten soll, wenn ein Clusterelement, wie z. B. eine Netzschnittstellenkarte oder ein Datenbankserver, ausfällt. Im Allgemeinen wird ein Cluster-Manager die Workload in diesem Fall von dem ausgefallenen Element auf ein alternatives Element übertragen, das zuvor für den Cluster-Manager als geeigneter Ersatz für das ausgefallene Element definiert wurde. Diese Übertragung der Workload von einem ausgefallenen Element auf ein anderes Element wird als *Funktionsübernahme* bezeichnet.

RoundRobin
: Die *Funktionsübernahmerichtlinie 'Umlauf'* (RoundRobin) sieht vor, dass der Datenbankmanager bei einem Ausfall eines einzigen Computers (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne die Workload des ausgefallenen Clusterdomänenknotens auf einem beliebigen anderen Knoten in der Clusterdomäne erneut startet.

Mutual
: Für die Konfiguration der *Funktionsübernahmerichtlinie 'Gegenseitige Übernahme'* (Mutual) müssen Sie zwei Computer (auch *Clusterdomänenknoten* oder einfach *Knoten* genannt) in der Clusterdomäne als ein Systempaar zuordnen. Tritt bei einem dieser beiden Knoten ein Ausfall auf, wird die Workload der Datenbankpartitionen des ausgefallenen Knotens von dem anderen Knoten übernommen. Eine gegenseitige Übernahme ist nur möglich, wenn Sie über mehrere Datenbankpartitionen verfügen.

NPlusM
: Wenn Sie die *Funktionsübernahmerichtlinie 'M+N'* (NPlusM) verwenden, wird bei einem Ausfall eines Computers (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne die Workload der Datenbankpartitionen auf dem ausgefallenen Knoten von einem beliebigen anderen Knoten in der Clusterdomäne übernommen. Eine M+N-Übernahme ist nur möglich, wenn Sie über mehrere Datenbankpartitionen verfügen.

LocalRestart
: Wenn Sie die *Funktionsübernahmerichtlinie 'Lokaler Neustart'* (LocalRestart) verwenden, startet der Datenbankmanager bei einem Ausfall auf einem Computer (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne die (lokal) vorhandene Datenbank auf dem Knoten, der ausgefallen ist, neu.

HADRFailover

Wenn Sie die *Funktionsübernahmerichtlinie 'HADR'* (HADRFailover) konfigurieren, ermöglichen Sie der DB2 HADR-Funktion (High Availability Disaster Recovery) die Verwaltung der Funktionsübernahme. Fällt eine HADR-Primärdatenbank aus, versetzt der Datenbankmanager die Workload von der ausgefallenen Datenbank auf eine HADR-Bereitschaftsdatenbank.

Custom

Wenn Sie die *Funktionsübernahmerichtlinie 'Benutzerdefiniert'* (Custom) konfigurieren, erstellen Sie eine Liste von Computern (auch *Clusterdomänenknoten* oder kurz *Knoten* genannt) in der Clusterdomäne, die der Datenbankmanager für eine Funktionsübernahme nutzen kann. Fällt ein Knoten in der Clusterdomäne aus, versetzt der Datenbankmanager die Workload vom ausgefallenen Knoten auf einen in der von Ihnen angegebenen Liste enthaltenen Knoten.

DB2PartitionSetType - XML-Schemadefinition für Eingabedateien für DB2 High Availability Instance Configuration Utility (db2haicu):

Das Element `DB2PartitionSetType` enthält Informationen zu Datenbankpartitionen. Eine Verwendung des Elements `DB2PartitionSetType` ist nur in Umgebungen mit partitionierten Datenbanken möglich.

„Superelemente"
„XML-Schemadefinition "
„Subelemente"
„Attribute"

Superelemente

`InterfaceType` ist ein Subelement von:

- `PhysicalNetworkType`

XML-Schemadefinition

```
<xs:complexType name='DB2PartitionSetType'>
  <xs:sequence>
    <xs:element name='DB2Partition'         type='DB2PartitionType' maxOccurs='unbounded'/>
  </xs:sequence>
</xs:complexType>
```

Subelemente

DB2Partition

Typ: `DB2PartitionType`

Das Element `DB2PartitionType` gibt eine Datenbankpartition sowie die DB2-Datenbankmanagerinstanz, zu der die Datenbankpartition gehört, und die Nummer der Datenbankpartition an.

Regeln zur Häufigkeit:
Sie müssen mindestens ein Element `DB2Partition` im Element `DB2PartitionSetType` angeben.

Attribute

Keine

DB2PartitionType - XML-Schemaelement für Eingabedateien für db2haicu:

Das Element DB2PartitionType gibt eine Datenbankpartition sowie die DB2-Datenbankmanagerinstanz, zu der die Datenbankpartition gehört, und die Nummer der Datenbankpartition an.

„Superelemente"
„XML-Schemadefinition "
„Subelemente "
„Attribute " auf Seite 120

Superelemente

InterfaceType ist ein Subelement von:

- DB2PartitionSetType

XML-Schemadefinition

```
<xs:complexType name='DB2PartitionType'>
  <xs:sequence>
   <xs:element name='VirtualIPAddress' type='IPAddressType'    minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='Mount'            type='MountType'        minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='HADRDB'           type='HADRDBType'       minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='MutualPair'       type='MutualPolicyType' minOccurs='0' maxOccurs='1'/>
   <xs:element name='NPlusMNode'       type='NPlusMPolicyType' minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='CustomNode'       type='CustomPolicyType' minOccurs='0' maxOccurs='unbounded'/>
 </xs:sequence>
 <xs:attribute name='instanceName'     type='xs:string'  use='required'/>
 <xs:attribute name='dbpartitionnum'   type='xs:integer' use='required'/>
</xs:complexType>
```

Subelemente

VirtualIPAddress

Typ: IPAddressType

Das Element IPAddressType enthält alle Einzelangaben zu einer IP-Adresse, z. B. die *Basisadresse*, die *Teilnetzmaske* und den Namen des Netzes, zu dem die IP-Adresse gehört.

Sie können die Angabe von VirtualIPAddress auslassen oder eine unbegrenzte Anzahl von VirtualIPAddress-Elementen in das Element DB2PartitionType einfügen.

Mount

Typ: MountType

Das Element MountType enthält Informationen zu einem *Mountpunkt*, z. B. dem Dateipfad, der die Position der angehängten Dateien angibt.

Sie können die Angabe von Mount auslassen oder eine unbegrenzte Anzahl von Mount-Elementen in das Element DB2PartitionType einfügen.

HADRDB

Typ: HADRDBType

Das Element HADRDBType enthält eine Liste mit Datenbankpaaren, die aus einer HADR-Primärdatenbank (High Availability Disaster Recovery) und einer HADR-Bereitschaftsdatenbank bestehen.

Sie können die Angabe von HADRDB auslassen oder eine unbegrenzte Anzahl von HADRDB-Elementen in das Element DB2PartitionType einfügen.

MutualPair

Typ: MutualPolicyType

Das Element MutualPolicyType enthält Informationen zu einem Paar von Clusterdomänenknoten, die für eine gegenseitige Funktionsübernahme definiert sind.

Sie können die Angabe von MutualPair auslassen oder eine unbegrenzte Anzahl von MutualPair-Elementen in das Element DB2PartitionType einfügen.

NPlusMNode
Typ: NPlusMPolicyType

Sie können die Angabe von NPlusMNode auslassen oder eine unbegrenzte Anzahl von NPlusMNode-Elementen in das Element DB2PartitionType einfügen.

CustomNode
Typ: CustomPolicyType

Sie können auf die Angabe von CustomNode verzichten oder eine unbegrenzte Anzahl von CustomNode-Elementen in das Element DB2PartitionType einfügen.

Attribute

instanceName (erforderlich)
Geben Sie im Attribut instanceName die Datenbankmanagerinstanz an, der dieses Element DB2PartitionType zugeordnet ist.

dbpartitionnum (erforderlich)
Geben Sie im Attribut dbpartitionnum die Datenbankpartitionsnummer an, die die Datenbankpartition eindeutig kennzeichnet (die Nummer dbpartitionnum wird z. B. in der Datei db2nodes.cfg angegeben).

MountType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element MountType enthält Informationen zu einem *Mountpunkt*, z. B. dem Dateipfad, der die Position der angehängten Dateien angibt.

„Superelemente"
„XML-Schemadefinition "
„Subelemente "
„Attribute" auf Seite 121

Superelemente

Die folgenden Elementtypen enthalten MountType-Subelemente:

- DB2PartitionType

XML-Schemadefinition

```
<xs:complexType name='MountType'>
  <xs:attribute name='filesystemPath' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

filesystemPath (erforderlich)
Geben Sie den Pfad an, der dem Mountpunkt beim Anhängen des Dateisystems zugeordnet wurde.

MutualPolicyType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `MutualPolicyType` enthält Informationen zu einem Paar von Clusterdomänenknoten, die für eine gegenseitige Funktionsübernahme definiert sind.

„Superelement"
„XML-Schemadefinition "
„Subelemente "
„Attribute"

Superelement

Die folgenden Elementtypen enthalten `MutualPolicyType`-Subelemente:

- `DB2PartitionType`

XML-Schemadefinition

```
<xs:complexType name='MutualPolicyType'>
  <xs:attribute name='systemPairNode1' type='xs:string' use='required'/>
  <xs:attribute name='systemPairNode2' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

systemPairNode1 (erforderlich)
Geben Sie für das Attribut `systemPairNode1` den Namen eines Clusterdomänenknotens an, der für den im Attribut `systemPairNode2` angegebenen Clusterdomänenknoten bei einer Funktionsübernahme verfügbar ist.

systemPairNode2 (erforderlich)
Geben Sie für das Attribut `systemPairNode2` den Namen eines Clusterdomänenknotens an, der für den im Attribut `systemPairNode1` angegebenen Clusterdomänenknoten bei einer Funktionsübernahme verfügbar ist.

NPlusMPolicyType - XML-Schemadefinition für Eingabedateien für db2haicu:

„Superelemente"
„XML-Schemadefinition " auf Seite 122
„Subelemente " auf Seite 122
„Attribute" auf Seite 122

Superelemente

Die folgenden Elementtypen enthalten `NPlusMPolicyType`-Subelemente:

- `DB2PartitionType`

XML-Schemadefinition

```
<xs:complexType name='NPlusMPolicyType'>
  <xs:attribute name='standbyNodeName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

standbyNodeName (erforderlich)
Geben Sie im Element `standbyNodeName` den Namen eines Clusterdomänenknotens an, der bei einer Funktionsübernahme für die Partition verfügbar ist, die das Element `NPlusMPolicyType` enthält.

CustomPolicyType - XML-Schemadefinition für Eingabedateien für DB2 High Availability Instance Configuration Utility (db2haicu):

„Superelemente”
„XML-Schemadefinition ”
„Subelemente”
„Attribute”

Superelemente

Die folgenden Elementtypen enthalten `CustomPolicyType`-Subelemente:

- `DB2PartitionType`

XML-Schemadefinition

```
<xs:complexType name='NPlusMPolicyType'>
  <xs:attribute name='standbyNodeName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

customNodeName (erforderlich)
Geben Sie im Element `customNodeName` den Namen eines Clusterdomänenknotens an, der bei einer Funktionsübernahme für die Partition verfügbar ist, die das Element `CustomPolicyType` enthält.

HADRDBType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element `HADRDBType` enthält eine Liste mit Datenbankpaaren, die aus einer HA-DR-Primärdatenbank (High Availability Disaster Recovery) und einer HADR-Bereitschaftsdatenbank bestehen.

„Superelemente” auf Seite 123
„XML-Schemadefinition ” auf Seite 123
„Subelemente ” auf Seite 123
„Attribute” auf Seite 123
„Hinweise zur Verwendung” auf Seite 123
„Einschränkungen” auf Seite 123

Superelemente

Die folgenden Elementtypen enthalten HADRDBType-Subelemente:

- DB2ClusterType
- DB2PartitionType

XML-Schemadefinition

```
<xs:complexType name='HADRDBType'>
 <xs:sequence>
   <xs:element name='VirtualIPAddress' type='IPAddressType' minOccurs='0' maxOccurs='unbounded'/>
   <xs:element name='HADRDB'           type='HADRDBDefn'                   maxOccurs='unbounded'/>
 </xs:sequence>
</xs:complexType>
```

Subelemente

VirtualIPAddress

Typ: IPAddressType

Das Element IPAddressType enthält alle Einzelangaben zu einer IP-Adresse, z. B. die *Basisadresse*, die *Teilnetzmaske* und den Namen des Netzes, zu dem die IP-Adresse gehört.

Regeln zur Häufigkeit:
Sie können keine oder beliebig viele Elemente VirtualIPAddress in das Element HADRDBType aufnehmen.

HADRDB

Typ: HADRDBDefn

Das Element HADRDBDefn enthält Informationen zu einem HADR-Datenbankpaar (High Availability Disaster Recovery), das aus einer HADR-Primärdatenbank und einer HADR-Bereichschaftsdatenbank besteht.

Regeln zur Häufigkeit:
Sie können das Element VirtualIPAddress beliebig oft in das Element HADRDBType aufnehmen.

Attribute

Keine

Hinweise zur Verwendung

Wenn Sie ein Element HADRDBType in die Spezifikation für eine bestimmte Clusterdomäne aufnehmen, müssen Sie in derselben Clusterdomänenspezifikation ebenfalls ein Element FailoverPolicy einfügen, das HADRFailover angibt.

Einschränkungen

Sie können das Element HADRDBType nicht in einer Umgebung mit partitionierten Datenbanken verwenden.

HADRDBDefn - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element HADRDBDefn enthält Informationen zu einem HADR-Datenbankpaar (High Availability Disaster Recovery), das aus einer HADR-Primärdatenbank und einer HADR-Bereichschaftsdatenbank besteht.

„Superelemente”
„XML-Schemadefinition ”
„Subelemente ”
„Attribute”

Superelemente

Die folgenden Elementtypen enthalten HADRDBDefn-Subelemente:

- HADRDBType

XML-Schemadefinition

```
<xs:complexType name='HADRDBDefn'>
  <xs:attribute name='databaseName'      type='xs:string' use='required'/>
  <xs:attribute name='localInstance'     type='xs:string' use='required'/>
  <xs:attribute name='remoteInstance'    type='xs:string' use='required'/>
  <xs:attribute name='localHost'         type='xs:string' use='required'/>
  <xs:attribute name='remoteHost'        type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

databaseName (erforderlich)
Geben Sie den Namen der HADR-Datenbank an.

localInstance (erforderlich)
localInstance gibt die Datenbankmanagerinstanz der HADR-Primärdatenbank an.

remoteInstance (erforderlich)
remoteInstance gibt die Datenbankmanagerinstanz der HADR-Bereitschaftsdatenbank an.

localHost (erforderlich)
localHost gibt den Hostnamen des Clusterdomänenknotens an, auf dem sich die HADR-Primärdatenbank befindet.

remoteHost (erforderlich)
remoteHost gibt den Hostnamen des Clusterdomänenknotens an, auf dem sich die HADR-Bereitschaftsdatenbank befindet.

HADBType - XML-Schemadefinition für Eingabedateien für db2haicu:

Das Element HADBType enthält eine Liste der Datenbanken, die zu der Clusterdomäne gehören und eine hohe Verfügbarkeit aufweisen sollen.

„Superelemente”
„XML-Schemadefinition ” auf Seite 125
„Subelemente ” auf Seite 125
„Attribute” auf Seite 125

Superelemente

Die folgenden Elementtypen enthalten HADBType-Subelemente:

- DB2ClusterType

XML-Schemadefinition

```
<xs:complexType name='HADBType'>
  <xs:sequence>
    <xs:element name='HADB'         type='HADBDefn' maxOccurs='unbounded'/>
  </xs:sequence>
  <xs:attribute name='instanceName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

HADB

Typ: HADBDefn

Das Element HADBDefn beschreibt eine Datenbank, die als Datenbank mit hoher Verfügbarkeit in die Clusterdomäne aufgenommen werden soll.

Regeln zur Häufigkeit:
Fügen Sie mindestens ein Element HADB im Element HADBType hinzu.

Attribute

instanceName (erforderlich)
Geben Sie im Attribut instanceName die DB2-Datenbankmanagerinstanz an, zu der die in den Elementen HADB angegebenen Datenbanken gehören.

HADBDefn - XML-Schemaelement für Eingabedateien für db2haicu:

Das Element HADBDefn beschreibt eine Datenbank, die als Datenbank mit hoher Verfügbarkeit in die Clusterdomäne aufgenommen werden soll.

„Superelemente"
„XML-Schemadefinition "
„Subelemente "
„Attribute"

Superelemente

HADBDefn ist ein Subelement von:

- HADRDBType

XML-Schemadefinition

```
<xs:complexType name='HADBDefn'>
  <xs:attribute name='databaseName' type='xs:string' use='required'/>
</xs:complexType>
```

Subelemente

Keine

Attribute

databaseName (erforderlich)
Geben Sie einen einzigen Datenbanknamen im Attribut databaseName an.

XML-Beispieleingabedateien für db2haicu:

Im Verzeichnis `sqllib` befindet sich im Unterverzeichnis `samples` eine Gruppe von XML-Beispieleingabedateien, die Sie modifizieren und mit db2haicu zum Konfigurieren der Clusterumgebung verwenden können.

db2ha_sample_sharedstorage_mutual.xml:

Die Datei `db2ha_sample_sharedstorage_mutual.xml` ist ein Beispiel für eine XML-Eingabedatei, die an DB2 High Availability Instance Configuration Utility (db2haicu) übergeben wird, um eine neue *Clusterdomaine* anzugeben. `db2ha_sample_sharedstorage_mutual.xml` befindet sich im Verzeichnis `sqllib/samples/ha/xml`.

Merkmale

Die Beispieldatei `db2ha_sample_sharedstorage_mutual.xml` veranschaulicht, wie Sie db2haicu mit einer XML-Eingabedatei verwenden können, um eine Clusterdomäne mit den folgenden Details zu definieren:
- Quorumeinheit: Netz
- Anzahl der Computer im Cluster (Clusterdomänenknoten): 2
- Richtlinie für Funktionsübernahme: Gegenseitige Übernahme
- Datenbankpartitionen: 1
- Virtuelle IP-Adressen (Service-IP-Adressen): 1
- Gemeinsame Mountpunkte für Funktionsübernahme: 1

XML-Quellencode

```
<!-- ================================================================ -->
<!-- = Verwenden Sie die XML-Schemadefinition db2ha.xsd für          = -->
<!-- = db2haicu, und geben Sie IBM Tivoli System                     = -->
<!-- = Automation for Multiplatforms (SA MP) Base Component          = -->
<!-- = als Cluster-Manager an.                                       = -->
<!-- ================================================================ -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:noNamespaceSchemaLocation="db2ha.xsd"
            clusterManagerName="TSA"
            version="1.0">

  <!-- ================================================================ -->
  <!-- = Erstellen Sie eine Clusterdomäne mit dem Namen 'db2HAdomain'. = -->
  <!-- ================================================================ -->
  <ClusterDomain domainName="db2HAdomain">

    <!-- ================================================================ -->
    <!-- = Geben Sie eine Netzquorumeinheit (IP-Adresse: 19.126.4.5)   = -->
    <!-- = an. Die IP muss jederzeit von allen Clusterdomänenknoten    = -->
    <!-- = mit Ping überprüft werden können.                           = -->
    <!-- ================================================================ -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ================================================================ -->
    <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_public_network_0' = -->
    <!-- = mit einem IP-Netzprotokoll.                                 = -->
    <!-- = Dieses Netz enthält 2 Computer: 'hasys01' und 'hasys02'.    = -->
    <!-- = Jeder Computer hat eine einzige Netzschnittstellenkarte     = -->
    <!-- = (NIC) mit der Bezeichnung 'eth0'.                           = -->
    <!-- = Die IP-Adresse der NIC auf 'hasys01' ist '19.126.52.139'.   = -->
    <!-- = Die IP-Adresse der NIC auf 'hasys02' ist '19.126.52.140'.   = -->
    <!-- ================================================================ -->
    <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                     physicalNetworkProtocol="ip">
```

```
      <Interface interfaceName="eth0" clusterNodeName="hasys01">
        <IPAddress baseAddress="19.126.52.139"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys02">
        <IPAddress baseAddress="19.126.52.140"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

    </PhysicalNetwork>

    <!-- ================================================================ -->
    <!-- = Listen Sie die Computer (Knoten) in der Clusterdomäne auf.  = -->
    <!-- ================================================================ -->
    <ClusterNode clusterNodeName="hasys01"/>
    <ClusterNode clusterNodeName="hasys02"/>

  </ClusterDomain>


  <!-- ================================================================ -->
  <!-- = Die Richtlinie für Funktionsübernahme gibt die Reihenfolge    = -->
  <!-- = für die Funktionsübernahme bei den Clusterdomänenknoten an.   = -->
  <!-- ================================================================ -->
  <FailoverPolicy>
     <Mutual />
  </FailoverPolicy>


  <!-- ================================================================ -->
  <!-- = Geben Sie alle Details zu der Datenbankpartition an.          = -->
  <!-- ================================================================ -->
  <DB2PartitionSet>

    <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
       <VirtualIPAddress baseAddress="19.126.52.222"
                         subnetMask="255.255.255.0"
                         networkName="db2_public_network_0"/>
       <Mount filesystemPath="/home/db2inst1"/>
       <MutualPair systemPairNode1="hasys01" systemPairNode2="hasys02" />
    </DB2Partition>

  </DB2PartitionSet>

</DB2Cluster>
```

db2ha_sample_DPF_mutual.xml:

Die Datei db2ha_sample_DPF_mutual.xml ist ein Beispiel für eine XML-Eingabedatei, die an DB2 High Availability Instance Configuration Utility (db2haicu) übergeben wird, um eine neue *Clusterdomaine* anzugeben. db2ha_sample_DPF_mutual.xml befindet sich im Verzeichnis sqllib/samples/ha/xml.

Merkmale

Die Beispieldatei db2ha_sample_DPF_mutual.xml veranschaulicht, wie Sie db2haicu mit einer XML-Eingabedatei verwenden können, um eine Clusterdomäne mit den folgenden Details zu definieren:

- Quorumeinheit: Netz
- Anzahl der Computer im Cluster (Clusterdomänenknoten): 4

- Richtlinie für Funktionsübernahme: Gegenseitige Übernahme
- Datenbankpartitionen: 2
- Virtuelle IP-Adressen (Service-IP-Adressen): 1
- Gemeinsame Mountpunkte für Funktionsübernahme: 2
- Für hohe Verfügbarkeit konfigurierte Datenbanken: 2

XML-Quellencode

```
<!-- ================================================================ -->
<!-- = Verwenden Sie die XML-Schemadefinition db2ha.xsd für          = -->
<!-- = db2haicu, und geben Sie IBM Tivoli System                     = -->
<!-- = Automation for Multiplatforms (SA MP) Base Component          = -->
<!-- = als Cluster-Manager an.                                       = -->
<!-- ================================================================ -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:noNamespaceSchemaLocation="db2ha.xsd"
            clusterManagerName="TSA"
            version="1.0">

  <!-- ================================================================ -->
  <!-- = Erstellen Sie eine Clusterdomäne mit dem Namen 'db2HAdomain'. = -->
  <!-- ================================================================ -->
  <ClusterDomain domainName="db2HAdomain">

    <!-- ================================================================ -->
    <!-- = Geben Sie eine Netzquorumeinheit (IP-Adresse: 19.126.4.5)   = -->
    <!-- = an. Die IP muss jederzeit von allen Clusterdomänenknoten    = -->
    <!-- = mit Ping überprüft werden können.                           = -->
    <!-- ================================================================ -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ================================================================ -->
    <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_public_network_0' = -->
    <!-- = mit einem IP-Netzprotokoll.                                 = -->
    <!-- = Dieses Netz enthält vier Computer: hasys01, hasys02,        = -->
    <!-- = hasys03 und hasys04.                                        = -->
    <!-- = Jeder Computer hat eine Netzschnittstellenkarte (eth0).     = -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys01' ist '19.126.124.30'.= -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys02' ist '19.126.124.31'.= -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys03' ist '19.126.124.32'.= -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys04' ist '19.126.124.33'.= -->
    <!-- ================================================================ -->
    <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                     physicalNetworkProtocol="ip">

      <Interface interfaceName="eth0" clusterNodeName="hasys01">
        <IPAddress baseAddress="19.126.124.30"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys02">
        <IPAddress baseAddress="19.126.124.31"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys03">
        <IPAddress baseAddress="19.126.124.32"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys04">
        <IPAddress baseAddress="19.126.124.33"
```

```
                subnetMask="255.255.255.0"
                networkName="db2_public_network_0"/>
    </Interface>

  </PhysicalNetwork>

  <!-- ============================================================= -->
  <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_private_network_0'= -->
  <!-- = mit einem IP-Netzprotokoll.                                 = -->
  <!-- = Dieses Netz enthält vier Computer: hasys01, hasys02,        = -->
  <!-- = hasys03 und hasys04 (wie 'db2_public_network_0'.)           = -->
  <!-- = Zusätzlich zu 'eth0' verfügt jeder Computer über eine       = -->
  <!-- = Netzschnittstellenkarte mit dem Namen 'eth1'.               = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys01' ist 192.168.23.101. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys02' ist 192.168.23.102. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys03' ist 192.168.23.103. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys04' ist 192.168.23.104. = -->
  <!-- ============================================================= -->
  <PhysicalNetwork physicalNetworkName="db2_private_network_0"
                   physicalNetworkProtocol="ip">

    <Interface interfaceName="eth1" clusterNodeName="hasys01">
      <IPAddress baseAddress="192.168.23.101"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys02">
      <IPAddress baseAddress="192.168.23.102"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys03">
      <IPAddress baseAddress="192.168.23.103"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys04">
      <IPAddress baseAddress="192.168.23.104"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

  </PhysicalNetwork>

  <!-- ============================================================= -->
  <!-- = Listen Sie die Computer (Knoten) in der Clusterdomäne auf.  = -->
  <!-- ============================================================= -->
  <ClusterNode clusterNodeName="hasys01"/>
  <ClusterNode clusterNodeName="hasys02"/>
  <ClusterNode clusterNodeName="hasys03"/>
  <ClusterNode clusterNodeName="hasys04"/>

</ClusterDomain>


<!-- ============================================================= -->
<!-- = Die Richtlinie für Funktionsübernahme gibt die Reihenfolge    = -->
<!-- = für die Funktionsübernahme bei den Clusterdomänenknoten an.   = -->
<!-- ============================================================= -->
<FailoverPolicy>
   <Mutual />
</FailoverPolicy>
```

```
  <!-- ================================================================ -->
  <!-- = Geben Sie alle Details zu den Datenbankpartitionen an.        = -->
  <!-- ================================================================ -->
  <DB2PartitionSet>

    <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
       <VirtualIPAddress baseAddress="19.126.124.251"
                         subnetMask="255.255.255.0"
                         networkName="db2_public_network_0"/>
       <Mount filesystemPath="/hafs/db2inst1/NODE0000"/>
       <MutualPair systemPairNode1="hasys01" systemPairNode2="hasys02" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="1" instanceName="db2inst1">
       <Mount filesystemPath="/hafs/db2inst1/NODE0001"/>
       <MutualPair systemPairNode1="hasys02" systemPairNode2="hasys01" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="2" instanceName="db2inst1">
       <Mount filesystemPath="/hafs/db2inst1/NODE0002"/>
       <MutualPair systemPairNode1="hasys03" systemPairNode2="hasys04" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="3" instanceName="db2inst1">
       <Mount filesystemPath="/hafs/db2inst1/NODE0003"/>
       <MutualPair systemPairNode1="hasys04" systemPairNode2="hasys03" />
    </DB2Partition>

  </DB2PartitionSet>


  <!-- ================================================================ -->
  <!-- = Liste der als hochverfügbar zu konfigurierenden Datenbanken   = -->
  <!-- ================================================================ -->
  <HADBSet instanceName="db2inst1">
    <HADB databaseName = "SAMPLE" />
    <HADB databaseName = "MYDB" />
  </HADBSet>


</DB2Cluster>
```

db2ha_sample_DPF_NPlusM.xml:

Die Datei db2ha_sample_DPF_NPlusM.xml ist ein Beispiel für eine XML-Eingabedatei, die an DB2 High Availability Instance Configuration Utility (db2haicu) übergeben wird, um eine neue *Clusterdomaine* anzugeben. db2ha_sample_DPF_NPlusM.xml befindet sich im Verzeichnis sqllib/samples/ha/xml.

Merkmale

Die Beispieldatei db2ha_sample_DPF_NPlusM.xml veranschaulicht, wie Sie db2haicu mit einer XML-Eingabedatei verwenden können, um eine Clusterdomäne mit den folgenden Details zu definieren:

- Quorumeinheit: Netz
- Anzahl der Computer im Cluster (Clusterdomänenknoten): 4
- Richtlinie für Funktionsübernahme: M+N
- Datenbankpartitionen: 2
- Virtuelle IP-Adressen (Service): 1
- Gemeinsame Mountpunkte für Funktionsübernahme: 4

XML-Quellencode

```
<!-- ================================================================= -->
<!-- = Verwenden Sie die XML-Schemadefinition db2ha.xsd für          = -->
<!-- = db2haicu, und geben Sie IBM Tivoli System                     = -->
<!-- = Automation for Multiplatforms (SA MP) Base Component          = -->
<!-- = als Cluster-Manager an.                                       = -->
<!-- ================================================================= -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:noNamespaceSchemaLocation="db2ha.xsd"
            clusterManagerName="TSA"
            version="1.0">

  <!-- ================================================================= -->
  <!-- = Erstellen Sie eine Clusterdomäne mit dem Namen 'db2HAdomain'. = -->
  <!-- ================================================================= -->
  <ClusterDomain domainName="db2HAdomain">

    <!-- ================================================================= -->
    <!-- = Geben Sie eine Netzquorumeinheit (IP-Adresse: 19.126.4.5)   = -->
    <!-- = an. Die IP muss jederzeit von allen Clusterdomänenknoten    = -->
    <!-- = mit Ping überprüft werden können.                           = -->
    <!-- ================================================================= -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ================================================================= -->
    <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_public_network_0' = -->
    <!-- = mit einem IP-Netzprotokoll.                                 = -->
    <!-- = Dieses Netz enthält vier Computer: hasys01, hasys02,        = -->
    <!-- = hasys03 und hasys04.                                        = -->
    <!-- = Jeder Computer hat eine Netzschnittstellenkarte (eth0).     = -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys01' ist '19.126.124.30'.= -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys02' ist '19.126.124.31'.= -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys03' ist '19.126.124.32'.= -->
    <!-- = Die IP-Adresse von 'eth0' auf 'hasys04' ist '19.126.124.33'.= -->
    <!-- ================================================================= -->
    <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                     physicalNetworkProtocol="ip">

      <Interface interfaceName="eth0" clusterNodeName="hasys01">
        <IPAddress baseAddress="19.126.124.30"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys02">
        <IPAddress baseAddress="19.126.124.31"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys03">
        <IPAddress baseAddress="19.126.124.32"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

      <Interface interfaceName="eth0" clusterNodeName="hasys04">
        <IPAddress baseAddress="19.126.124.33"
                   subnetMask="255.255.255.0"
                   networkName="db2_public_network_0"/>
      </Interface>

    </PhysicalNetwork>

    <!-- ================================================================= -->
    <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_private_network_0'= -->
    <!-- = mit einem IP-Netzprotokoll.                                 = -->
```

```
  <!-- = Dieses Netz enthält vier Computer: hasys01, hasys02,        = -->
  <!-- = hasys03 und hasys04 (wie 'db2_public_network_0'.)           = -->
  <!-- = Zusätzlich zu 'eth0' verfügt jeder Computer über eine       = -->
  <!-- = Netzschnittstellenkarte mit dem Namen 'eth1'.               = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys01' ist 192.168.23.101. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys02' ist 192.168.23.102. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys03' ist 192.168.23.103. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys04' ist 192.168.23.104. = -->
  <!-- ================================================================ -->
  <PhysicalNetwork physicalNetworkName="db2_private_network_0"
                   physicalNetworkProtocol="ip">

    <Interface interfaceName="eth1" clusterNodeName="hasys01">
      <IPAddress baseAddress="192.168.23.101"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys02">
      <IPAddress baseAddress="192.168.23.102"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys03">
      <IPAddress baseAddress="192.168.23.103"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys04">
      <IPAddress baseAddress="192.168.23.104"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

  </PhysicalNetwork>

  <!-- ================================================================ -->
  <!-- = Listen Sie die Computer (Knoten) in der Clusterdomäne auf.  = -->
  <!-- ================================================================ -->
  <ClusterNode clusterNodeName="hasys01"/>
  <ClusterNode clusterNodeName="hasys02"/>
  <ClusterNode clusterNodeName="hasys03"/>
  <ClusterNode clusterNodeName="hasys04"/>

</ClusterDomain>


<!-- ================================================================ -->
<!-- = Die Richtlinie für Funktionsübernahme gibt die Reihenfolge    = -->
<!-- = für die Funktionsübernahme bei den Clusterdomänenknoten an.   = -->
<!-- ================================================================ -->
<FailoverPolicy>
   <NPlusM />
</FailoverPolicy>


<!-- ================================================================ -->
<!-- = Geben Sie alle Details zu den Datenbankpartitionen an.        = -->
<!-- ================================================================ -->
<DB2PartitionSet>

  <DB2Partition dbpartitionnum="0" instanceName="db2inst1">
     <VirtualIPAddress baseAddress="19.126.124.250"
                       subnetMask="255.255.255.0"
                       networkName="db2_public_network_0"/>
```

```
      <Mount filesystemPath="/ha_dpf1/db2inst1/NODE0000"/>
      <Mount filesystemPath="/hafs/NODE0000"/>
      <NPlusMNode standbyNodeName="hasys03" />
    </DB2Partition>

    <DB2Partition dbpartitionnum="1" instanceName="db2inst1">
      <Mount filesystemPath="/ha_dpf1/db2inst1/NODE0001"/>
      <Mount filesystemPath="/hafs/NODE0001"/>
      <NPlusMNode standbyNodeName="hasys04" />
    </DB2Partition>

  </DB2PartitionSet>

</DB2Cluster>
```

db2ha_sample_HADR.xml:

Die Datei `db2ha_sample_DPF_HADR.xml` ist ein Beispiel für eine XML-Eingabedatei, die an DB2 High Availability Instance Configuration Utility (db2haicu) übergeben wird, um eine neue *Clusterdomaine* anzugeben. `db2ha_sample_HADR.xml` befindet sich im Verzeichnis `sqllib/samples/ha/xml`.

Merkmale

Die Beispieldatei `db2ha_sample_HADR.xml` veranschaulicht, wie Sie db2haicu mit einer XML-Eingabedatei verwenden können, um eine Clusterdomäne mit den folgenden Details zu definieren:

- Quorumeinheit: Netz
- Anzahl der Computer im Cluster (Clusterdomänenknoten): 2
- Richtlinie für Funktionsübernahme: HADR
- Datenbankpartitionen: 1
- Virtuelle IP-Adressen (Service-IP-Adressen): Keine
- Gemeinsame Mountpunkte für Funktionsübernahme: Keine

XML-Quellencode

```
<!-- ================================================================== -->
<!-- = Verwenden Sie die XML-Schemadefinition db2ha.xsd für          = -->
<!-- = db2haicu, und geben Sie IBM Tivoli System                     = -->
<!-- = Automation for Multiplatforms (SA MP) Base Component          = -->
<!-- = als Cluster-Manager an.                                       = -->
<!-- ================================================================== -->
<DB2Cluster xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
            xsi:noNamespaceSchemaLocation="db2ha.xsd"
            clusterManagerName="TSA"
            version="1.0">

  <!-- ================================================================== -->
  <!-- = Erstellen Sie eine Clusterdomäne mit dem Namen 'db2HAdomain'. = -->
  <!-- ================================================================== -->
  <ClusterDomain domainName="db2HAdomain">

    <!-- ================================================================== -->
    <!-- = Geben Sie eine Netzquorumeinheit (IP-Adresse: 19.126.4.5)   = -->
    <!-- = an. Die IP muss jederzeit von allen Clusterdomänenknoten    = -->
    <!-- = mit Ping überprüft werden können.                           = -->
    <!-- ================================================================== -->
    <Quorum quorumDeviceProtocol="network" quorumDeviceName="19.126.4.5"/>

    <!-- ================================================================== -->
    <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_public_network_0' = -->
    <!-- = mit einem IP-Netzprotokoll.                                 = -->
```

```
  <!-- = Dieses Netz enthält 2 Computer: 'hasys01' und 'hasys02'.    = -->
  <!-- = Jeder Computer hat eine Netzschnittstellenkarte (eth0).     = -->
  <!-- = Die IP-Adresse von 'eth0' auf 'hasys01' ist '19.126.52.139'.= -->
  <!-- = Die IP-Adresse von 'eth0' auf 'hasys02' ist '19.126.52.140'.= -->
  <!-- ============================================================== -->
  <PhysicalNetwork physicalNetworkName="db2_public_network_0"
                   physicalNetworkProtocol="ip">

    <Interface interfaceName="eth0" clusterNodeName="hasys01">
      <IPAddress baseAddress="19.126.52.139"
                 subnetMask="255.255.255.0"
                 networkName="db2_public_network_0"/>
    </Interface>

    <Interface interfaceName="eth0" clusterNodeName="hasys02">
      <IPAddress baseAddress="19.126.52.140"
                 subnetMask="255.255.255.0"
                 networkName="db2_public_network_0"/>
    </Interface>

  </PhysicalNetwork>

  <!-- ============================================================== -->
  <!-- = Erstellen Sie ein Netz mit dem Namen 'db2_private_network_0'= -->
  <!-- = mit einem IP-Netzprotokoll.                                  = -->
  <!-- = Dieses Netz enthält 2 Computer: 'hasys01' und 'hasys02'.    = -->
  <!-- = Zusätzlich zu 'eth0' verfügt jeder Computer über eine       = -->
  <!-- = Netzschnittstellenkarte mit dem Namen 'eth1'.                = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys01' ist 192.168.23.101. = -->
  <!-- = Die IP-Adresse von 'eth1' auf 'hasys02' ist 192.168.23.102. = -->
  <!-- ============================================================== -->
  <PhysicalNetwork physicalNetworkName="db2_private_network_0"
                   physicalNetworkProtocol="ip">

    <Interface interfaceName="eth1" clusterNodeName="hasys01">
      <IPAddress baseAddress="192.168.23.101"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

    <Interface interfaceName="eth1" clusterNodeName="hasys02">
      <IPAddress baseAddress="192.168.23.102"
                 subnetMask="255.255.255.0"
                 networkName="db2_private_network_0"/>
    </Interface>

  </PhysicalNetwork>

  <!-- ============================================================== -->
  <!-- = Listen Sie die Computer (Knoten) in der Clusterdomäne auf.  = -->
  <!-- ============================================================== -->
  <ClusterNode clusterNodeName="hasys01"/>
  <ClusterNode clusterNodeName="hasys02"/>

</ClusterDomain>


<!-- ================================================================== -->
<!-- = Die Richtlinie für Funktionsübernahme gibt die Reihenfolge    = -->
<!-- = für die Funktionsübernahme bei den Clusterdomänenknoten an.   = -->
<!-- ================================================================== -->
<FailoverPolicy>
   <HADRFailover />
</FailoverPolicy>


<!-- ================================================================== -->
```

```
  <!-- = Geben Sie alle Details zu den Datenbankpartitionen an.        = -->
  <!-- ================================================================ -->
  <DB2PartitionSet>
    <DB2Partition dbpartitionnum="0" instanceName="db2inst1" />
  </DB2PartitionSet>


  <!-- ================================================================ -->
  <!-- = Liste der HADR-Datenbanken                                     = -->
  <!-- ================================================================ -->
  <HADRDBSet>
    <HADRDB databaseName="HADRDB"
            localInstance="db2inst1"
            remoteInstance="db2inst1"
            localHost="hasys01"
            remoteHost="hasys02" />
</HADRDBSet>

</DB2Cluster>
```

db2haicu - Voraussetzungen

Vor der Verwendung von DB2 High Availability Instance Configuration Utility (db2haicu) müssen bestimmte Tasks ausgeführt werden.

Allgemein

Bevor der Eigner einer Datenbankmanagerinstanz 'db2haicu' ausführen kann, muss ein Benutzer mit Rootberechtigung den Befehl 'preprpnode' ausführen.

'preprpnode' ist Teil der RSCT-Dateigruppe (RSCT = Reliable Scalable Cluster Technology) für AIX und des RSCT-Pakets für Linux. 'preprpnode' führt die Initialisierung der Knoten für die clusterinterne Kommunikation aus. Der Befehl 'preprpnode' wird im Rahmen der Clusterkonfiguration ausgeführt. Weitere Informationen zu 'preprpnode' finden Sie auf den Websites zu folgenden Themen:

- `preprpnode` Command (AIX)
- RSCT for Linux Technical Reference - preprpnode

Weitere Informationen zu RSCT finden Sie unter RSCT Administration Guide - What is RSCT?

Vor der Ausführung von db2haicu muss der Eigner einer Datenbankmanagerinstanz die folgenden Tasks ausführen:

- Servicedateien auf allen Maschinen synchronisieren, die dem Cluster hinzugefügt werden.
- Script `db2profile` für die Datenbankmanagerinstanz ausführen, die zur Erstellung der Clusterdomäne verwendet wird.
- Datenbankmanager mit dem Befehl db2start starten.

DB2 High Availability Disaster Recovery (HADR)

Bei Verwendung der HADR-Funktion müssen Sie die folgenden Tasks ausführen:

- Stellen Sie sicher, dass alle DB2 HADR-Datenbanken in den entsprechenden Primär- und Bereitschaftsdatenbankrollen gestartet wurden und sich alle HADR-Datenbankpaare (Primär- und Bereitschaftsdatenbanken) im Peerstatus befinden.
- Konfigurieren Sie **hadr_peer_window** für alle HADR-Datenbanken mit einem Wert von mindestens 120 Sekunden.
- Inaktivieren Sie den DB2-Fehlermonitor.

Umgebung mit partitionierter Datenbank

Führen Sie die folgenden Schritte aus, wenn Sie mehrere Datenbankpartitionen für hohe Verfügbarkeit konfigurieren müssen:

- Konfigurieren Sie die Registrierdatenbankvariable DB2_NUM_FAILOVER_NODES auf allen Maschinen, die der Clusterdomäne hinzugefügt werden.
- (Optional) Aktivieren Sie die Datenbank vor der Ausführung von db2haicu.

Erstellen einer Clusterdomäne mit db2haicu

Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) zum ersten Mal für eine Datenbankmanagerinstanz ausführen, erstellt db2haicu ein Modell des Clusters, das als *Clusterdomäne* bezeichnet wird.

Von db2haicu automatisch erkannte Datenbankpfade:

Wenn Sie DB2 High Availability Instance Configuration Utility (db2haicu) zum ersten Mal ausführen, sucht db2haicu im Datenbanksystem nach Informationen zur Datenbankkonfiguration, die sich auf die Clusterkonfiguration beziehen.

Umgebung mit Einzelpartitionsdatenbank

In einer Umgebung mit einer Einzelpartitionsdatenbank erkennt db2haicu automatisch folgende Pfade:

- Ausgangsverzeichnispfad der Instanz
- Prüfprotokollpfad
- Protokollpfad des Prüfarchivs
- Protokollpfad des Synchronisationspunktmanagers (SPM)
- Protokollpfad der DB2-Diagnoseprogramme (db2diaglog)
- Datenbankspezifische Pfade:
  - Datenbankprotokollpfad
  - Pfad für Tabellenspeicherbereichscontainer der Datenbank
  - Pfad für das Tabellenspeicherbereichsverzeichnis der Datenbank
  - Lokales Datenbankverzeichnis

Umgebung mit mehreren Datenbankpartitionen

In einer Umgebung mit mehreren Datenbankpartitionen erkennt db2haicu lediglich die folgenden Pfade automatisch:

- Datenbankprotokollpfad
- Pfad für Tabellenspeicherbereichscontainer der Datenbank
- Pfad für das Tabellenspeicherbereichsverzeichnis der Datenbank
- Lokales Datenbankverzeichnis

Verwalten einer Clusterdomäne mit db2haicu

Wenn Sie das Clusterdomänenmodell der Clusterumgebung mit db2haicu modifizieren, gibt der Datenbankmanager die entsprechenden Änderungen an die Datenbankmanagerinstanz und die Clusterkonfiguration weiter.

Vorbereitung

Vor der Konfiguration der Clusterumgebung mit db2haicu müssen Sie eine Clusterdomäne erstellen und konfigurieren. Weitere Informationen hierzu finden Sie in „Erstellen einer Clusterdomäne mit db2haicu" auf Seite 136.

Informationen zu dieser Task

Zu den Wartungsaufgaben von db2haicu gehört das Hinzufügen von Datenbank- oder Clusterelementen wie beispielsweise Datenbanken oder Clusterknoten zur Clusterdomäne sowie das Entfernen von Elementen aus der Clusterdomäne. Ferner gehört zu den Wartungsaufgaben von db2haicu das Modifizieren der Details von Clusterdomänenelementen wie der Richtlinie für Funktionsübernahme für die Datenbankmanagerinstanz.

Vorgehensweise

1. Führen Sie db2haicu aus.

   Bei der Ausführung von db2haicu im Wartungsmodus zeigt db2haicu eine Liste der Operationen an, die Sie für die Clusterdomäne ausführen können:

   - Fügen Sie Clusterknoten hinzu, oder entfernen Sie sie. (Die Maschine wird durch den Hostnamen identifiziert.)
   - Fügen Sie eine Netzschnittstelle (Netzschnittstellenkarte) hinzu, oder entfernen Sie sie.
   - Fügen Sie DB2-Datenbankpartitionen hinzu, oder entfernen Sie sie. (Nur in Umgebungen mit partitionierten Datenbanken.)
   - Fügen Sie HADR-Datenbanken hinzu, oder entfernen Sie sie.
   - Fügen Sie eine Hochverfügbarkeitsdatenbank hinzu, oder entfernen Sie sie.
   - Fügen Sie einen Mountpunkt hinzu, oder entfernen Sie ihn.
   - Fügen Sie eine IP-Adresse hinzu, oder entfernen Sie sie.
   - Fügen Sie einen nicht kritischen Pfad hinzu, oder entfernen Sie ihn.
   - Versetzen Sie DB2-Datenbankpartitionen und HADR-Datenbanken für die planmäßige Wartung.
   - Ändern Sie die Richtlinie für Funktionsübernahme für diese Instanz.
   - Erstellen Sie eine neue Quorumeinheit für die Domäne.
   - Löschen Sie die Domäne.
2. Wählen Sie eine Task aus, und beantworten Sie die Fragen, die Ihnen von db2haicu gestellt werden.

Ergebnisse

Der Datenbankmanager verwendet die Informationen in der Clusterdomäne für die Koordination mit dem Cluster-Manager. Wenn Sie Datenbank- und Clusterelemente mit db2haicu konfigurieren, werden diese Elemente in die integrierte und automatisierte Clusterkonfiguration und -verwaltung von DB2 High Availability Feature eingefügt. Wenn Sie mit db2haicu eine Konfigurationsänderung der Datenbankmanagerinstanz vornehmen, führt der Datenbankmanager die erforderliche Konfigurationsänderung für den Cluster-Manager für Sie durch, damit kein nachfolgender Aufruf an den Cluster-Manager erforderlich ist.

Weitere Schritte

DB2 High Availability Instance Configuration Utility (db2haicu) verfügt über kein separates Diagnoseprotokoll. Fehler bei db2haicu können mit dem Diagnoseproto-

koll db2diag.log des Datenbankmanagers und dem Tool db2pd untersucht und diagnostiziert werden. Weitere Informationen hierzu finden Sie in „Fehler bei db2haicu beheben".

Fehler bei db2haicu beheben

DB2 High Availability Instance Configuration Utility (db2haicu) verfügt über kein separates Diagnoseprotokoll. Fehler bei db2haicu können mit dem Diagnoseprotokoll db2diag.log des Datenbankmanagers und dem Tool db2pd untersucht und diagnostiziert werden.

db2haicu - Einschränkungen

Für die Verwendung von DB2 High Availability Instance Configuration Utility (db2haicu) gelten bestimmte Einschränkungen.

- „Software und Hardware"
- „Konfigurationstasks"
- „Hinweise zur Verwendung"
- „Empfehlungen" auf Seite 139

Software und Hardware

- IBM Tivoli System Automation for Multiplatforms (SA MP) ist der einzige Cluster-Manager, der von db2haicu unterstützt wird. Eine Liste der Betriebssysteme und Hardware, die von SA MP Base Component unterstützt werden, finden Sie in „Unterstützte Software und Hardware für IBM Tivoli SA MP Base Component" auf Seite 94.
- db2haicu unterstützt IP-Version 6 nicht.

Konfigurationstasks

Die folgenden Tasks können mit db2haicu nicht ausgeführt werden:

- Mit db2haicu kann keine automatische Clientweiterleitung konfiguriert werden.
- Wenn Sie ein Upgrade von DB2 Database für Linux, UNIX und Windows Version 9 auf IBM Data Server Version 9.5 bzw. von Version 9.5 auf eine höhere Version durchführen, können Sie die Clusterkonfiguration mit db2haicu nicht migrieren. Zur Migration einer Clusterkonfiguration müssen Sie die folgenden Schritte ausführen:
  1. Löschen Sie gegebenenfalls die vorhandene Clusterdomäne.
  2. Führen Sie ein Upgrade des Datenbankservers durch.
  3. Erstellen Sie mit db2haicu eine neue Clusterdomäne.

Hinweise zur Verwendung

Beachten Sie bei der Planung von Clusterkonfiguration und Verwaltungsaktivitäten die folgenden Hinweise zur Verwendung von db2haicu:

- Obwohl db2haicu einige Verwaltungstasks ausführt, für die normalerweise eine Rootberechtigung erforderlich ist, wird db2haicu mit den Zugriffsrechten des Eigners der Datenbankmanagerinstanz ausgeführt. Die von einem Root ausgeführte Initialisierung von db2haicu ermöglicht db2haicu die Ausführung der erforderlichen Konfigurationsänderungen, obwohl nur die Zugriffsrechte des Instanzeigners vorliegen.

- Bei der Erstellung einer neuen Clusterdomäne überprüft db2haicu nicht, ob der angegebene Name für die neue Clusterdomäne korrekt ist. db2haicu prüft beispielsweise nicht, ob die Länge des Namens korrekt ist, ob der Name gültige Zeichen enthält oder dem Namen einer vorhandenen Clusterdomäne entspricht.
- db2haicu überprüft bzw. bestätigt keine Informationen, die von einem Benutzer angegeben und an einen Cluster-Manager übermittelt werden. Da db2haicu nicht alle Einschränkungen des Cluster-Managers hinsichtlich der Namen von Clusterobjekten kennen kann, leitet db2haicu beispielsweise Textdaten an den Cluster-Manager weiter, ohne diesen Text auf gültige Zeichen, Textlänge usw. zu überprüfen.
- Wenn ein Fehler auftritt und db2haicu während der Erstellung und Konfiguration einer neuen Clusterdomäne fehlschlägt, müssen Sie die folgenden Schritte ausführen:
  1. Entfernen Sie die Ressourcengruppen der teilweise erstellten Clusterdomäne, indem Sie db2haicu unter Verwendung des Parameters **-delete** ausführen.
  2. Wiederholen Sie die Erstellung der neuen Clusterdomäne, indem Sie db2haicu erneut aufrufen.
- Bei der Ausführung von db2haicu mit dem Parameter **-delete** werden die Ressourcengruppen, die der aktuellen Datenbankmanagerinstanz zugeordnet sind, von db2haicu unverzüglich gelöscht, ohne zu überprüfen, ob diese Ressourcengruppen gesperrt sind.
- Führen Sie die folgenden Schritte aus, um Ressourcengruppen zu entfernen, die den Datenbankmanagerinstanzen eines DB2 HADR-Datenbankpaares (Primär- und Bereitschaftsdatenbank) zugeordnet sind:
  1. Führen Sie db2haicu mit dem Parameter **-delete** zuerst für die Datenbankmanagerinstanz der HADR-Bereitschaftsdatenbank aus.
  2. Führen Sie anschließend db2haicu mit dem Parameter **-delete** für die Datenbankmanagerinstanz der HADR-Primärdatenbank aus.
- Wenn eine Cluster-Operation, die Sie mit db2haicu ausführen möchten, das Zeitlimit überschreitet, gibt db2haicu keinen Fehler zurück. Sie erhalten nur dann Kenntnis davon, dass eine Cluster-Operation das Zeitlimit überschritten hat, wenn Sie nach dem db2haicu-Aufruf die Diagnoseprotokolle überprüfen oder nach dem Fehlschlagen einer nachfolgenden Clusteraktion den Fehler untersuchen und dabei feststellen, dass die ursprüngliche Cluster-Operation das Zeitlimit überschritten hat.
- Wenn Sie versuchen, die Richtlinie für Funktionsübernahme für eine bestimmte Datenbankinstanz in 'Aktiv/Passiv' zu ändern, gibt es eine Bedingung, bei deren Eintreten die Konfigurationsoperation fehlschlägt, für die db2haicu jedoch keinen Fehler zurückgibt. Wenn Sie eine Maschine, die momentan offline ist, als *aktive* Maschine angeben, macht db2haicu diese Maschine nicht zur aktiven Maschine, gibt aber keinen Fehler zurück, der auf das Fehlschlagen der Änderung hinweist.

Empfehlungen

Die folgende Liste enthält Empfehlungen zur Konfiguration des Clusters und der Datenbankmanagerinstanzen bei Verwendung von db2haicu.

- Wenn Sie für den Cluster neue Mountpunkte hinzufügen, indem Sie unter `/etc/fstab` Einträge einfügen, können Sie mit der Option **noauto** verhindern, dass die Mountpunkte automatisch an mehr als eine Maschine im Cluster angehängt werden. Beispiel:

```
dev/vpatha1      /db/svtpdb/NODE0010        ext3  noauto 0 0
```

DB2-Cluster-Manager-API

Die DB2-Cluster-Manager-API definiert eine Reihe von Funktionen, mit deren Hilfe der Datenbankmanager dem Cluster-Manager Konfigurationsänderungen mitteilen kann.

Unterstützte Cluster-Management-Software

Cluster-Management-Software ermöglicht Ihnen die Weiterleitung von DB2-Datenbankoperationen von einer fehlgeschlagenen Primärdatenbank in einem Clusterknoten an eine sekundäre Datenbank in einem anderen Clusterknoten.

Die DB2-Datenbank unterstützt die folgende Cluster-Management-Software:

- High Availability Cluster Multi-Processing (HACMP) für AIX

  Detaillierte Informationen zu HACMP/ES finden Sie im White Paper „IBM DB2 Universal Database Enterprise Edition for AIX and HACMP/ES", das auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/) verfügbar ist.
- Tivoli System Automation für Linux

  Detaillierte Informationen zu Tivoli System Automation finden Sie im White Paper „Highly Available DB2 Universal Database using Tivoli System Automation for Linux", das auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/) verfügbar ist.
- Microsoft Cluster Server für Windows-Betriebssysteme.

  Informationen zu Microsoft Cluster Server finden Sie im folgenden White Paper, das auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/) verfügbar ist: „Implementing IBM DB2 Universal Database V8.1 Enterprise Server Edition with Microsoft Cluster Server".
- Sun Cluster oder VERITAS Cluster Server für das Solaris-Betriebssystem

  Informationen zu Sun Cluster finden Sie im White Paper „ DB2 Universal Database and High Availability on Sun Cluster 3.X", das auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/) verfügbar ist. Informationen zu VERITAS Cluster Server finden Sie im White Paper „DB2 UDB and High Availability with VERITAS Cluster Server", das auf der Website „IBM Support and downloads" (unter http://www.ibm.com/support/docview.wss?uid=swg21045033) verfügbar ist.
- Multi-Computer/ServiceGuard für Hewlett-Packard

High Availability Cluster Multi-Processing für AIX

High Availability Cluster Multi-Processing (HACMP) für AIX ist eine Software für das Cluster-Management. Die Knoten in HACMP-Clustern tauschen Nachrichten aus, sog. "Heartbeats" (Überwachungssignale) oder Keepalive-Pakete. Wenn ein Knoten diese Nachrichten nicht mehr sendet, ruft HACMP die Funktionsübernahme (Failover) für die anderen Knoten im Cluster auf, und wenn der fehlgeschlagene Knoten repariert ist, reintegriert ihn HACMP wieder in den Cluster.

Es gibt zwei Ereignistypen: Standardereignisse, die innerhalb der Funktionen von HACMP abgefangen werden, und benutzerdefinierte Ereignisse, die mit der Überwachung von Parametern in Hardware- und Softwarekomponenten verbunden sind.

Eines der Standardereignisse ist das node_down-Ereignis. Dies bedeutet, ein Knoten im Cluster ist fehlgeschlagen, und HACMP hat die Funktionsübernahme für alle anderen Knoten im Cluster initialisiert. Bei der Planung, welche Maßnahmen als Teil des Recoveryprozesses durchzuführen sind, ermöglicht HACMP zwei

Funktionsübernahmeoptionen: den reinen Bereitschaftsmodus (Hot bzw. Idle Standby) und den Modus der gegenseitigen Übernahme (Mutual Takeover).

Anmerkung: Wenn Sie HACMP verwenden, stellen Sie sicher, dass DB2-Instanzen nicht beim Booten des Systems gestartet werden. Verwenden Sie dazu das Dienstprogramm db2iauto wie folgt:

```
db2iauto -off instanzname
```

Dabei gilt Folgendes: *instanzname* ist der Anmeldename der Instanz.

Clusterkonfiguration

In einer Konfiguration im *Bereitschaftsmodus* (Hot Standby) hat der AIX-Prozessorknoten, der als Übernahmeknoten fungiert, *keine andere* Auslastung. Bei einer Konfiguration für die *gegenseitige Funktionsübernahme* hat der AIX-Prozessorknoten, der als Übernahmeknoten fungiert, *andere* Auslastungen.

Allgemein wird die DB2-Datenbank in Umgebungen mit partitionierten Datenbanken im Modus der gegenseitigen Übernahme (Mutual Takeover) mit Datenbankpartitionen auf jedem Knoten ausgeführt. Eine Ausnahme bildet ein Szenario, in dem die Katalogpartition Teil einer Bereitschaftskonfiguration ist.

Ein Planungsaspekt ist die Verwaltung großer Cluster. Die Verwaltung eines kleinen Clusters ist einfacher als die eines großen. Die Verwaltung eines einzelnen großen Clusters ist jedoch immer noch einfacher als die vieler kleiner Cluster. Bei der Planung ist auch zu bedenken, wie die Anwendungen in der Clusterumgebung verwendet werden. Wenn zum Beispiel eine einzige, umfangreiche und homogene Anwendung auf 16 Knoten betrieben wird, ist die Konfiguration als ein Cluster wahrscheinlich einfacher zu verwalten, als sie in acht Cluster mit jeweils zwei Knoten zu zerlegen. Wenn dieselben 16 Knoten viele verschiedene Anwendungen mit verschiedenen Netzwerken, Platten und Knotenbeziehungen enthalten, ist es wahrscheinlich besser, die Knoten in kleinere Cluster zu gruppieren. Beachten Sie, dass Knoten nur nacheinander, d. h., nur einer gleichzeitig, in einen HACMP-Cluster integriert werden. Eine Konfiguration mit mehreren Clustern lässt sich schneller starten als ein einziger großer Cluster. HACMP unterstützt sowohl einzelne als auch mehrere Cluster, vorausgesetzt, ein Knoten und der zugehörige Ausweichknoten befinden sich im selben Cluster.

Die HACMP-Funktion der Recovery durch Funktionsübernahme ermöglicht eine vordefinierte Zuordnung einer Ressourcengruppe zu einem physischen Knoten (auch als *hintereinandergeschaltete Zuordnung* bezeichnet). Die Recovery durch Funktionsübernahme ermöglicht außerdem eine gleitende (auch als *rotierende Zuordnung* bezeichnete) Zuordnung einer Ressourcengruppe zu einem physischen Knoten. IP-Adressen und externe Datenträgergruppen oder Dateisysteme oder NFS-Dateisysteme und Anwendungsserver innerhalb jeder Ressourcengruppe geben entweder eine Anwendung oder eine Anwendungskomponente an, die von HACMP zwischen den physischen Knoten durch Funktionsübernahme und Reintegration beeinflusst werden können. Die Funktionsübernahme und Reintegration wird durch die Art der erstellten Ressourcengruppe und durch die Anzahl der in der Ressourcengruppe angelegten Knoten angegeben.

Betrachten Sie zum Beispiel eine DB2-Datenbankpartition (logischer Knoten). Wenn die Protokoll- und Tabellenbereichscontainer auf externen Platten angelegt und andere Knoten mit diesen Platten verbunden würden, wäre es möglich, dass die anderen Knoten auf diese Platten zugreifen und die Datenbankpartition (auf einem Übernahmeknoten) neu starten. Eben diese Art von Operation wird durch HACMP

automatisiert. HACMP kann außerdem dazu verwendet werden, NFS-Dateisysteme wiederherzustellen, die von Hauptbenutzerverzeichnissen der DB2-Instanz verwendet werden.

Lesen Sie sich im Rahmen Ihrer Planung für die Recovery mit dem DB2-Datenbanksystem in einer Umgebung mit partitionierten Datenbanken die Dokumentation zu HACMP sorgfältig durch. Sie sollten die Handbücher zu Konzepten, Planung, Installation und Verwaltung lesen und anschließend die Recoveryarchitektur für Ihre Umgebung entwerfen. Ermitteln Sie für die einzelnen Subsysteme, die Sie aufgrund bekannter möglicher Fehlerpunkte zur Recovery vorgesehen haben, die benötigten HACMP-Cluster und die Recoveryknoten (entweder im Bereitschaftsmodus oder im Modus für gegenseitige Übernahme).

Es ist ausdrücklich zu empfehlen, sowohl Platten als auch Adapter in der externen Plattenkonfiguration zu spiegeln. Bei physischen DB2-Knoten, die für HACMP konfiguriert sind, muss sorgfältig sichergestellt werden, dass die Knoten in der Datenträgergruppe von den gemeinsamen externen Platten abweichen können. In einer Konfiguration für gegenseitige Übernahme macht diese Anordnung eine zusätzliche Planung erforderlich, sodass die zu Paaren verbundenen Knoten ohne Konflikt auf die Datenträgergruppen des jeweils anderen Knotens zugreifen können. In einer Umgebung mit partitionierten Datenbanken bedeutet dies, dass alle Containernamen über alle Datenbanken hinweg eindeutig sein müssen.

Eine Methode zur Sicherstellung dieser Eindeutigkeit besteht darin, die Datenbankpartitionsnummer als Teil des Namens zu verwenden. Sie können einen Knotenausdruck für die Syntax von Containerzeichenfolgen angeben, wenn Sie SMS- oder DMS-Container erstellen. Wenn Sie den Ausdruck angeben, kann die Knotennummer Teil des Containernamens sein, oder, wenn Sie zusätzliche Argumente angeben, können die Ergebnisse dieser Argumente Teil des Containernamens sein. Verwenden Sie das Argument " $N" (blank]$N), um den Knotenausdruck anzugeben. Das Argument muss an das Ende der Containerzeichenfolge gesetzt werden und kann nur in einem der folgenden Formate verwendet werden:

Tabelle 7. Argumente zur Containererstellung.. Als Knotennummer wird fünf (5) angenommen.

Syntax	Beispiel	Wert
blank]$N	" $N"	5
blank]$N+ number]	" $N+1011"	1016
blank]$N% number]	" $N%3"	2
blank]$N+ number]% number]	" $N+12%13"	4
blank]$N% number]+ number]	" $N%3+20"	22
Anmerkung: 1. % bedeutet Modulus. 2. In allen Fällen werden die Operatoren von links nach rechts ausgewertet.		

Es folgen einige Beispiele zur Erstellung von Containern mithilfe dieses speziellen Arguments:

- Erstellen von Containern zur Verwendung auf einem Zweiknotensystem

```
CREATE TABLESPACE TS1 MANAGED BY DATABASE USING
        (device '/dev/rcont $N' 20000)
```

  Die folgenden Container würden verwendet:

```
/dev/rcont0   - auf Knoten 0
/dev/rcont1   - auf Knoten 1
```

- Erstellen von Containern zur Verwendung auf einem Vierknotensystem

```
CREATE TABLESPACE TS2 MANAGED BY DATABASE USING
        (file '/DB2/containers/TS2/container $N+100' 10000)
```

  Die folgenden Container würden verwendet:

```
/DB2/containers/TS2/container100   - auf Knoten 0
/DB2/containers/TS2/container101   - auf Knoten 1
/DB2/containers/TS2/container102   - auf Knoten 2
/DB2/containers/TS2/container103   - auf Knoten 3
```

- Erstellen von Containern zur Verwendung auf einem Zweiknotensystem

```
CREATE TABLESPACE TS3 MANAGED BY SYSTEM USING
        ('/TS3/cont $N%2, '/TS3/cont $N%2+2')
```

  Die folgenden Container würden verwendet:

```
/TS3/cont0   - auf Knoten 0
/TS3/cont2   - auf Knoten 0
/TS3/cont1   - auf Knoten 1
/TS3/cont3   - auf Knoten 1
```

Konfigurieren von DB2-Datenbankpartitionen für HACMP

Nach der Konfiguration werden die Datenbankpartitionen in einer Instanz durch HACMP nacheinander, ein physischer Knoten nach dem anderen, gestartet. Mehrere Cluster werden empfohlen, wenn parallele DB2-Konfigurationen mit mehr als vier Knoten gestartet werden sollen. Beachten Sie, dass es in einer parallelen DB2-Konfiguration mit 64 Knoten schneller ist, 32 HACMP-Cluster mit zwei Knoten als vier Cluster mit 16 Knoten parallel zu starten.

Eine Scriptdatei gehört zum Lieferumfang von DB2 Enterprise Server Edition, um Hilfestellung bei der Konfiguration für HACMP-Funktionsübernahme bzw. -Recovery auf Bereitschaftsknoten oder Knoten mit gegenseitiger Übernahme zu geben. Die Scriptdatei hat den Namen `rc.db2pe.ee` für einen Einzelknoten und `rc.db2pe.eee` für mehrere Knoten. Diese Dateien befinden sich im Verzeichnis `sqllib/samples/hacmp/es`. Kopieren Sie die entsprechende Datei in das Verzeichnis `/usr/bin` auf jedem System im HACMP-Cluster und benennen Sie sie in `rc.db2pe` um.

Zusätzlich können aus `rc.db2pe` heraus DB2-Pufferpoolgrößen während der Funktionsübernahme bei Konfigurationen mit gegenseitiger Übernahme angepasst werden. (Pufferpoolgrößen können konfiguriert werden, um eine ordnungsgemäße Ressourcenzuordnung sicherzustellen, wenn zwei Datenbankpartitionen auf demselben physischen Knoten ausgeführt werden.)

HACMP-Ereignisüberwachung und benutzerdefinierte Ereignisse

Ein Beispiel für ein benutzerdefiniertes Ereignis ist das Einleiten einer Funktionsübernahmeoperation, wenn auf einem bestimmten Knoten ein Prozess abgebrochen wird. Ereignisse müssen im Rahmen der Clusterkonfiguration manuell als benutzerdefinierte Ereignisse konfiguriert werden.

Ausführliche Informationen zur Implementierung und zum Design von hoch verfügbaren IBM DB2-Datenbankumgebungen finden Sie auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/).

IBM Tivoli System Automation for Multiplatforms (Linux und AIX)

IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist eine Software für das Cluster-Management, mit der die automatische Verlegung von Benutzern, Anwendungen und Daten von einem Datenbanksystem zu einem anderen innerhalb eines Clusters möglich wird. SA MP Base Component automatisiert die Steuerung von IT-Ressourcen wie z. B. Prozessen, Dateisystemen und IP-Adressen.

SA MP Base Component stellt ein Framework zur automatischen Verwaltung der Verfügbarkeit so genannter Ressourcen bereit. Ressourcen sind zum Beispiel:

- Jede Softwarekomponente, zu deren Steuerung Start-, Überwachungs- und Stoppscripts geschrieben werden können.
- Jede beliebige Netzschnittstellenkarte (NIC), auf die SA MP Base Component Zugriff erteilt wurde. Das heißt, SA MP Base Component verwaltet die Verfügbarkeit jeder IP-Adresse, die ein Benutzer verwenden möchte, durch eine fließende Zuordnung dieser IP-Adresse unter den Netzschnittstellenkarten, auf die Zugriff besteht.

Zum Beispiel haben sowohl eine DB2-Instanz als auch die High Availability Disaster Recovery-Funktion Start-, Stopp- und Überwachungsbefehle. Dies ermöglicht das Schreiben von Tivoli SA MP-Scripts, um die Verwaltung dieser Ressourcen zu automatisieren. Ressourcen, die eng zusammengehören (z. B. solche, die gleichzeitig zusammen auf demselben Knoten ausgeführt werden), werden als *Ressourcengruppe* bezeichnet.

Vor der Installation von SA MP Base Component durch das Installationsprogramm von DB2 Version 9.5 müssen die folgenden RSCT-Voraussetzungen für AIX-Betriebssysteme erfüllt sein:

Tabelle 8. RSCT-Voraussetzungen für AIX-Betriebssysteme

Version von IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component	**RSCT-Version**	**RSCT-APAR-Nummer**
SA MP Base Component 2.2 Fixpack 3 im Produktpaket mit DB2 Version 9.5 GA	2.3.11.1 (AIX 5.2)	IY94673 (AIX 5.2)
	2.4.7.1 (AIX 5.3)	IY94672 (AIX 5.3)
SA MP Base Component 2.2 Fixpack 5 im Produktpaket mit DB2 Version 9.5 Fixpack 1 und Fixpack 2	2.3.12.2 (AIX 5.2)	IZ08646 (AIX 5.2)
	2.4.8.2 (AIX 5.3)	IZ08645 (AIX 5.3)
	2.5.0.2 (AIX 6.1)	IZ08643 (AIX 6.1)
SA MP Base Component 2.2 Fixpack 7 im Produktpaket mit DB2 Version 9.5 Fixpack 3, Fixpack 4 und Fixpack 5	2.3.12.3 (AIX 5.2)	IZ12977 (AIX 5.2)
	2.4.9.2 (AIX 5.3)	IZ21567 (AIX 5.3)
	2.5.1.2 (AIX 6.1)	IZ21568 (AIX 6.1)
SA MP Base Component 3.2 im Produktpaket mit DB2 Version 9.5 Fixpack 6 und späteren Fixpacks	2.4.13.1 (AIX 5.3)	
	3.1.0.0 (AIX 6.1 und AIX 7.1)	

DB2-Ressourcen

In einer DB2-Umgebung mit nur einer Partition wird eine einzelne DB2-Instanz auf einem Server ausgeführt. Diese DB2-Instanz hat lokalen Zugriff auf Daten (auf das eigene ausführbare Image sowie auf Datenbanken, deren Eigner die Instanz ist). Wenn diese DB2-Instanz fernen Clients zugänglich gemacht wird, muss dieser DB2-Instanz eine nicht genutzte IP-Adresse zugewiesen werden.

Die DB2-Instanz, die lokalen Daten und die IP-Adresse werden sämtlich als Ressourcen betrachtet, die durch SA MP Base Component automatisiert werden müssen. Da diese Ressourcen eng zusammengehören (z. B. werden sie gleichzeitig zusammen auf demselben Knoten ausgeführt), werden Sie als Ressourcengruppe bezeichnet.

Die gesamte Ressourcengruppe befindet sich auf einem Knoten im Cluster. Im Fall einer Funktionsübernahme wird die gesamte Ressourcengruppe auf einem anderen Knoten gestartet.

Die folgenden Abhängigkeiten bestehen zwischen den Ressourcen in der Gruppe:

- Die DB2-Instanz muss nach der lokalen Platte gestartet werden.
- Die DB2-Instanz muss vor der lokalen Platte gestoppt werden.
- Die HA-IP-Adresse muss mit der Instanz zusammengelegt werden.

Plattenspeicher

Die DB2-Datenbank kann die folgenden Ressourcen als lokalen Datenspeicher nutzen:

- Rohplatteneinheit (Beispiel: `/dev/sda1`)
- Logischer, durch Logical Volume Manager (LVM) verwalteter Datenträger
- Dateisystem (Beispiele: ext3, jfs)

DB2-Daten können entweder insgesamt auf einer oder mehreren Rohplatteneinheiten, insgesamt auf logischen Datenträgern, insgesamt in Dateisystemen oder in einer Mischung als allen drei Möglichkeiten gespeichert werden. Alle ausführbaren Dateien müssen sich in einer Art Dateisystem befinden.

DB2-Datenbankanforderungen für die HA-IP-Adresse

Die DB2-Datenbank hat keine besonderen Anforderungen im Hinblick auf die IP-Adresse. Es ist nicht erforderlich, eine hoch verfügbare IP-Adresse zu definieren, damit die Instanz als hoch verfügbar betrachtet wird. Jedoch ist es wichtig zu beachten, dass die IP-Adresse, die geschützt wird (sofern vorhanden), den Zugriffspunkt des Clients auf die Daten darstellt, sodass diese Adresse als solche allen Clients bekannt sein muss. Für die Praxis ist zu empfehlen, dass diese IP-Adresse die Adresse ist, die von den Clients in ihren Befehlen CATALOG TCPIP NODE verwendet wird.

Tivoli SA MP-Ressourcengruppen

IBM Tivoli System Automation for Multiplatforms (SA MP) Base Component ist ein Produkt, das eine hohe Verfügbarkeit realisiert, indem es Ressourcen wie Prozesse, Anwendungen, IP-Adressen und andere in Linux-basierten Clustern automatisiert. Zur Automatisierung einer IT-Ressource (wie zum Beispiel einer IP-Adresse) muss die Ressource in SA MP Base Component definiert werden. Darüber hinaus müs-

sen alle diese Ressourcen in mindestens einer Ressourcengruppe enthalten sein. Wenn diese Ressourcen stets auf derselben Hostmaschine aktiv sein müssen, sollten sie alle in dieselbe Ressourcengruppe eingefügt werden.

Jede Anwendung muss als Ressource definiert werden, um mit SA MP Base Component verwaltet und automatisiert werden zu können. Anwendungsressourcen werden in der Regel in der generischen Ressourcenklasse IBM.Application definiert. In dieser Ressourcenklasse sind verschiedene Attribute zur Definition einer Ressource enthalten. Mindestens drei von ihnen sind jedoch anwendungsspezifisch:

- StartCommand
- StopCommand
- MonitorCommand

Diese Befehle können Scripts oder ausführbare Binärdateien sein.

Einrichten von Tivoli SA MP Base Component in der DB2-Umgebung

Wenn Sie Konfigurationsinformationen benötigen, die Ihnen bei der Einrichtung von SA MP Base Component in Ihrer DB2-Umgebung helfen, lesen Sie den Abschnitt „Konfigurieren einer Clusterumgebung für hohe Verfügbarkeit" auf Seite 74 oder suchen Sie auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/) nach "Tivoli System Automation".

Verwenden von Tivoli SA MP-Befehlen mit DB2 High Availability Feature

Wenn Sie DB2 High Availability Feature verwenden, können viele DB2-Befehle die erforderlichen Cluster-Manager-Konfigurationen automatisch für Sie übernehmen, sofern die Instanz mit dem Befehl db2haicu für hohe Verfügbarkeit konfiguriert wurde.

Wenn Sie in diesen Umgebungen beispielsweise den Befehl db2stop einleiten, wird die Ressourcengruppe gesperrt, um zu verhindern, dass die Ressource von Tivoli SA MP wieder in den Onlinestatus versetzt wird. Ebenso können Sie bei einer Übernahmeoperation in einem normalen HADR-Übernahmeszenario den DB2-Befehl TAKEOVER HADR verwenden; der Datenbankmanager nimmt dann automatisch die entsprechenden Cluster-Manager-Konfigurationen vor, damit die HADR-Bereitschaftsdatenbank die Funktion einer HADR-Primärdatenbank übernehmen kann.

Anmerkung: Alternativ kann eine Übernahme auch mit dem SA MP-Befehl rgreq -o move erfolgen; dieser Befehl führt jedoch eine erzwungene Übernahmeoperation aus. Verwenden Sie im Rahmen eines Wartungsszenarios, in dem eine normale, nicht erzwungene Übernahme stattfinden soll, den DB2-Befehl TAKEOVER HADR.

Eine Liste der Konfigurations- und Verwaltungsoperationen der DB2-Datenbankmanagerinstanz, die eine entsprechende Cluster-Manager-Konfiguration über SA MP vornehmen, finden Sie in „Automatisches Konfigurieren eines Clusters mit DB2 High Availability Feature" auf Seite 96.

Unterstützung für Microsoft Failover Clustering (Windows)

Microsoft Failover Clustering (Funktionsübernahmeclustering) unterstützt Server-Cluster unter Windows-Betriebssystemen. Dieses Feature erkennt Server- oder Anwendungsfehler automatisch und reagiert entsprechend; darüber hinaus kann die Auslastung der Server ausgeglichen werden.

Einführung

Microsoft Failover Clustering ist ein Feature von Windows-Serverbetriebssystemen. Diese Software unterstützt die Verbindung zweier Server (bis zu vier Servern in DataCenter Server) in einem Cluster für hohe Verfügbarkeit sowie eine einfachere Verwaltung von Daten und Anwendungen. Failover Clustering stellt automatisch Server- und Anwendungsfehler bzw. -ausfälle fest und führt eine Recovery durch. Mithilfe von Failover Clustering können Serverauslastungen verlagert werden, um eine gleichmäßige Maschinennutzung zu gewährleisten und geplante Wartungsarbeiten ohne Ausfallzeiten durchführen zu können.

Die folgenden DB2-Produkte stellen Unterstützung für Microsoft Failover Clustering bereit:

- DB2 Connect-Serverprodukte (DB2 Connect Enterprise Edition, DB2 Connect Application Server Edition, DB2 Connect Unlimited Edition für iSeries und DB2 Connect Unlimited Edition für zSeries).
- DB2 Enterprise Server Edition
- DB2 Express-C (Befristete Lizenz)
- DB2 Express Edition
- DB2 Workgroup Server Edition

DB2 Failover Clustering-Komponenten

Ein Cluster ist eine Konfiguration mit mindestens zwei Knoten, die jeweils unabhängige Computersysteme sind. Der Cluster erscheint Netzclients wie ein einzelner Server.

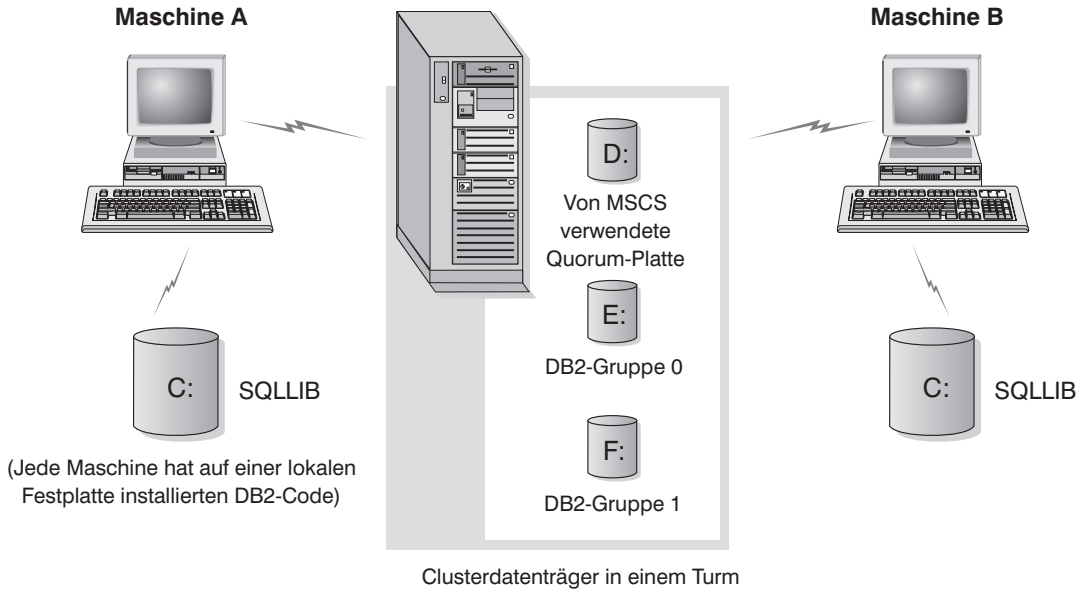


Abbildung 3. Beispielkonfiguration für Failover Clustering

Die Knoten in einem Failover Clustering-Cluster werden durch mindestens einen gemeinsam genutzten Speicherbus und mindestens ein physisch unabhängiges Netzwerk miteinander verbunden. Das Netz, das nur die Server miteinander verbindet, nicht jedoch die Clients mit dem Cluster, wird als *privates* Netz bezeichnet. Das Netz, das die Clientverbindungen unterstützt, wird als *öffentliches* Netz bezeichnet. Auf jedem Knoten befindet sich mindestens eine lokale Platte. Jeder ge-

meinsam genutzte Speicherbus ist mit mindestens einer Platte verbunden. Für jede Platte im gemeinsam genutzten Bus liegt das Eigentumsrecht jeweils nur bei einem Clusterknoten. Die DB2-Software befindet sich auf der lokalen Platte. Die DB2-Datenbankdateien (z. B. Tabellen, Indizes, Protokolldateien) befinden sich auf gemeinsam genutzten Platten. Da Microsoft Failover Clustering die Verwendung von unformatierten Partitionen in einem Cluster nicht unterstützt, kann DB2 in Microsoft Failover Clustering-Umgebungen nicht zur Verwendung von Roheinheiten konfiguriert werden.

Die DB2-Ressource

In einer Microsoft Failover Clustering-Umgebung ist eine Ressource eine Entität, die von der Clustersoftware verwaltet wird. Beispielsweise kann eine Platte, eine IP-Adresse oder ein generischer Service als Ressource verwaltet werden. Zur Integration von Microsoft Failover Clustering erstellt DB2 einen eigenen Ressourcentyp mit dem Namen 'DB2 Server'. Jede DB2-Serverressource verwaltet eine DB2-Instanz. Bei Ausführung in einer Umgebung mit partitionierten Datenbanken verwaltet jede DB2-Serverressource eine Datenbankpartition. Der Name der DB2-Serverressource ist der Instanzname. In Umgebungen mit partitionierten Datenbanken setzt sich der Name der DB2-Serverressource jedoch aus dem Instanznamen und der Nummer der Datenbankpartition (bzw. des Knotens) zusammen.

Prä-Online- und Post-Online-Scripts

Sie können Scripts ausführen, bevor eine DB2-Ressource in den Onlinestatus versetzt wird und nachdem sie in den Onlinestatus versetzt wurde. Diese Scripts werden als Prä-Online-Scripts bzw. Post-Online-Scripts bezeichnet. Diese Scripts sind BAT-Dateien, die DB2- und Systembefehle ausführen können.

In Situationen, in denen möglicherweise mehrere DB2-Instanzen auf derselben Maschine ausgeführt werden, können Sie mithilfe der Prä- und Post-Online-Scripts die Konfiguration anpassen, sodass beide Instanzen erfolgreich gestartet werden können. Wenn eine Funktionsübernahme stattfindet, können Sie mit dem Post-Online-Script eine manuelle Datenbankrecovery ausführen. Ein Post-Online-Script kann auch zum Starten von Anwendungen oder Services verwendet werden, die von DB2 abhängig sind.

Die DB2-Gruppe

Zugehörige oder abhängige Ressourcen werden in Ressourcengruppen organisiert. Alle in einer Gruppe vorhandenen Ressourcen werden zwischen Clusterknoten als eine Einheit versetzt. In einer typischen DB2-Clusterumgebung mit einer einzelnen Partition ist beispielsweise eine DB2-Gruppe vorhanden, die die folgenden Ressourcen enthält:

1. Ressource 'DB2'. Die Ressource 'DB2' verwaltet die DB2-Instanz (oder den Knoten).
2. Ressource 'IP-Adresse'. Die Ressource 'IP-Adresse' ermöglicht es Clientanwendungen, eine Verbindung zum DB2-Server herzustellen.
3. Ressource 'Netzwerkname'. Die Ressource 'Netzwerkname' ermöglicht es Clientanwendungen, unter Verwendung eines Namens anstatt einer IP-Adresse eine Verbindung zum DB2-Server herzustellen. Für die Ressource 'Netzwerkname' besteht eine Abhängigkeit zur Ressource 'IP-Adresse'. Die Ressource 'Netzwerkname' ist optional. (Das Konfigurieren einer Ressource 'Netzwerkname' kann sich auf die Leistung der Funktionsübernahme auswirken.)

4. Mindestens eine Ressource 'Physische Platte'. Jede Ressource 'Physische Platte' verwaltet eine gemeinsam genutzte Platte im Cluster.

Anmerkung: Die Ressource 'DB2' ist so konfiguriert, dass sie von allen anderen Ressourcen in derselben Gruppe abhängt, sodass der DB2-Server nur dann gestartet werden kann, wenn alle anderen Ressourcen online sind.

Konfigurationen für Funktionsübernahme

Es sind zwei Konfigurationstypen verfügbar:

- Bereitschaftsmodus
- Gegenseitige Übernahme

In einer Umgebung mit partitionierten Datenbanken verfügen nicht alle Cluster über denselben Konfigurationstyp. Sie können Cluster haben, die im Bereitschaftsmodus konfiguriert sind, und andere, die für gegenseitige Übernahme konfiguriert sind. Wenn Ihre DB2-Instanz zum Beispiel aus fünf Workstations besteht, können Sie zwei Maschinen zur gegenseitigen Übernahme und zwei im Bereitschaftsmodus konfigurieren und eine Maschine von der Funktionsübernahmekonfiguration ausnehmen.

Konfiguration für Bereitschaftsmodus

In einer Konfiguration für den Bereitschaftsmodus stellt eine Maschine im Microsoft Failover Clustering-Cluster dedizierte Übernahmeunterstützung bereit, die andere Maschine ist Teil des Datenbanksystems. Wenn die zum Datenbanksystem gehörige Maschine ausfällt, wird ihr Datenbankserver auf der Übernahmemaschine gestartet. Wenn Sie in einer Umgebung mit partitionierten Datenbanken mehrere logische Knoten auf einer Maschine betreiben und die Maschine ausfällt, werden die logischen Knoten auf der Übernahmemaschine gestartet. Abb. 4 zeigt ein Beispiel einer Konfiguration für den Bereitschaftsmodus.

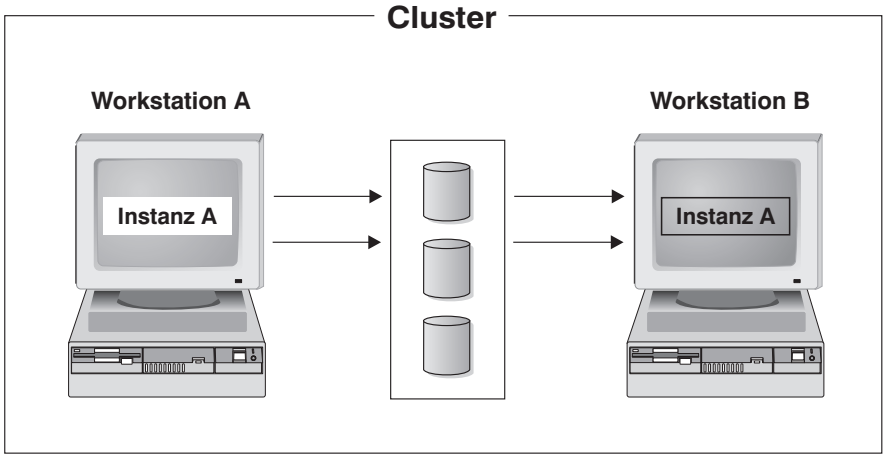


Abbildung 4. Konfiguration für Bereitschaftsmodus

Konfiguration zur gegenseitigen Übernahme

In einer Konfiguration zur gegenseitigen Übernahme sind beide Workstations an einem Datenbanksystem beteiligt (d. h., auf jeder Maschine wird mindestens ein Datenbankserver ausgeführt). Wenn eine der Workstations im Microsoft Failover Clustering-Cluster ausfällt, wird der Datenbankserver der ausgefallenen Maschine

zur Ausführung auf der anderen Maschine gestartet. In einer Konfiguration zur gegenseitigen Übernahme kann ein Datenbankserver auf einer Maschine unabhängig vom Datenbankserver auf einer anderen Maschine ausfallen. Jeder Datenbankserver kann zu einem beliebigen Zeitpunkt auf jeder Maschine aktiv sein. Abb. 5 zeigt ein Beispiel für die Konfiguration zur gegenseitigen Übernahme.

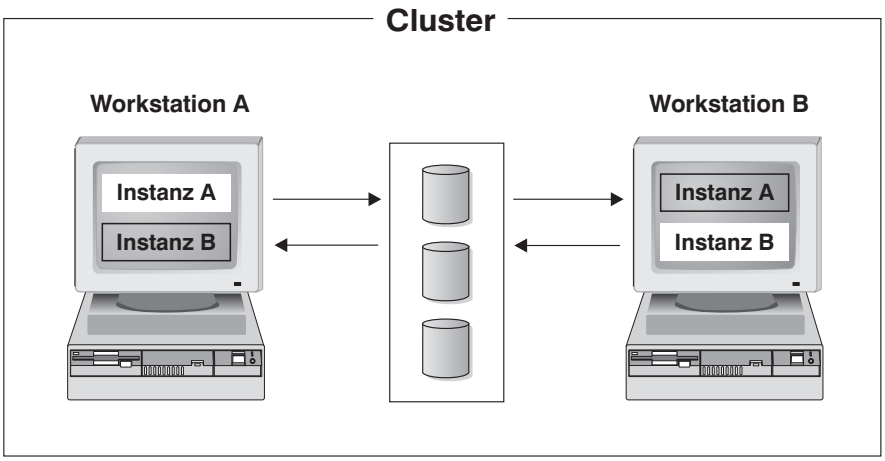


Abbildung 5. Konfiguration zur gegenseitigen Übernahme

Detaillierte Informationen zu Implementierung und Design von hoch verfügbaren IBM DB2-Datenbankumgebungen unter Windows-Betriebssystemen finden Sie auf der Website "IBM Software Library" (http://www.ibm.com/software/sw-library/).

Unterstützung für Failover Clustering unter Windows Server 2008

Gehen Sie wie folgt vor, um partitionierte DB2-Datenbanksysteme so zu konfigurieren, dass sie auf Windows Server 2008-Clustern zur Funktionsübernahme ausgeführt werden:

1. Folgen Sie den Prozeduren, die im White Paper „Implementing IBM DB2 Universal Database V8.1 Enterprise Server Edition with Microsoft Cluster Server" beschrieben werden, das auf der Website der IBM Softwarebibliothek (http://www.ibm.com/software/sw-library/) verfügbar ist.
2. Aufgrund von Änderungen in der Komponente Failover Clustering von Windows Server 2008 sind möglicherweise die folgenden zusätzlichen Konfigurationsschritte erforderlich:
   - In den Windows Server 2008-Clustern zur Funktionsübernahme wird der Windows-Cluster-Service unter einem speziellen Konto für das lokale System ausgeführt, unter Windows Server 2003 dagegen wird der Windows-Cluster-Service unter einem Administratoraccount ausgeführt. Dies wirkt sich auf die Operationen der DB2-Ressource (`db2server.dll`) aus, die im Kontext des Cluster-Service-Kontos ausgeführt wird.

     Wenn in Umgebungen mit partitionierten Datenbanken die Registrierdatenbankvariable **DB2_EXTSECURITY** in einem Windows-Cluster zur Funktionsübernahme auf YES gesetzt wird, müssen die Gruppen DB2ADMNS und DB2USERS Domänengruppen sein.

     Wenn eine Instanz mit mehreren Partitionen in einem Windows-Cluster zur Funktionsübernahme ausgeführt wird, muss der Pfad auf einen Netzpfad gesetzt werden (beispielsweise `\\`*`netzname`*`\DB2MSCS-DB2\DB2PROFS`). Dies erfolgt automatisch, wenn Sie das DB2-Datenbanksystem mit dem Befehl db2mscs zu einem Cluster zusammenfassen.

Bei der Bildung des Windows Server 2008-Clusters zur Funktionsübernahme wird in Active Directory ein Computerobjekt erstellt, das den neuen Cluster darstellt. Wenn beispielsweise der Name des Clusters MYCLUSTER lautet, wird in Active Directory das Computerobjekt MYCLUSTER erstellt. Wenn ein Benutzer eine Instanz mit mehreren Partitionen zu einem Cluster zusammenfasst und die Registrierdatenbankvariable **DB2_EXTSECURITY** auf YES gesetzt wird, muss dieses Computerobjekt zu der Gruppe DB2ADMNS hinzugefügt werden. Sie müssen dies so tun, dass die DB2-Ressourcen-DLL auf den Pfad *netzname*\DB2MSCS-DB2\DB2PROFS zugreifen kann. Wenn beispielsweise die DB2-Administratorgruppe die Bezeichnung MYDOMAIN\DB2ADMNS hat, muss das Computerobjekt MYCLUSTER zu dieser Gruppe hinzugefügt werden. Nach dem Hinzufügen das Computerobjekts zu der Gruppe DB2ADMNS müssen Sie für die beiden Knoten im Cluster einen Warmstart durchführen.

- In Windows Server 2008 Failover Clustering wird die Dateifreigaberessource des Clusters („cluster fileshare resource") nicht weiter unterstützt. Stattdessen wird der Dateiserver des Clusters verwendet. Die Dateifreigabe (eine normale Dateifreigabe) basiert auf der Dateiserverressource des Clusters. Bei Microsoft müssen die im Cluster erstellten Clusterdateiserver für die Namensauflösung DNS (Domain Name System) verwenden. Bei der Ausführung von Instanzen mit mehreren Partitionen ist für die Unterstützung der Dateifreigabe eine Dateiserverressource erforderlich. Die in der Datei db2mscs.cfg angegebenen Parameter **NETNAME_NAME**, **NETNAME_VALUE** und **NETNAME_DEPENDENCY** werden zum Erstellen der Dateiserver- und Dateifreigaberessourcen verwendet. Der *netzname* basiert auf einer IP-Adresse und dieser *netzname* muss in DNS vorliegen. Beispiel: Wenn die Datei db2mscs.cfg die folgenden Parameter enthält, wird eine Dateifreigabe namens \\MSCSV\DB2MSCS-DB2 erstellt:

  ```
  ...
  NETNAME_NAME = MSCSN
  NETNAME_VALUE = MSCSV
  ...
  ```

  Der MSCSV-Name muss in DNS registriert werden. Andernfalls schlägt der Dateiserver oder die Dateifreigabe, der bzw. die für den DB2-Cluster erstellt wurde, fehl, wenn die DNS-Auflösung nicht erfolgreich ist.

Unterstützung für Solaris-Betriebssystemcluster

DB2 unterstützt zwei für das Betriebssystem Solaris verfügbare Cluster: Sun Cluster und Veritas Cluster Server (VCS).

Anmerkung: Wenn Sie Sun Cluster 3.0 oder Veritas Cluster Server verwenden, stellen Sie sicher, dass DB2-Instanzen beim Booten nicht gestartet werden. Verwenden Sie dazu das Dienstprogramm db2iauto wie folgt:

```
db2iauto -off instanzname
```

Dabei gilt Folgendes: *instanzname* ist der Anmeldename der Instanz.

Hohe Verfügbarkeit

Computersysteme, die als Host für Datenbankservices dienen, enthalten viele unterschiedliche Komponenten, und jede Komponente besitzt eine für sie ermittelte „mittlere ausfallfreie Zeit" (MTBF - Mean Time Before Failure). Die MTBF ist die durchschnittliche Dauer, die eine Komponente verwendbar bleibt. Die MTBF für ein Qualitätsfestplattenlaufwerk liegt in der Größenordnung von 1 Million Stunden

(annähernd 114 Jahren). Obwohl dies wie ein langer Zeitraum erscheint, ist es wahrscheinlich, dass eine von 200 Festplatten innerhalb eines Zeitraums von sechs Monaten ausfällt.

Wenngleich es eine Reihe von Methoden zur Erhöhung der Verfügbarkeit für einen Datenbankservice gibt, ist die am häufigsten eingesetzte Methode ein HA-Cluster (High Availability-Cluster). Wenn ein Cluster zur Implementierung hoher Verfügbarkeit eingesetzt wird, besteht er aus mindestens zwei Maschinen, einer Gruppe privater Netzwerkschnittstellen, mindestens einer öffentlichen Netzwerkschnittstelle sowie einigen gemeinsam benutzten Platten. Diese spezielle Konfiguration ermöglicht das Versetzen eines Datenbankservice von einer Maschine auf eine andere. Durch das Versetzen des Datenbankservice auf eine andere Maschine im Cluster wird es möglich, den Zugriff auf die zugehörigen Daten weiterhin aufrecht zu erhalten. Das Versetzen eines Datenbankservice von einer Maschine auf eine andere wird als *Funktionsübernahme* bezeichnet (siehe Abb. 6 zur Illustration).

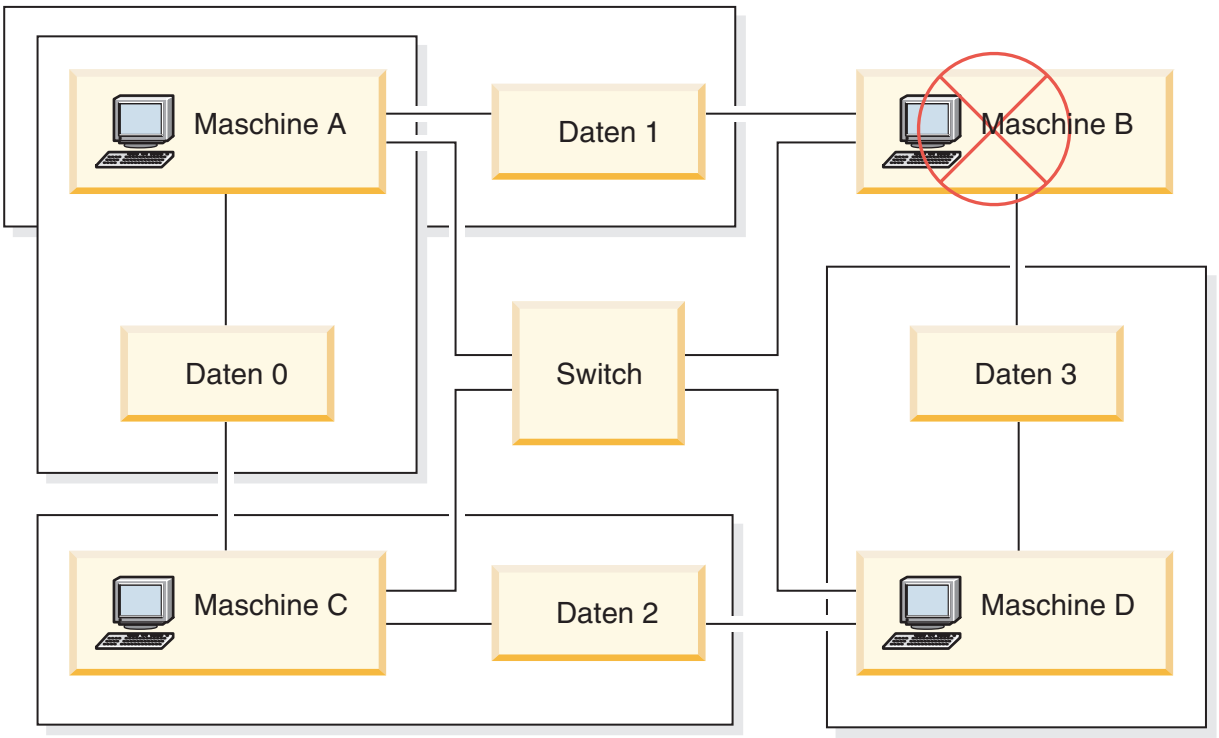


Abbildung 6. Funktionsübernahme. Wenn Maschine B ausfällt, wird ihr Datenbankservice auf eine andere Maschine im Cluster versetzt, sodass weiter auf die Daten zugegriffen werden kann.

Die privaten Netzwerkschnittstellen werden zum Senden von *Überwachungssignal*nachrichten und von Steuernachrichten zwischen den Maschinen im Cluster verwendet. Die öffentlichen Netzwerkschnittstellen dienen zur direkten Kommunikation mit Clients des HA-Clusters. Die Platten in einem HA-Cluster sind mit zwei oder mehr Maschinen im Cluster verbunden, sodass bei Ausfall einer Maschine eine andere Maschine auf sie zugreifen kann.

Ein Datenbankservice, der auf einem HA-Cluster aktiv ist, besitzt mindestens eine öffentliche Netzwerkschnittstelle und eine Gruppe von Platten, die ihm zugeordnet sind. Die Clients eines HA-Datenbankservice stellen eine Verbindung über TCP/IP nur zu den logischen Netzwerkschnittstellen des Datenbankservices her. Wenn es

zu einer Funktionsübernahme kommt, werden der Datenbankservice zusammen mit den zugehörigen Netzwerkschnittstellen und der Gruppe von Platten auf eine andere Maschine versetzt.

Einer der Vorteile eines HA-Clusters liegt darin, dass ein Datenbankservice ohne Hilfe durch Unterstützungspersonal wiederhergestellt werden kann, und dass dies zu jedem beliebigen Zeitpunkt möglich ist. Ein weiterer Vorteil ist die Redundanz. Alle Teile im Cluster sollten redundant sein, einschließlich der Maschinen selbst. Der Cluster sollte in der Lage sein, jeden einzelnen Fehlerpunkt (SPoF, Single Point of Failure) überbrücken zu können.

Auch wenn sich hochverfügbare Datenbankservices in ihrer Art sehr unterscheiden können, besitzen sie einige allgemeine Anforderungen. Clients eines hochverfügbaren Datenbankservice erwarten, dass sich die Netzadresse und der Hostname des Datenbankservices nicht ändern und dass sie die Möglichkeit haben, ihre Anforderungen unabhängig von der Maschine, auf der der Datenbankservice aktiv ist, auf dieselbe Weise zu tätigen.

Betrachten Sie einen Web-Browser, der auf einen hochverfügbaren Web-Server zugreift. Die Anforderung wird mit einer URL-Adresse (Uniform Resource Locator) abgesetzt, die sowohl einen Hostnamen als auch den Pfad zu einer Datei auf dem Web-Server enthält. Der Browser erwartet, dass sowohl der Hostname als auch der Pfad nach einer Übernahme der Web-Server-Funktion gleich bleiben. Wenn der Browser gerade eine Datei vom Web-Server herunterlädt, wenn die Serverfunktion von einer anderen Maschine übernommen wird, muss der Browser die Anforderung erneut absetzen.

Die Verfügbarkeit eines Datenbankservice wird in der Zeitdauer gemessen, die der Datenbankservice seinen Benutzern zur Verfügung steht. Die am häufigsten verwendete Maßeinheit für Verfügbarkeit ist der Prozentsatz der Betriebszeit (Up Time). Diese wird oft als Anzahl von „Neunen" angegeben:

```
99,99% => Service fällt (maximal) 52,6 Minuten / Jahr aus
99,999% => Service fällt (maximal) 5,26 Minuten / Jahr aus
99,9999% => Service fällt (maximal) 31,5 Sekunden / Jahr aus
```

Beachten Sie beim Entwerfen und Testen eines HA-Clusters folgende Punkte:

1. Stellen Sie sicher, dass der Administrator des Clusters mit dem System vertraut ist und weiß, was geschehen soll, wenn eine Funktionsübernahme auftritt.
2. Stellen Sie sicher, dass jeder Teil des Clusters wirklich redundant ist und im Falle eines Ausfalls rasch ausgetauscht werden kann.
3. Erzwingen Sie die Funktionsübernahme eines Testsystems in einer kontrollierten Umgebung, und stellen Sie sicher, dass die Funktionsübernahme jedes Mal ordnungsgemäß vollzogen wird.
4. Protokollieren Sie die Gründe für jede eingetretene Funktionsübernahme. Obwohl dies nicht oft geschehen sollte, ist es wichtig, alle Punkte zu untersuchen, die einen Cluster instabil machen. Wenn zum Beispiel eine Komponente des Clusters fünf Mal in einem Monat eine Funktionsübernahme erforderlich machte, stellen Sie fest, woran dies liegt, und beheben Sie das Problem.
5. Stellen Sie sicher, dass das Unterstützungspersonal für den Cluster benachrichtigt wird, wenn eine Funktionsübernahme eintritt.
6. Überlasten Sie den Cluster nicht. Stellen Sie sicher, dass die verbliebenen Systeme nach einer Funktionsübernahme die Auslastung weiterhin mit einer akzeptablen Leistung bewältigen können.

7. Überprüfen Sie ausfallanfälligere Komponenten (z. B. Festplatten) häufig, sodass sie ersetzt werden können, bevor Probleme auftreten.

Fehlertoleranz

Eine andere Methode zur Erhöhung der Verfügbarkeit eines Datenbankservice ist Fehlertoleranz. Bei einer *fehlertoleranten* Maschine ist sämtliche Redundanz integriert, sodass die Maschine in der Lage ist, einen Einzelausfall einer beliebigen Komponente, einschließlich CPU und Speicher, zu überbrücken. Fehlertolerante Maschinen werden hauptsächlich in Nischenmärkten eingesetzt und ihre Implementierung ist gewöhnlich mit hohen Kosten verbunden. Ein HA-Cluster mit Maschinen an verschiedenen geographischen Standorten hat den zusätzlichen Vorteil, einen Ausfall, der nur einen Teil dieser Standorte betrifft, überbrücken zu können.

Ein HA-Cluster ist die gängigste Lösung zur Erhöhung der Verfügbarkeit, weil sie skalierbar, einfach zu handhaben und relativ kostengünstig in der Implementierung ist.

Unterstützung für Sun Cluster 3.0 (und höher):

Wenn Sie vorhaben, Ihre DB2-Datenbanklösung auf einem Solaris-Betriebssystemcluster auszuführen, können Sie Sun Cluster 3.0 für die Verwaltung des Clusters verwenden. Ein Agent für hohe Verfügbarkeit fungiert als Mediator zwischen der DB2-Datenbank und Sun Cluster 3.0.

Die Anweisungen in diesem Abschnitt zur Unterstützung für Sun Cluster 3.0 gelten auch für höhere Sun Cluster-Versionen.

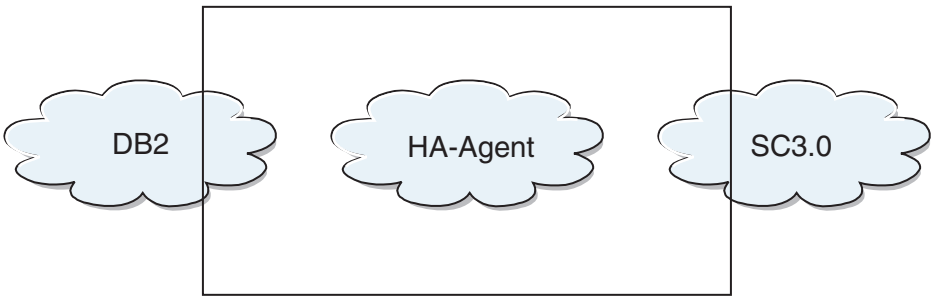


Abbildung 7. DB2, Sun Cluster 3.0 und hohe Verfügbarkeit. Die Beziehungen zwischen der DB2-Datenbank, Sun Cluster 3.0 und dem Agenten für hohe Verfügbarkeit.

Funktionsübernahme

Sun Cluster 3.0 bietet hohe Verfügbarkeit mittels Funktionsübernahme für Anwendungen. Jeder Knoten wird regelmäßig überwacht, und die Clustersoftware verlagert eine clustersensitive Anwendung automatisch von einem fehlgeschlagenen primären Knoten auf einen angegebenen sekundären Knoten. Bei einer Funktionsübernahme stellen Clients möglicherweise eine kurze Unterbrechung des Service fest und müssen die Verbindung zum Server wiederherstellen. Die Tatsache, dass sie nun auf einem anderen physischen Server auf die Anwendungen und Daten zugreifen, hat jedoch keine Auswirkungen. Dadurch, dass bei einem Ausfall des primären Servers automatisch andere Knoten in einem Cluster die Arbeitslast übernehmen, sorgt Sun Cluster 3.0 für eine erhebliche Reduzierung der Ausfallzeiten und eine spürbare Steigerung der Produktivität.

Mehrhostplatten

Sun Cluster 3.0 erfordert Mehrhostplattenspeicher. Das bedeutet, dass Platten gleichzeitig mit mehr als einem Knoten verbunden sein können. In einer Sun Cluster 3.0-Umgebung ermöglicht der Mehrhostspeicher eine hohe Verfügbarkeit der Platteneinheiten. Platteneinheiten, die sich im Mehrhostspeicher befinden, können die Ausfälle einzelner Knoten tolerieren, da über alternative Serverknoten immer ein physischer Pfad zu den Daten verfügbar ist. Auf Mehrhostplatten kann global über einen primären Knoten zugegriffen werden. Wenn Clientanforderungen über einen Knoten auf Daten zugreifen und dieser Knoten fehlschlägt, werden die Anforderungen an einen anderen Knoten umgeleitet, der eine direkte Verbindung zu derselben Platte hat. Ein Volume Manager stellt Spiegelkonfigurationen oder Konfigurationen mit RAID 5 bereit, die die Datenredundanz auf den Mehrhostplatten gewährleisten. Sun Cluster 3.0 unterstützt derzeit Solstice DiskSuite und VERITAS Volume Manager als Datenträgermanager. Die Kombination von Mehrhostplatten mit Plattenspiegelung und einheitenübergreifendem Lesen und Schreiben von Daten bietet gleichermaßen Schutz vor Knotenausfällen und Ausfällen einzelner Platten.

Globale Einheiten

Globale Einheiten stellen clusterweiten Zugriff mit hoher Verfügbarkeit auf alle Einheiten in einem Cluster bereit, und zwar von jedem Knoten aus und unabhängig von der physischen Position der Einheit. Alle Platten sind im globalen Namensbereich mit einer zugeordneten Einheiten-ID (Device ID, DID) erfasst und als globale Einheiten konfiguriert. Daher sind die Platten für alle Clusterknoten sichtbar.

Dateisysteme/Globale Dateisysteme

Ein Cluster bzw. ein globales Dateisystem ist ein Proxy zwischen dem Kernel (auf einem Knoten) und dem Datenträgermanager des zugrunde liegenden Dateisystems (auf einem Knoten, der eine physische Verbindung zu mindestens einer Platte hat). Clusterdateisysteme sind von globalen Einheiten abhängig, die jeweils zu mindestens einem Knoten eine physische Verbindung haben. Sie sind unabhängig von dem zugrunde liegenden Dateisystem und Datenträgermanager. Derzeit können Clusterdateisysteme auf UFS eingerichtet werden, unter Verwendung von Solstice DiskSuite oder VERITAS Volume Manager. Die Daten werden nur dann allen Knoten zur Verfügung gestellt, wenn die Dateisysteme auf den Platten global als Clusterdateisystem eingerichtet sind.

Gerätegruppe

Eine Mehrhostplatte muss vom Sun Cluster-Framework gesteuert werden. Auf einer Mehrhostplatte werden zuerst so genannte Plattengruppen (Disk Groups) erstellt, die entweder von Solstice DiskSuite oder VERITAS Volume Manager verwaltet werden. Anschließend werden sie als Sun Cluster-Platteneinheitengruppen registriert. Eine Platteneinheitengruppe ist ein Typ der globalen Einheiten. Mehrhosteinheitengruppen haben eine hohe Verfügbarkeit. Wenn ein Knoten, der die Einheitengruppe gerade verwaltet, ausfällt, sind die Platten über einen alternativen Pfad weiter verfügbar. Der Ausfall des Knotens, der die Einheitengruppe verwaltet, hat keine Auswirkungen auf die Einheitengruppe, bis auf eine kurze zeitliche Unterbrechung, die zur Ausführung der Recovery und der Konsistenzprüfung erforderlich ist. Während dieser Zeit werden alle Anforderungen blockiert (für die Anwendung transparent), bis das System die Einheitengruppe wieder zur Verfügung stellt.

Ressourcengruppenmanager (RGM)

Der RGM stellt einen Hochverfügbarkeitsmechanismus bereit und wird auf jedem Clusterknoten als Dämon ausgeführt. Er übernimmt anhand von vordefinierten Richtlinien das automatische Starten und Stoppen von Ressourcen auf ausgewählten Knoten. Der RGM ermöglicht die hohe Verfügbarkeit einer Ressource im Fall eines Knotenausfalls bzw. stoppt die Ressource auf dem betreffenden Knoten und startet sie auf einem anderen Knoten erneut. Der RGM startet und stoppt ebenfalls automatisch ressourcenspezifische Monitore, die Ressourcenfehler feststellen und fehlschlagende Ressourcen auf andere Knoten verlagern können.

Datenbankservices

Mit dem Begriff Datenbankservice wird eine Anwendung eines Fremdanbieters bezeichnet, die zur Ausführung auf einem Cluster anstatt auf einem einzelnen Server konfiguriert ist. Ein Datenbankservice umfasst die Anwendungssoftware und die Software Sun Cluster 3.0, die die Anwendung startet, stoppt und überwacht. Sun Cluster 3.0 stellt Datenbankservicemethoden bereit, mit denen die Anwendung innerhalb des Clusters gesteuert und überwacht wird. Diese Methoden werden unter der Steuerung des Ressourcengruppenmanagers (RGM) ausgeführt, der mit diesen Methoden die Anwendung auf den Clusterknoten startet, stoppt und überwacht. Zusammen mit der Clusterframeworksoftware und mit Mehrhostplatten ermöglichen es diese Methoden, dass Anwendungen als Datenbankservices mit hoher Verfügbarkeit fungieren können. Als Datenbankservices mit hoher Verfügbarkeit können sie wesentliche Anwendungsunterbrechungen nach einzelnen Ausfällen innerhalb des Clusters verhindern, unabhängig davon, ob der Ausfall bzw. Fehler auf einem Knoten, einer Schnittstellenkomponente oder in der Anwendung selbst auftrat. Der RGM verwaltet auch die Ressourcen im Cluster, einschließlich der Netzwerkressourcen (wie logische Hostnamen und gemeinsam benutze Adressen) und der Anwendungsinstanzen.

Ressourcentyp, Ressource und Ressourcengruppe

Ein Ressourcentyp setzt sich aus Folgendem zusammen:

1. Einer Softwareanwendung, die auf dem Cluster ausgeführt wird.
2. Steuerprogrammen, die vom RGM als Rückrufmethoden verwendet werden, um die Anwendung als Clusterressource zu verwalten.
3. Einigen Merkmalen, die Teil der statischen Konfiguration eines Clusters sind.

Der RGM verwendet die Ressourcentypmerkmale zur Verwaltung der Ressourcen eines bestimmten Typs.

Eine Ressource erbt die Merkmale und Werte ihres Ressourcentyps. Eine Ressource ist eine Instanz der zugrunde liegenden Anwendung, die auf dem Cluster ausgeführt wird. Der Name der Instanz muss auf Clusterebene eindeutig sein. Jede Ressource muss in einer Ressourcengruppe konfiguriert sein. Der RGM versetzt alle Ressourcen in einer Gruppe auf demselben Knoten gleichzeitig in den Online- oder Offlinestatus. Zum Versetzen einer Ressourcengruppe in den Online- oder Offlinestatus ruft der RGM für jede Ressource in der Gruppe Rückrufmethoden auf.

Die Knoten, auf denen eine Ressourcengruppe online ist, werden als ihre primären Knoten oder Primärknoten bezeichnet. Eine Ressourcengruppe wird von ihren Primärknoten verwaltet. Jede Ressourcengruppe hat ein zugeordnetes Merkmal, eine "Knotenliste", die vom Clusteradministrator definiert wird und anhand derer alle potenziellen Primärknoten oder Master der Ressourcengruppe angegeben werden.

Unterstützung für VERITAS Cluster Server:

Wenn Sie vorhaben, Ihre DB2-Datenbanklösung auf einem Solaris-Betriebssystemcluster auszuführen, können Sie VERITAS Cluster Server für die Verwaltung des Clusters verwenden. VERITAS Cluster Server kann eine große Bandbreite an Anwendungen in heterogenen Umgebungen verwalten und unterstützt bis zu 32 Knotencluster sowohl in Speicherbereichsnetzwerken (SAN) als auch in traditionellen Client/Server-Umgebungen.

Hardwarevoraussetzungen

VERITAS Cluster Server unterstützt derzeit folgende Hardware:

- Für Serverknoten:
  - Jeden SPARC/Solaris-Server von Sun Microsystems mit mindestens 128 MB RAM, auf dem Solaris 2.6 oder höher ausgeführt wird.
- Für Plattenspeicher:
  - EMC Symmetrix, IBM Enterprise Storage Server, HDS 7700 und 9xxx, Sun T3, Sun A5000, Sun A1000, Sun D1000 und jeden anderen Plattenspeicher, der von VCS 2.0 oder höher unterstützt wird. Weitere Auskünfte zu den unterstützten Plattensubsystemen können Sie bei Ihrem VERITAS-Ansprechpartner einholen oder der VCS-Dokumentation entnehmen.
  - Typische Umgebungen erfordern (in jedem Clusterknoten) gespiegelte, private Platten für die DB2-Binärdateien sowie von Knoten gemeinsam genutzte Platten für die DB2-Daten.
- Für Netzverbindungen:
  - Für die öffentlichen Netzverbindungen eine beliebige Netzverbindung, die eine Adressierung auf IP-Basis unterstützt.
  - Für die Überwachungssignalverbindungen (clusterintern) sind redundante Überwachungssignalverbindungen erforderlich. Diese Voraussetzung kann durch den Einsatz von zwei zusätzlichen Ethernet-Controllern pro Server bzw. durch die Verwendung von einem zusätzlichen Ethernet-Controller pro Server und einer gemeinsam genutzten GAB-Platte pro Cluster erfüllt werden.

Softwarevoraussetzungen

Die folgenden VERITAS-Softwarekomponenten gelten als qualifizierte Konfigurationen:

- VERITAS Volume Manager 3.2 oder höher, VERITAS File System 3.4 oder höher, VERITAS Cluster Server 2.0 oder höher
- DB Edition für DB2 für Solaris 1.0 oder höher

Obwohl VERITAS Cluster Server keinen Datenträgermanager erfordert, wird dringend die Verwendung von VERITAS Volume Manager empfohlen, um die Installation, Konfiguration und Verwaltung zu vereinfachen.

Funktionsübernahme

VERITAS Cluster Server ist eine Hochverfügbarkeitslösung für Cluster, die eine höhere Verfügbarkeit von Anwendungsservices, zum Beispiel einer DB2-Datenbank, sicherstellt, indem sie eine Funktionsübernahme für Anwendungen ermöglicht. Der Status jedes einzelnen Clusterknotens und seiner zugehörigen Softwareservices wird regelmäßig überwacht. Wird ein Anwendungsservice (in diesem Fall der DB2-Datenbankservice) durch einen Fehler unterbrochen, stellt VERITAS Cluster Server

oder der VCS HA-DB2-Agent (oder beide) den Fehler fest und führt automatisch Schritte zur Wiederherstellung des Service aus. Diese Schritte zur Wiederherstellung des Service können den Neustart des DB2-Datenbanksystems auf demselben Knoten oder das Versetzen des DB2-Datenbanksystems auf einen anderen Knoten im Cluster und den anschließenden Neustart auf diesem Knoten beinhalten. Wenn eine Anwendung auf einen neuen Knoten migriert werden muss, versetzt VERITAS Cluster Server alles zur Anwendung Gehörende (z. B. IP-Netzwerkadressen, das Eigentumsrecht an zugehörigem Speicher) auf den neuen Knoten, sodass Benutzer nicht bemerken, dass der Service auf einem anderen Knoten ausgeführt wird. Die Benutzer verwenden zum Zugriff auf den Service weiterhin dieselben IP-Adressen, die nur jetzt auf einen anderen Clusterknoten verweisen.

Wenn in VERITAS Cluster Server ein Fehler auftritt, ist es möglich, dass Benutzer eine Unterbrechung des Service feststellen oder auch nicht. Dies hängt von dem Verbindungstyp ('stateful' oder 'stateless', d. h. mit oder ohne Austausch von Statusinformationen) zwischen dem Client und dem Anwendungsservice ab. In Anwendungsumgebungen, in denen Verbindungen mit Austausch von Statusinformationen verwendet werden (wie bei der DB2-Datenbank), stellen Benutzer möglicherweise eine kurze Serviceunterbrechung fest und müssen sich nach erfolgter Funktionsübernahme gegebenenfalls neu anmelden. In Anwendungsumgebungen, die Verbindungen ohne Austausch von Statusinformationen verwenden (wie bei NFS), stellen Benutzer möglicherweise eine kurze Serviceverzögerung, jedoch keine Unterbrechung fest und müssen sich nicht erneut anmelden.

Durch die Unterstützung einer Anwendung als Service, der automatisch zwischen Clusterknoten migriert werden kann, reduziert VERITAS Cluster Server nicht nur ungeplante Ausfallzeiten, sondern kann auch geplante Ausfallzeiten (z. B. für Wartung und Upgrades) verkürzen. Funktionsübernahmen können auch manuell eingeleitet werden. Wenn auf einem bestimmten Knoten ein Hardware- oder Betriebssystemupgrade ausgeführt werden muss, kann das DB2-Datenbanksystem auf einen anderen Knoten im Cluster migriert werden, das Upgrade ausgeführt und das DB2-Datenbanksystem anschließend wieder auf den ursprünglichen Knoten zurückmigriert werden.

Für diese Typen von Clusterumgebungen wird der Einsatz von absturztoleranten Anwendungen empfohlen. Eine absturztolerante Anwendung kann nach einem unvermittelt auftretenden Absturz wiederhergestellt werden, und die Integrität der festgeschrieben Daten bleibt erhalten. Absturztolerante Anwendungen werden auch als *clustertolerante* Anwendungen bezeichnet. Das DB2-Datenbanksystem ist eine absturztolerante Anwendung.

Gemeinsam genutzter Speicher

Beim Einsatz zusammen mit dem VCS HA-DB2-Agenten erfordert VERITAS Cluster Server gemeinsam genutzten Speicher. Gemeinsam genutzter Speicher ist ein Speicher, der eine physische Verbindung zu mehreren Knoten im Cluster hat. Im gemeinsam genutzten Speicher vorhandene Platteneinheiten sind Knotenausfällen gegenüber tolerant, da der physische Pfad zu den Platteneinheiten über die anderen Cluster gewährleistet bleibt.

Durch die Steuerung von VERITAS Cluster Server erhalten Clusterknoten Zugriff auf gemeinsam genutzten Speicher über ein logisches Konstrukt, so genannte Plattengruppen (Disk Groups). Plattengruppen sind eine Sammlung logisch definierter Speichereinheiten, deren Eigentumsrecht automatisch zwischen Knoten in einem Cluster migriert werden kann. Eine Plattengruppe kann nur jeweils in einen Knoten importiert werden. Wenn z. B. Plattengruppe A in Knoten 1 importiert wird

und Knoten 1 ausfällt, kann Plattengruppe A aus dem ausgefallenen Knoten exportiert und in einen neuen Knoten im Cluster importiert werden. VERITAS Cluster Server kann gleichzeitig mehrere Plattengruppen in einem einzelnen Cluster steuern.

Zusätzlich zur Definition von Plattengruppen kann ein Datenträgermanager unter Verwendung von Spiegelungstechnologie oder RAID 5 auf gemeinsam genutztem Speicher redundante Datenkonfigurationen bereitstellen. VERITAS Cluster Server unterstützt VERITAS Volume Manager und Solstice DiskSuite als Manager für logische Datenträger (LVM - Logical Volume Manager). Die Kombination von gemeinsam genutztem Speicher mit Plattenspiegelung und einheitenübergreifendem Lesen und Schreiben von Daten bietet gleichermaßen Schutz vor Knotenausfällen und Ausfällen einzelner Platten oder Controller.

VERITAS Cluster Server Global Atomic Broadcast (GAB) und Low Latency Transport (LLT)

In Clusterkonfigurationen ist ein knotenübergreifender Kommunikationsmechanismus erforderlich, damit Knoten Informationen zum Hardware- und Softwarestatus austauschen, Clusterzugehörigkeiten verfolgen und diese Informationen über alle Clusterknoten hinweg synchron halten können. VERITAS Cluster Server nutzt hierzu GAB (Global Atomic Broadcast), das zusammen mit LLT (Low Latency Transport) den erforderlichen Hochgeschwindigkeitsmechanismus mit geringen Latenzzeiten bereitstellt. GAB wird auf jedem Clusterknoten als Kernelmodul geladen und bietet einen ganzheitlichen Broadcastmechanismus, mit dem sichergestellt werden kann, dass alle Knoten die aktualisierten Statusinformationen gleichzeitig erhalten.

Durch den Einsatz eines Leistungsspektrums für kernelübergreifende Kommunikation bietet LLT eine Hochgeschwindigkeitsübertragungsmöglichkeit mit geringen Latenzzeiten für alle Informationen, die zwischen Clustern ausgetauscht und synchronisiert werden müssen. GAB wird auf LLT ausgeführt. VERITAS Cluster Server verwendet nicht IP als Überwachungssignalmechanismus, sondern bietet zwei zuverlässigere Optionen an. Es kann entweder GAB mit LLT als Überwachungssignalmechanismus konfiguriert werden, oder es kann eine GAB-Platte als Überwachungssignalfunktion auf Plattenbasis konfiguriert werden. Das Überwachungssignal muss über redundante Verbindungen ausgeführt werden. Diese Verbindungen können entweder zwei private Ethernet-Verbindungen zwischen Clusterknoten sein oder eine private Ethernet-Verbindung und eine GAB-Plattenverbindung. Die Verwendung von zwei GAB-Platten wird als Konfiguration nicht unterstützt, da für den Austausch von Clusterstatusinformationen zwischen Knoten eine private Ethernet-Verbindung erforderlich ist.

Weitere Informationen zu GAB oder LLT bzw. zu deren Konfiguration in VERITAS Cluster Server-Konfigurationen entnehmen Sie bitte dem Benutzerhandbuch von VERITAS Cluster Server 2.0 für Solaris.

Agentenbündel und Unternehmensagenten

Ein Agent ist ein Programm, das die Verfügbarkeit einer bestimmten Ressource oder Anwendung verwaltet. Wird ein Agent gestartet, ruft er die erforderlichen Konfigurationsinformationen von VCS ab und beginnt, die Ressource oder Anwendung regelmäßig zu überwachen und VCS mit dem Status zu aktualisieren. Im Allgemeinen werden Agenten dazu verwendet, Ressourcen in den Online- oder Offlinestatus zu versetzen und Ressourcen zu überwachen, wobei vier Servicetypen bereitgestellt werden: 'start' (starten), 'stop' (stoppen), 'monitor' (überwachen) und

'clean' (bereinigen). Mit 'start' und 'stop' werden Ressourcen in den Online- oder Offlinestatus versetzt, mit 'monitor' wird eine bestimmte Ressource oder Anwendung auf ihren Status hin überprüft, und 'clean' wird im Recoveryprozess verwendet.

Im Lieferumfang von VERITAS Cluster Server ist ein Agentenbündel enthalten, das mit VERITAS Cluster Server installiert wird. Die darin enthaltenen Agenten sind VCS-Prozesse, die vordefinierte Ressourcentypen verwalten, die in der Regel in Clusterkonfigurationen vorkommen, z. B. 'IP', 'mount' (anhängen), 'process' (verarbeiten) und 'share' (gemeinsam nutzen). Diese Agenten vereinfachen die Installation und Konfiguration von Clustern erheblich. Das mit VERITAS Cluster Server gelieferte Bündel umfasst über 20 Agenten.

Unternehmensagenten werden häufig auf bestimmte Anwendungen, z. B. die DB2-Datenbankanwendung, fokussiert. Der VCS HA-DB2-Agent kann als Unternehmensagent betrachtet werden, und er tauscht über das VCS-Agentenframework Daten mit VCS aus.

Ressourcen, Ressourcentypen und Ressourcengruppen von VCS

Ein Ressourcentyp ist eine Objektdefinition, mit der innerhalb eines VCS-Clusters die zu überwachenden Ressourcen definiert werden können. Ein Ressourcentyp umfasst den Ressourcentypnamen und eine Reihe von Merkmalen, die dieser Ressource zugeordnet sind und die unter Gesichtspunkten der Hochverfügbarkeit wichtig sind. Eine Ressource übernimmt die Merkmale und Werte ihres Ressourcentyps, und der Ressourcenname muss auf Clusterebene eindeutig sein.

Es gibt zwei Ressourcentypen: 'persistent' und 'standard' (nicht persistent). Persistente Ressourcen sind Ressourcen wie Netzschnittstellencontroller (NICs), die zwar überwacht, aber von VCS nicht in den Online- oder Offlinestatus versetzt werden. Standardressourcen sind Ressourcen, deren Online- und Offlinestatus von VCS gesteuert wird.

Das auf unterster Ebene überwachte Objekt ist eine Ressource, und es gibt verschiedene Ressourcentypen (z. B. gemeinsam genutzt, angehängt). Jede Ressource muss in eine Ressourcengruppe konfiguriert werden, und VCS versetzt alle Ressourcen einer Ressourcengruppe gleichzeitig in den Online- oder Offlinestatus. Zum Versetzen einer Ressourcengruppe in den Online- oder Offlinestatus ruft VCS für jede Ressource in der Gruppe die Methoden 'start' oder 'stop' auf. Es gibt zwei Ressourcengruppentypen: 'failover' (Funktionsübernahme) und 'parallel'. Eine DB2-Datenbankkonfiguration mit hoher Verfügbarkeit (in einer partitionierten oder nicht partitionierten Umgebung) verwendet Ressourcengruppen des Typs 'failover'.

Ein primärer Knoten bzw. Masterknoten ist ein Knoten, der potenziell eine Ressource enthalten kann. Mit dem Ressourcengruppenattribut `systemlist` wird angegeben, welche Knoten in einem Cluster primäre Knoten für eine bestimmte Ressourcengruppe sein können. In einem Cluster mit zwei Knoten sind normalerweise beide Knoten in der `systemlist` enthalten. In großen Clustern mit mehreren Knoten, die vielleicht mehrere Hochverfügbarkeitsanwendungen enthalten, kann es jedoch erforderlich sein, sicherzustellen, dass für bestimmte, von ihren Ressourcen auf der untersten Ebene definierte Anwendungsservices keine Funktionsübernahme auf bestimmten Knoten erfolgt.

Zwischen Ressourcengruppen können Abhängigkeiten definiert werden. VERITAS Cluster Server greift auf die Hierarchie dieser Ressourcengruppenabhängigkeiten zurück, um die Auswirkungen verschiedener Ressourcenausfälle zu bewerten und

die Recovery entsprechend zu verwalten. Beispiel: Falls die Ressourcengruppe 'ClientAnw1' erst in den Onlinestatus versetzt werden kann, wenn die Ressourcengruppe 'DB2' erfolgreich gestartet wurde, wird Ressourcengruppe 'ClientAnw1' als abhängig von Ressourcengruppe 'DB2' betrachtet.

Synchronisieren der Systemzeit in einer Umgebung mit partitionierten Datenbanken

Unter den Datenbankpartitionsservern sollte für eine relative hohe Synchronisation der Systemuhren gesorgt werden, um einen reibungslosen Datenbankbetrieb und eine uneingeschränkte Möglichkeit zur aktualisierenden Recovery zu gewährleisten. Zeitunterschiede zwischen den Datenbankpartitionsservern, einschließlich aller potenziellen betriebs- und kommunikationsbedingten Verzögerungen für eine Transaktion, sollten kleiner sein als der für den Konfigurationsparameter *max_time_diff* des Datenbankmanagers angegebene Wert, mit dem die maximale Zeitdifferenz zwischen Knoten definiert wird.

Um sicherzustellen, dass die Zeitmarken der Protokollsätze die Reihenfolge von Transaktionen in einer Umgebung mit partitionierten Datenbanken richtig wiedergeben, verwendet DB2 die Systemuhr und die in der Datei `SQLOGCTL.LFH` gespeicherte virtuelle Zeitmarke der jeweiligen Maschine als Basis für die Zeitmarken in den Protokollsätzen. Wenn die Systemuhr jedoch vorgestellt wird, wird die Protokolluhr automatisch mit vorgestellt. Obwohl die Systemuhr wieder zurückgestellt werden kann, kann die Uhr für die Protokolle dies nicht und bleibt auf dieser *vorgestellten* Zeit, bis die Systemuhr mit dieser Zeit übereinstimmt. Dann sind die Uhren synchron. Dies hat zur Konsequenz, dass ein kurzfristiger Systemuhrfehler auf einem Datenbankknoten einen langfristigen Effekt auf die Zeitmarken von Datenbankprotokollen haben kann.

Nehmen Sie zum Beispiel an, dass die Systemuhr auf Datenbankpartitionsserver A irrtümlicherweise auf den 7. November 2005 gesetzt wird, obwohl das Jahr 2003 ist, und nehmen Sie weiter an, dass der Fehler korrigiert wurde, *nachdem* eine Transaktion zur Aktualisierung in der Datenbankpartition auf diesem Datenbankpartitionsserver festgeschrieben wurde. Wenn die Datenbank ständig verwendet und regelmäßig aktualisiert wird, bleibt jeder Zeitpunkt zwischen dem 7. November 2003 und dem 7. November 2005 durch die aktualisierende Recovery praktisch unerreichbar. Wenn die COMMIT-Operation auf Datenbankpartitionsserver A erfolgt ist, wird die Zeitmarke im Datenbankprotokoll auf 2005 gesetzt und die Uhr des Protokolls bleibt auf dem 7. November 2005 stehen, bis die Systemuhr diesen Zeitpunkt erreicht. Wenn Sie versuchen, eine aktualisierende Recovery bis zu einem Zeitpunkt innerhalb dieses Zeitrahmens durchzuführen, stoppt die Operation an der ersten Zeitmarke, die über den angegebenen Stoppzeitpunkt, d. h. den 7. November 2003, hinausgeht.

Zwar kann DB2 etwaige Aktualisierungen der Systemuhr nicht steuern, der Konfigurationsparameter des Datenbankmanagers *max_time_diff* kann jedoch die Wahrscheinlichkeit reduzieren, dass diesbezüglich Probleme auftreten:

- Die konfigurierbaren Werte für diesen Parameter reichen von 1 Minute bis zu 24 Stunden.
- Wenn die erste Verbindungsanforderung an eine Nichtkatalogpartition erfolgt, sendet der Datenbankpartitionsserver seine Zeit an die Katalogpartition für die Datenbank. Die Katalogpartition überprüft, ob die Zeit in der Datenbankpartition, die die Verbindung anfordert, und ihre eigene Zeit innerhalb des durch den Parameter *max_time_diff* definierten Bereichs liegen. Falls dieser Bereich überschritten wird, wird die Verbindung verweigert.

- Eine Aktualisierungstransaktion, die auf mehr als zwei Datenbankpartitionsserver in der Datenbank zugreift, muss überprüfen, ob die Systemuhren auf den beteiligten Datenbankpartitionsservern synchron sind, bevor die Aktualisierung festgeschrieben werden kann. Wenn zwei oder mehr Datenbankpartitionsserver eine Zeitdifferenz aufweisen, die den durch den Parameter *max_time_diff* gesetzten Grenzwert überschreitet, wird die Transaktion rückgängig gemacht, um zu verhindern, dass die falsche Zeit auf weitere Datenbankpartitionsserver übertragen wird.

Konvertieren von Client/Server-Zeitmarken

In diesem Abschnitt wird die Generierung von Zeitmarken in einer Client/Server-Umgebung erläutert:

- Wenn Sie für eine aktualisierende Recovery eine Ortszeit angeben, werden alle Nachrichten ebenfalls in Ortszeit zurückgegeben.

  Anmerkung: Alle Zeitangaben werden auf dem Server und (in Umgebungen mit partitionierten Datenbanken) in der Katalogdatenbankpartition konvertiert.
- Die Zeichenfolge der Zeitmarke wird auf dem Server in GMT konvertiert, sodass die Zeitmarke die Zeitzone des Servers und nicht des Clients wiedergibt. Wenn sich der Client in einer anderen Zeitzone befindet als der Server, sollte die Ortszeit des Servers verwendet werden.
- Wenn die Zeichenfolge der Zeitmarke aufgrund der Sommerzeit nahe am Zeitwechsel liegt, muss klar sein, ob die Stoppzeit vor oder nach dem Zeitwechsel liegt, damit sie korrekt angegeben wird.

Kapitel 4. Verwaltung und Pflege einer Hochverfügbarkeitslösung

Wenn Ihre hoch verfügbare DB2-Datenbanklösung erstellt und konfiguriert ist und aktiv ausgeführt wird, gibt es einige laufend auszuführenden Aktivitäten. Sie müssen Ihre Datenbanklösung überwachen, pflegen und reparieren, damit sie für die Clientanwendungen zur Verfügung steht.

Die folgenden Dinge müssen von Ihnen bei der Ausführung Ihres Datenbanksystems überwacht und gehandhabt werden:

1. Das Verwalten von Protokolldateien:

   Protokolldateien nehmen an Umfang zu und müssen archiviert werden; einige Protokolldateien müssen kopiert oder versetzt werden, damit sie für eine Restoreoperation zur Verfügung stehen.
2. Das Ausführen von Verwaltungsaktivitäten:
   - Software installieren
   - Upgrades für Hardware durchführen
   - Datenbanktabellen reorganisieren
   - Datenbankleistung optimieren
   - Datenbank-Backups
3. Primäre und sekundäre bzw. Bereitschaftsdatenbank synchronisieren, sodass die Funktionsübernahme (Failover) reibungslos funktioniert.
4. Unerwartete Ausfälle oder Fehler in der Hardware oder Software identifizieren und beseitigen.

Protokolldateiverwaltung

Der DB2 Datenbankmanager verwendet ein Nummerierungsschema zur Benennung der Protokolldateien. Diese Benennungsstrategie hat Auswirkungen auf die Wiederverwendung von Protokolldateien und auf die Protokollfolgen. Darüber hinaus verwendet eine DB2-Datenbank ohne Clientanwendungsverbindung eine neue Protokolldatei, wenn die nächste Clientanwendung eine Verbindung zu dem Datenbankserver herstellt. Diese beiden Aspekte des Datenbankprotokollierungsverhaltens von DB2 Data Server wirken sich auf Ihre Auswahl in Bezug auf die Protokolldateiverwaltung aus.

Bei der Verwaltung von Datenbankprotokollen sind folgende Faktoren zu beachten:

- Das Nummerierungsschema für Archivprotokolldateien beginnt mit S0000000.LOG und reicht bis S9999999.LOG und bietet somit Platz für bis zu 10 Millionen Protokolldateien. Der Datenbankmanager beginnt in folgenden Situationen erneut bei S0000000.LOG:
  - Die Konfigurationsdatei der Datenbank wurde geändert, um die aktualisierende Recovery zu aktivieren.
  - Die Konfigurationsdatei der Datenbank wurde geändert, um die aktualisierende Recovery zu *inaktivieren*.
  - S9999999.LOG wurde verwendet

  Protokolldateinamen werden vom DB2-Datenbankmanager nach dem Wiederherstellen einer Datenbank (mit oder ohne aktualisierende Recovery) wiederverwendet. Der Datenbankmanager stellt sicher, dass bei der aktualisierenden Reco-

very kein falsches Protokoll verwendet wird. Falls der DB2-Datenbankmanager nach einer Wiederherstellungsoperation einen Protokolldateinamen wiederverwendet, werden die neuen Protokolldateien in separaten Verzeichnissen archiviert, damit es möglich ist, mehrere Protokolldateien mit demselben Namen zu archivieren. Die Position der Protokolldateien wird in der Datei des Recoveryprotokolls aufgezeichnet, sodass die Dateien während einer aktualisierenden Recovery angewendet werden können. Sie müssen sicherstellen, dass die richtigen Protokolle für die aktualisierende Recovery zur Verfügung stehen.

Wenn die aktualisierende Recovery erfolgreich durchgeführt wurde, wird das letzte bei der aktualisierenden Recovery verwendete Protokoll abgeschnitten und die Protokollierung mit dem nächsten sequenziellen Protokoll fortgesetzt. Dies hat zur Folge, dass jedes Protokoll im Protokollpfadverzeichnis mit einer Folgenummer, die größer als die des letzten für die aktualisierende Recovery verwendeten Protokolls ist, erneut verwendet wird. Im abgeschnittenen Protokoll werden alle auf die Position des Schnitts folgenden Einträge mit Nullen überschrieben. Stellen Sie sicher, dass Sie vor dem Aufrufen des Dienstprogramms zur aktualisierenden Recovery eine Kopie der Protokolle erstellen. (Sie können ein Benutzerexitprogramm aufrufen, um die Protokolle an eine andere Position zu kopieren.)

- Wenn eine Datenbank nicht aktiviert wurde (über den Befehl ACTIVATE DATABASE), schneidet der DB2-Datenbankmanager die laufende Protokolldatei ab, sobald für alle Anwendungen die Verbindung zur Datenbank getrennt worden ist. Beim nächsten Herstellen einer Verbindung zur Datenbank durch die Anwendung startet der DB2-Datenbankmanager die Protokollierung in einer neuen Protokolldatei. Wenn auf Ihrem System viele kleine Protokolldateien erstellt werden, kommt vielleicht die Verwendung des Befehls ACTIVATE DATABASE in Betracht. Sie sparen hierdurch nicht nur Systemaufwand zur Initialisierung der Datenbank, wenn Anwendungen eine Verbindung herstellen, sondern auch den Systemaufwand, eine große Protokolldatei zuzuordnen, die Datei abzuschneiden und eine neue große Protokolldatei zuzuordnen.
- Ein archiviertes Protokoll kann zwei oder mehreren verschiedenen *Protokollsequenzen* einer Datenbank zugeordnet sein, weil die Protokollnamen wieder verwendet werden (siehe Abb. 8 auf Seite 165). Wenn Sie beispielsweise Backup 2 wiederherstellen wollen, könnten hierfür zwei verschiedene Protokollfolgen verwendet werden. Wenn Sie während einer vollständigen Recovery der Datenbank die aktualisierende Recovery bis zu einem bestimmten Zeitpunkt durchführen und dann vor Erreichen des Protokollendes stoppen, erstellen Sie dadurch eine neue Protokollfolge. Es ist nicht möglich, die beiden Protokollfolgen zu kombinieren. Wenn Sie über ein Image eines Online-Backups verfügen, das sich über die erste Protokollfolge erstreckt, müssen Sie die aktualisierende Recovery mit der ersten Protokollfolge beenden.

  Wenn Sie nach der Recovery eine neue Protokollfolge erstellt haben, sind Tabellenbereichsbackups in den alten Protokollfolgen ungültig. Dies wird normalerweise beim Restore erkannt, das Restoredienstprogramm kann jedoch ein Tabellenbereichsbackup in einer alten Protokollfolge nicht erkennen, wenn die Restoreoperation für den Tabellenbereich unmittelbar auf eine Restoreoperation für die Datenbank folgt. Bis zur tatsächlichen aktualisierenden Recovery der Datenbank ist die zu verwendende Protokollfolge nicht bekannt. Wenn der Tabellenbereich zu einer alten Protokollfolge gehört, muss er von der Operation zur aktualisierenden Recovery des Tabellenbereichs „erfasst" werden. Eine Restoreoperation, bei der ein ungültiges Backup-Image verwendet wird, kann erfolgreich beendet werden, eine anschließende aktualisierende Recovery dieses Tabellenbereichs wird jedoch fehlschlagen, und der Tabellenbereich befindet sich weiter im Status *Restore anstehend*.

Nehmen Sie beispielsweise an, dass eine Backup-Operation auf Tabellenbereichsebene, Backup 3, zwischen S0000013.LOG und S0000014.LOG in der oberen Protokollfolge beendet wird (siehe Abb. 8). Wenn Sie unter Verwendung des Backup-Images auf Datenbankebene, Backup 2, einen Restore und eine aktualisierende Recovery durchführen wollen, müssen Sie die aktualisierende Recovery bis S0000012.LOG durchführen. Anschließend könnten Sie die aktualisierende Recovery entweder bis zum Ende der oberen Protokollfolge oder bis zum Ende der unteren (neueren) Protokollfolge fortsetzen. Wenn Sie die untere Protokollfolge verwenden, können Sie das Backup-Image auf Tabellenbereichsebene, Backup 3, nicht zum Restore und zur aktualisierenden Recovery verwenden.

Um unter Verwendung des Backup-Images auf Tabellenbereichsebene, Backup 3, für einen Tabellenbereich eine aktualisierende Recovery bis zum Protokollende auszuführen, müssen Sie das Backup-Image auf Datenbankebene, Backup 2, wiederherstellen und anschließend die obere Protokollfolge für eine aktualisierende Recovery verwenden. Nach dem Restore des Backup-Images auf Tabellenbereichsebene, Backup 3, können Sie eine aktualisierende Recovery bis zum Ende der Protokolle durchführen.

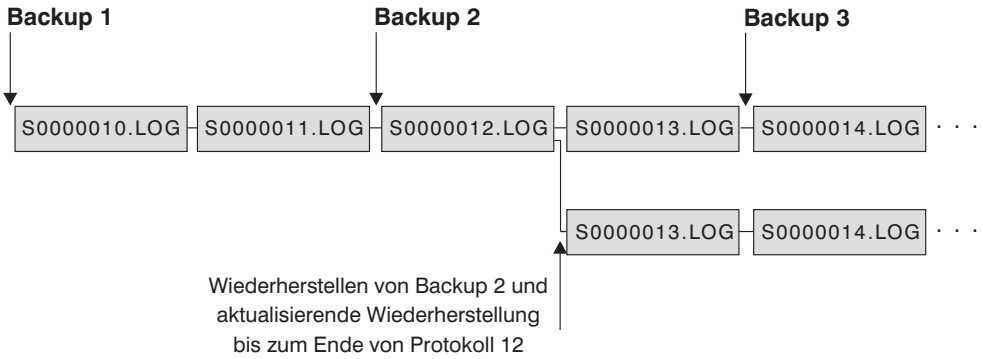


Abbildung 8. Erneutes Verwenden der Namen von Protokolldateien

Bedarfsgesteuerte Protokollarchivierung

IBM Data Server unterstützt das Schließen (und die Archivierung, falls aktiviert) der aktiven Protokolldatei bei einer wiederherstellbaren Datenbank zu jedem beliebigen Zeitpunkt. Dadurch haben Sie die Möglichkeit, eine Reihe von Protokolldateien bis zu einem bekannten Zeitpunkt zu sammeln und diese Dateien anschließend zur Aktualisierung einer Bereitschaftsdatenbank zu verwenden.

Sie können die bedarfsgesteuerte Protokollarchivierung einleiten, indem Sie den Befehl ARCHIVE LOG oder die Anwendungsprogrammierschnittstelle db2ArchiveLog aufrufen.

Protokollarchivierung mit db2tapemgr

Sie können das Dienstprogramm 'db2tapemgr' zum Speichern von archivierten Protokolldateien auf Bandeinheiten verwenden. Das Dienstprogramm db2tapemgr kopiert die Protokolldateien von der Platte auf die angegebene Bandeinheit und aktualisiert die Datei des Recoveryprotokolls mit der neuen Speicherposition der kopierten Protokolldateien.

Konfiguration

Setzen Sie den Datenbankkonfigurationsparameter LOGARCHMETH1 für die Protokolldateien, die Sie auf Band kopieren möchten, auf die momentane Speicherpo-

sition auf der Platte. Das Dienstprogramm db2tapemgr liest den Wert LOGARCHMETH1, um die zu kopierenden Protokolldateien zu suchen. In einer Umgebung mit partitionierten Datenbanken muss der Konfigurationsparameter LOGARCHMETH1 auf alle Datenbankpartitionen gesetzt werden, die zu kopierende Protokolldateien enthalten.

Das Dienstprogramm db2tapemgr verwendet nicht den Datenbankkonfigurationsparameter LOGARCHMETH2.

Optionen STORE und DOUBLE STORE

Führen Sie den Befehl DB2TAPEMGR entweder mit der Option STORE oder mit der Option DOUBLE STORE aus, um archivierte Protokolle von der Platte auf Band zu übertragen.

- Die Option STORE speichert einige oder alle Protokolldateien aus dem Protokollarchivverzeichnis auf einer angegebenen Bandeinheit und löscht die Dateien von der Platte.
- Die Option DOUBLE STORE durchsucht die Verlaufsdatei, um zu prüfen, ob zuvor bereits Protokolle auf Band gespeichert wurden.
  - Wenn ein Protokoll zuvor nicht gespeichert wurde, speichert DB2TAPEMGR die Protokolldatei auf Band, löscht sie jedoch nicht von der Platte.
  - Wenn ein Protokoll zuvor gespeichert wurde, speichert DB2TAPEMGR die Protokolldatei auf Band und löscht sie von der Platte.

  Verwenden Sie die Option DOUBLE STORE, wenn Sie Duplikatkopien Ihrer archivierten Protokolle auf Band und auf Platte behalten möchten oder wenn Sie dieselben Protokolle auf zwei verschiedenen Bändern speichern wollen.

Wenn Sie den Befehl DB2TAPEMGR entweder mit der Option STORE oder mit der Option DOUBLE STORE ausführen, durchsucht das Dienstprogramm 'db2tapemgr' zunächst die Verlaufsdatei nach Einträgen, in denen der Konfigurationsparameter LOGARCHMETH1 auf eine Plattenposition eingestellt ist. Wenn es feststellt, dass Dateien, die sich auf Platte befinden sollten, dort nicht zu finden sind, gibt es eine Warnung aus. Wenn das Dienstprogramm 'db2tapemgr' keine Protokolldateien zum Speichern findet, stoppt es die Ausführung und gibt eine Nachricht aus, um Sie zu informieren, dass nichts zu tun ist.

RETRIEVE-Optionen

Führen Sie den Befehl DB2TAPEMGR mit der Option RETRIEVE aus, um Dateien vom Band auf Platte zu übertragen.

- Verwenden Sie die Option RETRIEVE ALL LOGS oder LOGS n TO n, um alle archivierten Protokolle, die den von Ihnen angegebenen Kriterien entsprechen, abzurufen und auf Platte zu kopieren.
- Verwenden Sie die Option RETRIEVE FOR ROLLFORWARD TO POINT-IN-TIME, um alle archivierten Protokolle, die zur Durchführung einer aktualisierenden Recoveryoperation erforderlich sind, abzurufen und auf Platte zu kopieren.
- Verwenden Sie die Option RETRIEVE HISTORY FILE, um die Verlaufsdatei vom Band abzurufen und auf Platte zu kopieren.

Verhalten

- Wenn das Dienstprogramm 'db2tapemgr' Protokolldateien auf Platte findet, liest es die Bandkopfdaten, um sicherzustellen, dass die Protokolldateien auf das Band geschrieben werden können. Außerdem aktualisiert es die Verlaufsdatei für

die Dateien, die sich momentan auf Band befinden. Wenn die Aktualisierung fehlschlägt, stoppt es die Ausführung und zeigt eine Fehlernachricht an.

- Wenn das Band beschreibbar ist, kopiert das Dienstprogramm 'db2tapemgr' die Protokolle auf Band. Nach dem Kopieren der Dateien werden die Protokolldateien von der Platte gelöscht. Zum Schluss kopiert das Dienstprogramm 'db2tapemgr' die Verlaufsdatei auf das Band und löscht sie von der Platte.
- Das Dienstprogramm 'db2tapemgr' hängt keine Protokolldateien an ein Band an. Wenn eine Speicheroperation nicht das gesamte Band füllt, kann der verbleibende freie Bandspeicherplatz nicht mehr genutzt werden.
- Das Dienstprogramm 'db2tapemgr' speichert Protokolldateien nur einmal auf einem gegebenen Band. Durch diese Einschränkung sollen Probleme, die mit dem Schreiben auf Banddatenträgern verbunden sind, zum Beispiel eine Streckung des Bandes, vermieden werden.
- In einer Umgebung mit partitionierten Datenbanken lässt sich das Dienstprogramm 'db2tapemgr' nur jeweils für eine Datenbankpartition gleichzeitig ausführen. Sie müssen den entsprechenden Befehl für jede Datenbankpartition ausführen, indem Sie die Datenbankpartitionsnummer in der Option ON DBPARTITIONNUM des Befehls DB2TAPEMGR angeben. Darüber hinaus müssen Sie sicherstellen, dass jede Datenbankpartition Zugriff auf eine Bandeinheit besitzt.

Beispiele

Das folgende Beispiel zeigt, wie der Befehl DB2TAPEMGR verwendet wird, um alle Protokolldateien aus dem primären Archivprotokollpfad für die Datenbank SAMPLE in der Datenbankpartition mit der Nummer 0 auf einer Bandeinheit zu speichern und die Protokolldateien aus dem Archivprotokollpfad zu entfernen:

```
db2tapemgr db sample on dbpartitionnum 0 store on /dev/rmt0.1 all logs
```

Das folgende Beispiel zeigt, wie die ersten 10 Protokolldateien aus dem primären Archivprotokollpfad auf einer Bandeinheit gespeichert und die Protokolldateien aus dem Archivprotokollpfad entfernt werden:

```
db2tapemgr db sample on dbpartitionnum store on /dev/rmt0.1 10 logs
```

Das folgende Beispiel zeigt, wie die ersten 10 Protokolldateien aus dem primären Archivprotokollpfad auf einer Bandeinheit gespeichert werden, anschließend dieselben Protokolldateien auf einem zweiten Band gespeichert werden und schließlich diese Protokolldateien aus dem Archivprotokollpfad entfernt werden:

```
db2tapemgr db sample on dbpartitionnum double store on /dev/rmt0.1 10 logs
db2tapemgr db sample on dbpartitionnum double store on /dev/rmt1.1 10 logs
```

Das folgende Beispiel zeigt, wie alle Protokolldateien von einem Band in ein Verzeichnis abgerufen werden:

```
db2tapemgr db sample on dbpartitionnum retrieve all logs from /dev/rmt1.1
         to /home/dbuser/archived_logs
```

Archivierung und Abruf von Protokolldateien mithilfe von Benutzerexitprogrammen automatisieren

Sie können zur Automtisierung der Archivierung und des Abrufs von Protokolldateien ein Benutzerexitprogramm erstellen, das vom DB2-Datenbankmanager aufgerufen wird, um die Archivierungs- oder Abrufoperation auszuführen.

Wenn der DB2-Datenbankmanager Ihr Benutzerexitprogramm aufruft, geschieht Folgendes:

- Der Datenbankmanager übergibt die Steuerung an das Benutzerexitprogramm,
- der Datenbankmanager übergibt Parameter an das Benutzerexitprogramm, und
- das Benutzerexitprogramm gibt bei seiner Beendigung einen Rückkehrcode an den Datenbankmanager zurück.

Konfiguration

Bevor Sie ein Benutzerexitprogramm zur Archivierung oder zum Abruf von Protokolldateien aufrufen, muss der Datenbankkonfigurationsparameter logarchmeth1 auf USEREXIT gesetzt sein. Dies ermöglicht außerdem die aktualisierende Recovery der Datenbank.

Voraussetzungen für Benutzerexitprogramme

- Die ausführbare Datei für Ihr Benutzerexitprogramm muss auf den Namen db2uext2 lauten.
- Benutzerexitprogramme müssen Protokolldateien aus dem Pfad für aktive Protokolldateien in den Archivprotokollpfad kopieren, nicht versetzen. Entfernen Sie keine Protokolldateien aus dem Pfad für aktive Protokolldateien. Wenn Sie Protokolldateien aus dem Pfad für aktive Protokolldateien entfernen, kann Ihre DB2-Datenbank nach einem Fehler oder Ausfall möglicherweise nicht mehr erfolgreich wiederhergestellt werden.

  Für die DB2-Datenbank müssen die Protokolldateien während der Recovery im Pfad für aktive Protokolldateien vorhanden sein. Der DB2-Datenbankserver entfernt archivierte Protokolldateien aus dem Pfad der aktiven Protokolldateien, sobald diese Protokolldateien für die Recovery nicht mehr erforderlich sind.
- Benutzerexitprogramme müssen Fehlerbedingungen handhaben können. Dies ist notwendig, da der DB2-Datenbankmanager nur eine begrenzte Anzahl an Rückkehrbedingungen bearbeiten kann.

  Siehe hierzu „Fehlerbehandlung für Benutzerexits" auf Seite 170.
- Jede DB2-Datenbankmanagerinstanz kann lediglich ein Benutzerexitprogramm aufrufen. Daher müssen Sie für jede Operation, die Ihr Benutzerexitprogramm möglicherweise ausführen soll, einen Abschnitt entwickeln.

Benutzerexitbeispielprogramme

Benutzerexitbeispielprogramme sind für alle unterstützten Plattformen verfügbar. Sie können diese Programme modifizieren, um sie an Ihre spezifischen Anforderungen anzupassen. Die Beispielprogramme sind gut kommentiert und helfen Ihnen, sie sehr effektiv zu nutzen.

Sie müssen beachten, dass Benutzerexitprogramme Protokolldateien aus dem Pfad für aktive Protokolldateien in den Archivprotokollpfad *kopieren* müssen. Entfernen Sie keine Protokolldateien aus dem Pfad für aktive Protokolldateien. (Dies könnte bei der Datenbankrecovery Fehler verursachen.) DB2 entfernt archivierte Protokolldateien aus dem Pfad der aktiven Protokolldateien, sobald diese Protokolldateien für die Recovery nicht mehr erforderlich sind.

Im Folgenden werden die mit DB2 Data Server ausgelieferten Benutzerexitbeispielprogramme beschrieben.

- **UNIX-basierte Systeme**

  Die Benutzerexitbeispielprogramme für DB2 Data Server für UNIX-basierte Systeme befinden sich im Unterverzeichnis `sqllib/samples/c`. Obwohl die Beispielprogramme in der Programmiersprache C codiert sind, können Sie Ihr Benutzerexitprogramm auch in einer anderen Programmiersprache schreiben.

Das Benutzerexitprogramm muss eine ausführbare Datei sein, deren Name db2uext2 lautet.

Die folgenden vier Benutzerexitbeispielprogramme für UNIX-basierte Systeme sind verfügbar:

– db2uext2.ctsm

  Dieses Beispielprogramm verwendet Tivoli Storage Manager, um Datenbankprotokolldateien zu archivieren und abzurufen.

– db2uext2.ctape

  Dieses Beispielprogramm verwendet Banddatenträger, um Datenbankprotokolldateien zu archivieren und abzurufen.

– db2uext2.cdisk

  Dieses Beispielprogramm verwendet den Betriebssystembefehl COPY und Platten, um Datenbankprotokolldateien zu archivieren und abzurufen.

– db2uxt2.cxbsa

  Dieses Beispiel arbeitet mit XBSA Draft 0.8 von X/Open Group. Sie können damit Datenbankprotokolldateien archivieren und abrufen. Dieses Beispielprogramm wird nur unter AIX unterstützt.

- **Windows-Betriebssysteme**

  Die Benutzerexitbeispielprogramme für DB2 Data Server für Windows-Betriebssysteme befinden sich im Unterverzeichnis sqllib\samples\c. Obwohl die Beispielprogramme in der Programmiersprache C codiert sind, können Sie Ihr Benutzerexitprogramm auch in einer anderen Programmiersprache schreiben.

  Das Benutzerexitprogramm muss eine ausführbare Datei sein, deren Name db2uext2 lautet.

  Für Windows-Betriebssysteme sind zwei Benutzerexitbeispielprogramme verfügbar:

  – db2uext2.ctsm

    Dieses Beispielprogramm verwendet Tivoli Storage Manager, um Datenbankprotokolldateien zu archivieren und abzurufen.

  – db2uext2.cdisk

    Dieses Beispielprogramm verwendet den Betriebssystembefehl COPY und Platten, um Datenbankprotokolldateien zu archivieren und abzurufen.

Aufrufformat des Benutzerexitprogramms

Wenn der DB2-Datenbankmanager ein Benutzerexitprogramm aufruft, übergibt er einen Satz von Parametern (mit dem Datentyp CHAR) an das Programm.

Befehlssyntax

```
     db2uext2 -OS<bs> -RL<db2rel> -RQ<anforderung>  -DB<dbname>
     -NN<knotennr> -LP<protokollpfad> -LN<protokollname> -AP<tsmkennwort>
 -SP<startseite> -LS<protokollgröße>
```

bs Gibt die Plattform an, auf der die Instanz aktiv ist. Gültige Werte: AIX, Solaris, HP-UX, SCO, Linux und NT.

db2rel Gibt den Releasestand von DB2 an. Beispiel: SQL07020

anforderung
Gibt den Anforderungstyp an. Gültige Werte: ARCHIVE und RETRIEVE.

dbname
Gibt einen Datenbanknamen an.

nodenum
Gibt die Nummer des lokalen Knotens an, z. B. 5.

logpath
Gibt den vollständig qualifizierten Pfad zu den Protokolldateien an. Der Pfad muss das abschließende Pfadtrennzeichen enthalten. Beispiel: `/u/database/log/path/` oder `d:\logpath\`.

protokollname
Gibt den Namen der Protokolldatei an, die archiviert oder abgerufen werden soll, z. B. `S0000123.LOG`.

tsmkennwort
Gibt das TSM-Kennwort an. (Wurde zuvor für den Datenbankkonfigurationsparameter *tsm_password* ein Wert angegeben, wird dieser Wert an das Benutzerexitprogramm übergeben.)

startseite
Gibt die Zahl der relativen 4-KB-Adressseiten der Einheit an, auf welcher der Protokollspeicherbereich beginnt.

protokollgröße
Gibt die Größe des Protokollspeicherbereichs in 4-KB-Seiten an. Dieser Parameter ist nur gültig, wenn Sie zum Protokollieren eine unformatierte Einheit einsetzen.

Fehlerbehandlung für Benutzerexits

Wenn Sie ein Benutzerexitprogramm zur Automtisierung der Archivierung und des Abrufs von Protokolldateien erstellen, übergibt das Benutzerexitprogramm Rückkehrcodes an den DB2-Datenbankmanager, der das Programm aufgerufen hat. Der DB2-Datenbankmanager kann nur eine begrenzte Liste spezifischer Fehlercodes bearbeiten. Ihr Benutzerexitprogramm trifft jedoch möglicherweise auf viele verschiedene Fehlerbedingungen wie z. B. Betriebssystemfehler. Das Benutzerexitprogramm muss die auftretenden Fehlerbedingungen den Fehlercodes zuordnen, die der Datenbankmanager bearbeiten kann.

Tabelle 9 zeigt die Codes, die von einem Benutzerexitprogramm zurückgegeben werden können, und beschreibt, wie der Datenbankmanager diese Codes interpretiert. Wenn ein Rückkehrcode in dieser Tabelle nicht aufgeführt ist, wird er so behandelt, als hätte er den Wert 32.

Tabelle 9. Rückkehrcodes von Benutzerexitprogrammen. Nur für Archivierungs- und Abrufoperationen gültig.

Rückkehrcode	Erläuterung
0	Erfolgreich beendet.
4	Temporärer Ressourcenfehler trat auf.[a]
8	Bedienereingriff ist erforderlich.[a]
12	Hardwarefehler.[b]
16	Fehler bei Benutzerexitprogramm oder einer vom Programm verwendeten Softwarefunktion.[b]
20	Fehler bei mindestens einem Parameter, der an das Benutzerexitprogramm übergeben wurde. Prüfen Sie, ob das Benutzerexitprogramm die angegebenen Parameter korrekt verarbeitet.[b]
24	Das Benutzerexitprogramm wurde nicht gefunden. [b]
28	Ein Fehler trat auf, der durch einen E/A-Fehler oder durch das Betriebssystem verursacht wurde.[b]

Tabelle 9. Rückkehrcodes von Benutzerexitprogrammen (Forts.). Nur für Archivierungs- und Abrufoperationen gültig.

Rückkehrcode	Erläuterung
32	Das Benutzerexitprogramm wurde vom Benutzer beendet.[b]
255	Das Benutzerexitprogramm verursachte einen Fehler, da es die Bibliotheksdatei für die ausführbare Datei nicht laden konnte.[c]

[a] Bei Archivierungs- und Abrufanforderungen bewirkt ein Rückkehrcode 4 oder 8 eine Wiederholung nach fünf Minuten. Wenn das Benutzerexitprogramm auf Abrufanforderungen für dieselbe Protokolldatei weiterhin 4 oder 8 zurückgibt, führt DB2 so lange Wiederholungen aus, bis die Operation erfolgreich ist. (Dies gilt für aktualisierende Recoverys oder für Aufrufe der Anwendungsprogrammierschnittstelle db2ReadLog, die vom Replikationsdienstprogramm verwendet wird.)

[b] Aufrufe für Benutzerexitprogramme werden fünf Minuten lang ausgesetzt. Während dieser Zeit werden alle Anforderungen ignoriert, einschließlich der Anforderung, die die Fehlerbedingung verursachte. Nach dieser fünfminütigen Aussetzung wird die nächste Anforderung verarbeitet. Wenn diese Anforderung ohne Fehler verarbeitet wurde, wird die Verarbeitung neuer Benutzerexitanforderungen fortgesetzt, und DB2 setzt die Archivierungsanforderung, die fehlgeschlagen oder vorher ausgesetzt worden war, erneut ab. Wenn während der Wiederholung ein Rückkehrcode größer als acht generiert wird, werden Anforderungen für weitere fünf Minuten ausgesetzt. Die fünfminütigen Aussetzungen werden solange fortgesetzt, bis der Fehler behoben ist oder bis die Datenbank gestoppt und erneut gestartet wurde. Nachdem alle Anwendungen von der Datenbank getrennt wurden, setzt DB2 eine Archivierungsanforderung für alle Protokolldateien ab, die vorher möglicherweise nicht erfolgreich archiviert wurden. Wenn das Benutzerexitprogramm Protokolldateien nicht archivieren kann, füllt sich die Platte möglicherweise mit Protokolldateien an, und die Leistung verschlechtert sich. Wenn die Platte voll ist, nimmt der Datenbankmanager keine weiteren Anwendungsanforderungen für Datenbankaktualisierungen mehr entgegen. Wenn das Benutzerexitprogramm zum Abrufen von Protokolldateien aufgerufen wurde, wird die aktualisierende Recovery ausgesetzt, jedoch nicht gestoppt, es sei denn, die Option ROLLFORWARD STOP wurde angegeben. Wenn Sie die Option STOP nicht angegeben haben, können Sie den Fehler beheben und die Recovery wieder aufnehmen.

[c] Wenn das Benutzerexitprogramm den Fehlercode 255 zurückgibt, kann das Programm die Bibliotheksdatei für die ausführbare Datei wahrscheinlich nicht laden. Rufen Sie das Benutzerexitprogramm manuell auf, um dies zu prüfen. Weitere Informationen werden angezeigt.

Anmerkung: Bei Archivierungs- und Abrufoperationen wird ein Alert für alle Rückkehrcodes außer 0 und 4 ausgegeben. Der Alert enthält den Rückkehrcode vom Benutzerexitprogramm und eine Kopie der Eingabeparameter, die an das Benutzerexitprogramm übergeben wurden.

Zuordnen und Entfernen von Protokolldateien

Eine für die Recovery nach einem Systemabsturz benötigte Protokolldatei wird als aktive Protokolldatei bezeichnet. Sofern nicht die Endlosprotokollierung aktiviert ist, werden die im Datenbankprotokollverzeichnis vorhandenen Protokolldateien nicht entfernt, wenn sie eventuell zur Recovery nach einem Systemabsturz benötigt werden.

Wurde die Endlosprotokollierung aktiviert und muss Speicherbereich für weitere aktive Protokolldateien verfügbar sein, archiviert der Datenbankmanager eine aktive Protokolldatei und benennt sie um, um eine neue aktive Protokolldatei zu erstellen. Falls eine Recovery nach Systemabsturz notwendig ist, wenn die End-

losprotokollierung aktiviert ist, müssen die Protokolldateien möglicherweise aus dem Archivprotokollpfad abgerufen werden, um die Recovery nach Systemabsturz abzuschließen.

Wenn der Datenbankkonfigurationsparameter *logarchmeth1* nicht auf OFF gesetzt ist, wird eine volle Protokolldatei nur dann entfernt, wenn sie nicht länger zur Recovery nach einem Systemabsturz benötigt wird, es sei denn, die Endlosprotokollierung ist aktiviert. In diesem Fall können die Protokolldateien stattdessen in den Archivprotokollpfad verschoben werden.

Der Prozess der Zuordnung neuer Protokolldateien und des Entfernens alter Protokolldateien ist von den Einstellungen der Datenbankkonfigurationsparameter *logarchmeth1* und *logarchmeth2* abhängig:

logarchmeth1 **und** *logarchmeth2* **sind auf `OFF` gesetzt**
: Es wird Umlaufprotokollierung verwendet. Bei der Umlaufprotokollierung wird die Recovery nach einem Systemabsturz unterstützt, die aktualisierende Recovery jedoch nicht.

  Während der Umlaufprotokollierung werden außer sekundären Protokollen keine neuen Protokolldateien erstellt, und alte Protokolldateien werden nicht gelöscht. Die Protokolldateien werden umlaufend verwendet. Das heißt, wenn die letzte Protokolldatei voll ist, schreibt der Datenbankmanager wieder in die erste Protokolldatei.

  Wenn alle Protokolldateien aktiv sind und der Prozess der Umlaufprotokollierung nicht erneut in die erste Protokolldatei schreiben kann, gelten alle Protokolldateien als voll. Wenn alle primären Protokolldateien aktiv und voll sind, werden sekundäre Protokolldateien erstellt. Sekundäre Protokolldateien werden gelöscht, wenn der von ihnen belegte Speicherbereich für die aktiven Protokolldateien benötigt wird.

logarchmeth1 **oder** *logarchmeth2* **wird auf `LOGRETAIN` gesetzt.**
: Es wird Archivprotokollierung verwendet. Die Datenbank ist eine wiederherstellbare Datenbank. Sowohl die Recovery nach einem Systemabsturz als auch die aktualisierende Recovery sind aktiviert. Der Datenbankmanager verwaltet nicht die Protokolldateien. Nachdem Sie die Protokolldateien archiviert haben, müssen Sie sie aus dem Pfad für aktive Protokolldateien löschen, sodass der Plattenspeicherplatz für neue Protokolldateien verwendet werden kann. Welche der Protokolldateien archivierte Protokolldateien sind, können Sie am Wert des Datenbankkonfigurationsparameters *loghead* erkennen. Dieser Parameter gibt das Protokoll mit der niedrigsten Nummer an, das aktiv ist. Bei Protokollen mit Folgenummern, die niedriger als der Wert von *loghead* sind, handelt es sich um nicht aktive Protokolldateien, die archiviert und entfernt werden können.

logarchmeth1 **oder** *logarchmeth2* **ist auf einen anderen Wert als `OFF` oder `LOGRETAIN` gesetzt**
: Es wird Archivprotokollierung verwendet. Die Datenbank ist eine wiederherstellbare Datenbank. Sowohl die Recovery nach einem Systemabsturz als auch die aktualisierende Recovery sind aktiviert. Wenn eine Protokolldatei voll ist, wird sie automatisch vom Datenbankmanager archiviert.

  Protokolldateien werden in der Regel nicht gelöscht. Wenn eine neue Protokolldatei benötigt wird, aber keine verfügbar ist, wird stattdessen eine Archivprotokolldatei umbenannt und erneut verwendet. Eine Archivprotokolldatei wird nicht umbenannt oder gelöscht, nachdem sie geschlossen und in das Verzeichnis für Archivprotokolldateien kopiert wurde. Der Datenbankmanager wartet, bis eine neue Protokolldatei benötigt wird und be-

nennt dann die älteste Archivprotokolldatei um. Eine Protokolldatei, die während der Recovery in das Datenbankverzeichnis versetzt wurde, wird während des Recoveryprozesses entfernt, sobald sie nicht mehr benötigt wird.

Wenn beim Archivieren der Protokolldateien ein Fehler auftritt, wird die Archivierung für den vom Datenbankkonfigurationsparameter *archretrydelay* angegebenen Zeitraum ausgesetzt. Sie können auch mit dem Datenbankkonfigurationsparameter *numarchretry* angeben, wie oft der Datenbankmanager versuchen soll, eine Protokolldatei im primären oder sekundären Archivverzeichnis zu archivieren, bevor es versucht, die Protokolldateien im Funktionsübernahmeverzeichnis (das vom Datenbankkonfigurationsparameter *failarchpath* angegeben wird) zu archivieren. *Numarchretry* wird nur verwendet, wenn der Datenbankkonfigurationsparameter *failarchpath* definiert ist. Wenn *numarchretry* auf 0 gesetzt ist, wird der Datenbankmanager kontinuierlich versuchen, die Protokolldateien im primären oder sekundären Verzeichnispfad zu archivieren.

Die einfachste Möglichkeit, alte Protokolldateien zu entfernen, ist ein Neustart der Datenbank. Nach dem erneuten Start der Datenbank befinden sich im Datenbankverzeichnis nur noch neue Protokolldateien und Protokolldateien, die vom Datenbankmanager nicht archiviert werden konnten.

Nach einem erneuten Start der Datenbank entspricht die Mindestanzahl Protokolle im Datenbankprotokollverzeichnis der Anzahl primärer Protokolle, die mit dem Datenbankkonfigurationsparameter *logprimary* konfiguriert werden kann. Im Protokollverzeichnis können sich mehr Protokolldateien als die maximale Anzahl primärer Protokolle befinden. Dies kann auftreten, wenn zum Zeitpunkt des Herunterfahrens der Datenbank die Anzahl leerer Protokolle im Protokollverzeichnis größer ist als der Wert des Konfigurationsparameters *logprimary* zum Zeitpunkt des erneuten Datenbankstarts. Dies ist der Fall, wenn der Wert des Konfigurationsparameters *logprimary* zwischen dem Herunterfahren und dem erneuten Start der Datenbank geändert wird oder wenn sekundäre Protokolle zugeordnet, aber nicht benutzt wurden.

Wenn beim erneuten Start einer Datenbank die Anzahl leerer Protokolle kleiner als die vom Konfigurationsparameter *logprimary* angegebene Anzahl primärer Protokolle ist, werden zusätzliche Protokolldateien zugeordnet, um die Differenz auszugleichen. Falls im Datenbankverzeichnis mehr leere Protokolle als primäre Protokolle verfügbar sind, kann die Datenbank mit der Anzahl der im Datenbankverzeichnis vorhandenen, leeren Protokolle erneut gestartet werden. Nach dem Herunterfahren der Datenbank verbleiben die erstellten, sekundären Protokolldateien bei einem erneuten Start der Datenbank im Pfad für aktive Protokolldateien.

Einschließen von Protokolldateien in ein Backup-Image

Wenn Sie eine Online-Backup-Operation ausführen, können Sie angeben, ob die für einen Restore und eine Recovery erforderlichen Protokolldateien in das Backup-Image aufgenommen werden. Dies bedeutet, dass Sie die Protokolldateien nicht separat senden oder selbst packen müssen, wenn Sie Backup-Images an einen Standort senden, an dem eine Recovery nach einem Katastrophenfall durchgeführt werden muss. Außerdem müssen Sie so nicht selbst entscheiden, welche Protokolldateien erforderlich sind, um die Konsistenz eines Online-Backups zu gewährleisten. Dies bietet einen gewissen Schutz vor dem Löschen von Protokolldateien, die für eine erfolgreiche Recovery erforderlich sind.

Wenn Sie diese Funktion verwenden wollen, geben Sie die Option INCLUDE LOGS des Befehls BACKUP DATABASE an. Wenn Sie diese Option angeben, schneidet das Backup-Dienstprogramm die aktive Protokolldatei ab und kopiert die erforderlichen Protokollspeicherbereiche in das Backup-Image.

Verwenden Sie zum Wiederherstellen der Protokolldateien aus einem Backup-Image die Option LOGTARGET des Befehls RESTORE DATABASE, und geben Sie einen vollständig qualifizierten, auf dem DB2-Server vorhandenen Pfad an. Das Datenbankrestoredienstprogramm schreibt anschließend die Protokolldateien aus dem Image in den Zielpfad. Wenn eine Protokolldatei desselben Namens bereits im Zielpfad vorhanden ist, schlägt die Restoreoperation fehl, und es wird ein Fehler zurückgegeben. Wird die Option LOGTARGET nicht angegeben, werden keine Protokolldateien aus dem Backup-Image wiederhergestellt.

Wenn die Option LOGTARGET angegeben wird und das Backup-Image keine Protokolldateien enthält, wird ein Fehler zurückgegeben, bevor versucht wird, Tabellenbereichsdaten wiederherzustellen. Die Restoreoperation schlägt ebenfalls fehl, wenn ein ungültiger oder schreibgeschützter Pfad angegeben wird. Wenn während des Restores einer Datenbank oder eines Tabellenbereichs, für den die Option LOGTARGET angegeben wurde, mindestens eine Protokolldatei nicht extrahiert werden kann, schlägt die Restoreoperation fehl, und es wird ein Fehler zurückgegeben.

Sie können auch nur die in einem Backup-Image gespeicherten Protokolldateien wiederherstellen. Geben Sie dazu die Option LOGS mit der Option LOGTARGET des Befehls RESTORE DATABASE an. Wenn beim Restore von Protokolldateien in diesem Modus Fehler auftreten, schlägt die Restoreoperation fehl, und es wird ein Fehler zurückgegeben.

Bei einer Operation zum automatischen inkrementellen Restore werden nur die Protokolle aus dem Backup-Image abgerufen, die sich im Zielimage der Restoreoperation befinden. Protokolle, die sich in temporären Images befinden, auf die während des inkrementellen Restores verwiesen wird, werden nicht aus diesen Backup-Images extrahiert. Wenn Sie beim manuellen inkrementellen Restore eines Backup-Images, das Protokolldateien enthält, ein Protokollzielverzeichnis angeben, werden die in diesem Backup-Image enthaltenen Protokolldateien wiederhergestellt.

Wenn Sie eine aktualisierende Recovery einer Datenbank ausführen, die aus einem Online-Backup-Image, das Protokolldateien enthält, mit Restore wiederherstellt wurde, empfangen Sie möglicherweise den Fehler SQL1268N. Dieser Fehler gibt an, dass die aktualisierende Recovery aufgrund eines Fehlers beim Abruf eines Protokolls gestoppt wurde. Dieser Fehler wird generiert, wenn das Zielsystem, auf dem Sie den Restore des Backup-Images versuchen, keinen Zugriff auf die Einrichtung hat, die vom Quellensystem zum Archivieren der Transaktionsprotokolle verwendet wird.

Wenn Sie die Option INCLUDE LOGS des Befehls BACKUP DATABASE beim Backup einer Datenbank angeben und nachfolgend eine Restoreoperation und eine aktualisierende Recoveryoperation durchführen, die dieses Backup-Image verwenden, sucht DB2 bei der aktualisierenden Recovery auch dann noch nach weiteren Transaktionsprotokollen, wenn das Backup-Image Protokolle enthält. Es gehört zur Standardfunktionsweise der aktualisierenden Recovery, dass die Suche nach weiteren Transaktionsprotokollen fortgesetzt wird, bis keine Protokolle mehr gefunden werden. Es ist möglich, mehr als eine Protokolldatei mit der gleichen Zeitmarke zu haben. Infolgedessen wie DB2 nicht gestoppt, sobald die erste Zeitmarke angetrof-

fen wird, die mit dem für die aktualisierende Recovery der Datenbank angegebenen Zeitpunkt übereinstimmt, da es noch weitere Protokolldateien geben könnte, die ebenfalls diese Zeitmarke haben. Vielmehr setzt DB2 die Suche im Transaktionsprotokoll fort, bis eine Zeitmarke gefunden wird, die größer ist als der angegebene Zeitpunkt.

Wenn keine weiteren Protokolle zu finden sind, wird die aktualisierende Recoveryoperation erfolgreich beendet. Wenn während der Suche nach weiteren Transaktionsprotokolldateien jedoch ein Fehler auftritt, wird die Fehlernachricht SQL1268N zurückgegeben. Der Fehler SQL1268N kann dadurch verursacht werden, dass beim anfänglichen Restore bestimmte Konfigurationsparameter der Datenbank zurückgesetzt oder überschrieben wurden. Drei dieser Datenbankkonfigurationsparameter sind die TSM-Parameter TSM_NODENAME, TSM_OWNER und TSM_PASSWORD. Sie werden alle auf NULL zurückgesetzt. Zur Durchführung einer aktualisierenden Recovery bis zum Ende der Protokolle müssen Sie vor der aktualisierenden Recoveryoperation die Datenbankkonfigurationsparameter so ändern, dass sie dem Quellensystem entsprechen. Alternativ können Sie auch die Option NORETRIEVE im Befehl ROLLFORWARD DATABASE angeben. Dadurch wird verhindert, dass das DB2-Datenbanksystem versucht, potenziell fehlende Transaktionsprotokolle von anderen Positionen abzurufen.

Anmerkung:

1. Diese Funktion wird für Offline-Backups nicht unterstützt.
2. Wenn in ein Online-Backup-Image Protokolldateien eingeschlossen werden, kann das daraus resultierende Image nicht in Releases der DB2-Datenbank wiederhergestellt werden, die älter als Version 8.2 sind.

Zufälligen Verlust von Protokolldateien verhindern

In Situationen, in denen Sie eine Datenbank löschen oder eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt durchführen müssen, verlieren Sie möglicherweise Protokolldateien, die für zukünftige Recoveryoperationen erforderlich sind. In solchen Fällen ist es wichtig, Kopien aller im aktuellen Protokollverzeichnispfad der Datenbank vorhandenen Protokolle zu erstellen.

Betrachten Sie die folgenden Szenarien:

- Wenn Sie vor einer RESTORE-Operation eine Datenbank löschen wollen, müssen Sie die im Pfad für aktive Protokolldateien vorhandenen Protokolldateien speichern, bevor Sie den Befehl DROP DATABASE absetzen. Nach dem Wiederherstellen der Datenbank werden diese Protokolldateien möglicherweise für die aktualisierende Recovery benötigt, da einige dieser Protokolldateien eventuell vor dem Löschen der Datenbank nicht archiviert wurden. In der Regel ist es nicht erforderlich, eine Datenbank vor dem Absetzen des Befehls RESTORE zu löschen. Es ist jedoch möglich, dass Sie die Datenbank löschen müssen (oder die Datenbank in einer Datenbankpartition löschen müssen, indem Sie die Option AT NODE mit dem Befehl DROP DATABASE angeben), da sie beschädigt wurde, sodass der Befehl RESTORE fehlschlägt. Sie können eine Datenbank auch löschen, um vor der Restoreoperation einen Neuanfang zu machen.
- Wenn Sie eine Datenbank bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen, werden alle Protokolldaten nach der von Ihnen angegebenen Zeitmarke überschrieben. Wenn Sie nach Beendigung der aktualisierenden Recovery bis zu einem bestimmten Zeitpunkt und der erneuten Verbindungsherstellung zur Datenbank feststellen, dass Sie die Datenbank eigentlich bist zu einem späteren Zeitpunkt hätten wiederherstellen müssen, können Sie dies nicht mehr nachholen, da die Protokolle überschrieben wurden. Zwar ist es möglich, dass

die ursprünglichen Protokolldateien archiviert wurden; DB2 könnte jedoch ein Benutzerexitprogramm aufrufen, um die neu generierten Protokolldateien automatisch zu archivieren. Je nachdem, wie das Benutzerexitprogramm geschrieben ist, könnten hierbei die ursprünglichen Protokolldateien im Archivprotokollverzeichnis überschrieben werden. Auch wenn sich die ursprünglichen Protokolldateien zusammen mit den neuen Protokolldateien im Archivprotokollverzeichnis befinden (als verschiedene Versionen derselben Dateien), müssen Sie möglicherweise bestimmen, welche Protokolle nun für spätere Recoveryoperationen verwendet werden sollen.

Beeinträchtigung der Verfügbarkeit durch Verwaltungs- und Wartungsaktivitäten minimieren

In Ihrer DB2-Datenbanklösung werden Sie für Ihre Geschäftsaktivitäten auch Verwaltungs- und Wartungsaktivitäten wie Software- oder Hardware-Upgrades, Optimierung der Datenbankleistung, Datenbankbackups, Statistikerfassung und Überwachung durchführen müssen. Sollen die Auswirkungen dieser Verwaltungs- und Wartungsaktivitäten auf die Verfügbarkeit Ihrer Lösung minimiert werden, ist sorgfältige Terminplanung für die Offlineverwaltung notwendig, sowie der Einsatz von DB2-Komponenten und -Funktionalitäten, um die Beeinträchtigung der Verfügbarkeit durch die Offlineverwaltung reduzieren.

Die beiden folgenden Aufgaben sind zu erledigen, bevor Sie die anschließenden Schritte ausführen können, um die Beeinträchtigung der Verfügbarkeit Ihrer DB2-Datenbanklösung durch Verwaltungs- und Wartungsaktivitäten zu minimieren:

- Konfigurieren der automatischen Verwaltung und
- Installieren der High Availability Disaster Recovery-Funktion (HADR-Funktion).

1. Die automatische Verwaltung erledigt die Verwaltungsaktivitäten für Sie.

   Die DB2-Datenbank kann viele Datenbankverwaltungsaktivitäten automatisieren. Sobald die automatische Verwaltung konfiguriert wurde, finden die Verwaltungsaktivitäten statt, ohne dass Sie dafür zusätzliche Schritte unternehmen müssten.
2. Verwenden Sie den schrittweisen Upgrade von DB2 High Availability Disaster Recovery (HADR), um eine Beeinträchtigung anderer Verwaltungsaktivitäten zu minimieren.

   Wenn Sie für Software oder Hardware ein Upgrade durchführen oder einige Konfigurationsparameter des Datenbankmanagers modifizieren möchten, können Sie diese Änderungen mit der HADR-Funktion bei nur minimaler Unterbrechung der Verfügbarkeit ausführen. Diese durch HADR ermöglichte reibungslose Änderung wird als schrittweises Upgrade bezeichnet.

   Bei einigen Verwaltungsaktivitäten ist es jedoch selbst in der HADR-Umgebung notwendig, dass Sie eine Datenbank beenden, bevor Sie die Verwaltungsarbeit durchführen können. Unter bestimmten Umständen unterscheidet sich die Prozedur zum Beenden einer HADR-Datenbank geringfügig von der Prozedur zum Beenden einer Standarddatenbank: Wenn eine HADR-Datenbank von einer Clientanwendung gestartet wird, die eine Verbindung zu ihr herstellt, müssen Sie den Befehl DEACTIVATE DATABASE verwenden.

Stoppen von DB2 High Availability Disaster Recovery (HADR)

Wenn Sie die DB2 High Availability Disaster Recovery-Funktion (HADR-Funktion) verwenden, kann die Verwaltung der beiden DB2-Datenbanksystem, d. h. der primären und der Bereitschaftsdatenbank, komplizierter sein, als das Verwalten eines Standalone-Datenbankservers. Falls Sie HADR für die Verwaltungsarbeit stoppen

müssen, verwenden Sie dazu den Befehl STOP HADR. Wenn Sie die Verwaltungsarbeit nur in der Bereitschaftsdatenbank ausführen, müssen Sie HADR auch nur auf der Bereitschaftsdatenbank stoppen. Wenn Sie HADR vollständig stoppen möchten, können Sie dies auf beiden Datenbanken tun.

Warnung: Wenn Sie die angegebene Datenbank stoppen möchten, diese jedoch ihre Rolle als Primär- oder Bereitschaftsdatenbank in einer HADR-Konfiguration behalten soll, verwenden Sie den Befehl STOP HADR nicht. Wenn Sie den Befehl STOP HADR ausführen, wird die Datenbank zu einer Standarddatenbank und erfordert möglicherweise eine Reinitialisierung, wenn sie den Betrieb als HADR-Datenbank wieder aufnehmen soll. Führen Sie stattdessen den Befehl DEACTIVATE DATABASE aus.

Sie können den Befehl STOP HADR nur für eine Primärdatenbank oder eine Bereitschaftsdatenbank ausführen. Wenn Sie diesen Befehl für eine Standarddatenbank absetzen, wird ein Fehler zurückgegeben.

Sie können die HADR über den Befehlszeilenprozessor (CLP), in der Steuerzentrale über das Fenster 'HADR verwalten' oder über die Anwendungsprogrammierschnittstelle (API) db2HADRStop stoppen.

Wenn Sie HADR-Operationen in der Primärdatenbank oder der Bereitschaftsdatenbank über den CLP stoppen wollen, setzen Sie den Befehl STOP HADR für die Datenbank ab, in der HADR-Operationen gestoppt werden sollen.

Im folgenden Beispiel werden HADR-Operationen in der Datenbank SOCKS gestoppt.

```
STOP HADR ON DATABASE SOCKS
```

Wenn Sie diesen Befehl für eine inaktive Primärdatenbank absetzen, wird die Datenbank zu einer Standarddatenbank und bleibt offline.

Wenn Sie diesen Befehl für eine inaktive Bereitschaftsdatenbank absetzen, wird die Datenbank zu einer Standarddatenbank, erhält den Status *Aktualisierende Recovery anstehend* und bleibt offline.

Wenn Sie diesen Befehl für eine aktive Primärdatenbank absetzen, werden keine Protokolle mehr an die Bereitschaftsdatenbank übertragen, und in der Bereitschaftsdatenbank werden alle HADR-EDUs (EDU - Engine Dispatchable Unit) heruntergefahren. Die Datenbank wird zu einer Standarddatenbank und bleibt offline. Die Transaktionsverarbeitung kann fortgesetzt werden. Sie können den Befehl START HADR mit der Option AS PRIMARY verwenden, um der Datenbank wieder die Rolle einer Primärdatenbank zuzuweisen.

Wenn Sie diesen Befehl für eine aktive Bereitschaftsdatenbank ausführen, wird eine Fehlernachricht zurückgegeben, die angibt, dass Sie die Bereitschaftsdatenbank inaktivieren müssen, bevor Sie sie in eine Standarddatenbank konvertieren können.

Gehen Sie wie folgt vor, um das Fenster **HADR stoppen** zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie die Datenbank finden, für die HADR verwaltet werden soll. Klicken Sie die Datenbank mit der rechten Maustaste an, und klicken Sie im Kontextmenü die Option **HADR (High Availability Disaster Recovery) → Verwalten** an. Das Fenster **HADR verwalten** wird geöffnet.
2. Klicken Sie **HADR stoppen** an. Das Fenster **HADR stoppen** wird geöffnet.

3. Wenn Sie HADR nur für eine Datenbank stoppen möchten, wählen Sie das Markierungsfeld für die andere Datenbank ab.
4. Wenn nur eine Datenbank gestartet ist (entweder die Primär- oder die Bereitschaftsdatenbank), wird der Name der Datenbank im Fenster **HADR stoppen** angezeigt.
5. Klicken Sie OK an. Das Fenster wird geschlossen. Möglicherweise wird ein Statusanzeigefeld für die Befehlsausführung geöffnet. Nach Abschluss der Befehlsausführung werden Sie benachrichtigt, ob der Befehl erfolgreich ausgeführt wurde oder nicht.

   Weitere Informationen enthält die Kontexthilfefunktion der Steuerzentrale.

Aktivierung und Inaktivierung von Datenbanken in einer DB2-HADR-Umgebung

Wenn eine Standarddatenbank durch eine Clientverbindung gestartet wird, wird die Datenbank gestoppt, wenn der letzte Client die Verbindung trennt. Wenn eine HADR-Primärdatenbank durch eine Clientverbindung gestartet wird, ist dieser Vorgang zum Starten der Datenbank über den Befehl ACTIVATE DATABASE äquivalent. Zum Stoppen einer HADR-Primärdatenbank, die durch eine Clientverbindung gestartet wurde, müssen Sie explizit den Befehl DEACTIVATE DATABASE ausführen.

In einer Standarddatenbank im Status *aktualisierende Recovery anstehend* sind die Befehle ACTIVATE DATABASE und DEACTIVATE DATABASE nicht anwendbar. Sie können nur die aktualisierende Recovery (ROLLFORWARD) fortsetzen, die aktualisierende Recovery stoppen oder die Datenbank mit dem Befehl START HADR als HADR-Bereitschaftsdatenbank starten. Wenn eine Datenbank als HADR-Bereitschaftsdatenbank gestartet ist, können Sie die Befehle ACTIVATE DATABASE und DEACTIVATE DATABASE zum Starten und Stoppen der Datenbank verwenden.

Zum Aktivieren einer Primärdatenbank stehen die folgenden Methoden zur Verfügung:

- Herstellen einer Clientverbindung
- Befehl ACTIVATE DATABASE
- Befehl START HADR mit der Option AS PRIMARY

Zum Inaktivieren einer Primärdatenbank stehen die folgenden Methoden zur Verfügung:

- Befehl DEACTIVATE DATABASE
- Befehl DB2STOP mit der Option FORCE

Zum Aktivieren einer Bereitschaftsdatenbank stehen die folgenden Methoden zur Verfügung:

- Befehl ACTIVATE DATABASE
- Befehl START HADR mit der Option AS STANDBY

Zum Inaktivieren einer Bereitschaftsdatenbank stehen die folgenden Methoden zur Verfügung:

- Befehl DEACTIVATE DATABASE
- Befehl DB2STOP mit der Option FORCE

Ausführen von schrittweisen Aktualisierungen und Upgrades in einer DB2-HADR-Umgebung

Verwenden Sie diese Prozedur in einer HADR-Umgebung (High Availability Disaster Recovery), wenn Sie einen Upgrade der Software (Betriebssystem oder DB2-Datenbanksystem) oder Hardware ausführen bzw. Änderungen an Datenbankkonfigurationsparametern vornehmen. Durch diese Prozedur bleibt der Datenbankservice während des Upgradeprozesses verfügbar, wobei nur beim Wechsel von einer Datenbank zur nächsten eine kurze Serviceunterbrechung auftritt. Da HADR eine optimale Leistung bringt, wenn sich die Primärdatenbank und die Bereitschaftsdatenbank auf identischen Systemen befinden, sollten Sie Änderungen so schnell wie möglich auf beide Systeme anwenden.

Anmerkung: Alle Fixpacks und Upgrades für das DB2-Datenbanksystem sollten zuerst in einer Testumgebung implementiert werden, bevor Sie sie auf Ihr Produktionssystem anwenden.

Das HADR-Paar sollte sich im Peerstatus befinden, bevor der schrittweise Upgrade gestartet wird.

Mit dieser Prozedur können Sie keine Migration von einer früheren auf eine spätere Version eines DB2-Datenbanksystems durchführen; Sie können diese Prozedur beispielsweise nicht zum Migrieren eines Datenbanksystems von Version 8 auf Version 9 verwenden. Mit dieser Prozedur können Sie Ihr Datenbanksystem nur von einer Modifikationsstufe auf eine andere aktualisieren, z. B. indem Sie ein Fixpack anwenden.

Diese Prozedur funktioniert nicht, wenn Sie die DB2-HADR-Konfigurationsparameter aktualisieren. Aktualisierungen an HADR-Konfigurationsparametern sollten getrennt vorgenommen werden. Da HADR voraussetzt, dass die Parameter in der Primär- und der Bereitschaftsdatenbank identisch sind, kann dies erfordern, dass die Primär- und die Bereitschaftsdatenbank zur gleichen Zeit inaktiviert und aktualisiert werden.

Gehen Sie wie folgt vor, um einen schrittweisen Upgrade in einer HADR-Umgebung auszuführen:

1. Führen Sie einen Upgrade für das System aus, auf dem sich die Bereitschaftsdatenbank befindet:
   a. Fahren Sie die Bereitschaftsdatenbank mit dem Befehl DEACTIVATE DATABASE herunter.
   b. Falls erforderlich, fahren Sie die Instanz in der Bereitschaftsdatenbank herunter.
   c. Nehmen Sie Änderungen an der Software, Hardware oder den DB2-Konfigurationsparametern vor.

      Anmerkung: Sie können keine HADR-Konfigurationsparameter ändern, wenn Sie einen schrittweisen Upgrade ausführen.
   d. Falls erforderlich, starten Sie die Instanz in der Bereitschaftsdatenbank erneut.
   e. Starten Sie die Bereitschaftsdatenbank mit dem Befehl ACTIVATE DATABASE erneut.
   f. Stellen Sie sicher, dass die Bereitschaftsdatenbank den Peerstatus erhält. Überprüfen Sie dies mit dem Befehl GET SNAPSHOT.

2. Tauschen Sie die Rollen der Primärdatenbank und die Bereitschaftsdatenbank:
   a. Setzen Sie den Befehl TAKEOVER HADR für die Bereitschaftsdatenbank ab.
   b. Leiten Sie Clients an die neue Primärdatenbank um. Sie können dazu die automatische Clientweiterleitung verwenden.

      Anmerkung: Da die Bereitschaftsdatenbank die Rolle der Primärdatenbank übernimmt, wird nun ein Upgrade für die neue Primärdatenbank ausgeführt. Wenn Sie ein Fixpack für ein DB2-Datenbanksystem anwenden, ändert der Befehl TAKEOVER HADR die Rolle der ursprünglichen Primärdatenbank in die der Bereitschaftsdatenbank. Der Befehl erstellt jedoch keine Verbindung zwischen der neuen Bereitschaftsdatenbank und der aktualisierten Primärdatenbank. Da die neue Bereitschaftsdatenbank eine ältere Version des DB2-Datenbanksystems verwendet, kann sie möglicherweise die neuen von der aktualisierten Primärdatenbank generierten Protokollsätze nicht verstehen und wird heruntergefahren. Damit die neue Bereitschaftsdatenbank wieder eine Verbindung zur neuen Primärdatenbank herstellt (d. h., damit das HADR-Paar erneut gebildet werden kann), muss auch die neue Bereitschaftsdatenbank aktualisiert werden.
3. Führen Sie einen Upgrade für die ursprüngliche Primärdatenbank (die aktuelle Bereitschaftsdatenbank) aus, und verwenden Sie dazu die oben in Schritt 1 beschriebene Prozedur. Wenn Sie dies ausgeführt haben, sind beide Datenbanken aktualisiert und stellen im HADR-Peerstatus eine Verbindung zueinander her. Das HADR-System stellt den vollständigen Datenbankservice und einen vollständigen Hochverfügbarkeitsschutz zur Verfügung.
4. Optional. Wenn Sie zu Ihrer ursprünglichen Konfiguration zurückkehren wollen, tauschen Sie die Rollen der Primärdatenbank und die Bereitschaftsdatenbank, wie in Schritt 2 beschrieben.

Schrittweises Upgrade in einer automatisierten HADR-Umgebung (HADR = High Availability Disaster Recovery)

Wenn Sie das integrierte High Availability Feature (HA) zur Automatisierung von HADR verwenden, sind zusätzliche Schritte für das Upgrade der Software (Betriebssystem oder DB2-Datenbanksystem), für das Upgrade der Hardware oder für Änderungen der Datenbankkonfigurationsparameter erforderlich. Gehen Sie wie folgt vor, um ein schrittweises Upgrade in einer automatisierten HADR-Umgebung durchzuführen.

Vorbereitung

Die folgenden Voraussetzungen müssen erfüllt sein, damit die unter 'Vorgehensweise' beschriebenen Schritte ausgeführt werden können:

- Es müssen zwei DB2-Instanzen vorhanden sein (in diesem Fall haben sie den Namen 'stevera' auf jedem Knoten).
- Es müssen zwei Knoten vorhanden sein ('grom04' und 'grom03'). Auf der 'grom04'-Maschine befindet sich zu Beginn die HADR-Primärdatenbank.
- Die Instanzen müssen ursprünglich mit dem Code von DB2 Version 9.5 GA ausgeführt werden.
- Die Instanzen müssen mit der integrierten HA-Steuerung der HADR-Funktionsübernahme konfiguriert sein. Die Clusterdomäne heißt 'test'.

Anmerkung: Alle Fixpacks und Upgrades für das DB2-Datenbanksystem sollten zuerst in einer Testumgebung implementiert werden, bevor Sie sie auf Ihr Produktionssystem anwenden.

Das HADR-Paar sollte sich im Peerstatus befinden, bevor der schrittweise Upgrade gestartet wird.

Informationen zu dieser Task

Einschränkungen

Mit dieser Prozedur können Sie keine Migration von einer früheren auf eine spätere Version eines DB2-Datenbanksystems durchführen; Sie können diese Prozedur beispielsweise nicht zum Migrieren eines Datenbanksystems von Version 8 auf Version 9 verwenden. Mit dieser Prozedur können Sie Ihr Datenbanksystem nur von einer Modifikationsstufe auf eine andere aktualisieren, z. B. indem Sie ein Fixpack anwenden.

Diese Prozedur funktioniert nicht, wenn Sie die DB2-HADR-Konfigurationsparameter aktualisieren. Aktualisierungen an HADR-Konfigurationsparametern sollten getrennt vorgenommen werden. Da HADR voraussetzt, dass die Parameter in der Primär- und der Bereitschaftsdatenbank identisch sind, kann dies erfordern, dass die Primär- und die Bereitschaftsdatenbank zur gleichen Zeit inaktiviert und aktualisiert werden.

Vorgehensweise

1. Anfangsstatus des Systems anzeigen:

```
root@grom03:# lsrpdomain
Name   OpState   RSCTActiveVersion   MixedVersions   TSPort   GSPort
-----  --------  ------------------  -------------   -------  -------
test   Online    2.4.7.1             No              12347    12348

root@grom03:# lssam
Online IBM.ResourceGroup:db2_stevera_grom03_0-rg Nominal=Online
        '- Online IBM.Application:db2_stevera_grom03_0-rs
                '- Online IBM.Application:db2_stevera_grom03_0-rs:grom03
Online IBM.ResourceGroup:db2_stevera_grom04_0-rg Nominal=Online
        '- Online IBM.Application:db2_stevera_grom04_0-rs
                '- Online IBM.Application:db2_stevera_grom04_0-rs:grom04
Online IBM.ResourceGroup:db2_stevera_stevera_SVTDB-rg Nominal=Online
        |- Online IBM.Application:db2_stevera_stevera_SVTDB-rs
                |- Offline IBM.Application:db2_stevera_stevera_SVTDB-rs:grom03
                '- Online IBM.Application:db2_stevera_stevera_SVTDB-rs:grom04
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:grom03
                '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:grom04

root@grom03:# lsrpnode
Name     OpState   RSCTVersion
-----    -------   ------------
grom03   Online    2.4.7.1
grom04   Online    2.4.7.1
```

   Das oben dargestellte Beispiel zeigt, dass Sie ein Upgrade für die Bereitschaftsinstanz auf 'grom03' durchführen müssen. Stoppen Sie hierzu alle Ressourcengruppen, die sich auf 'grom03' befinden.

2. Stoppen aller Ressourcengruppen auf dem Bereitschaftsknoten und Bestätigen der Änderung:

```
root@grom03:# chrg -o Offline db2_stevera_grom03_0-rg

root@grom03:# lssam –g db2_stevera_grom03_0-rg
Offline IBM.ResourceGroup:db2_stevera_grom03_0-rg Nominal=Offline
        '- Offline IBM.Application:db2_stevera_grom03_0-rs
                '- Offline IBM.Application:db2_stevera_grom03_0-rs:grom03
```

3. Stoppen des Clusterknotens (Bereitschaftsknotens) und Bestätigen der Änderung:

```
root@grom03:# stoprpnode grom03

root@grom03:# lsrpdomain
Name   OpState   RSCTActiveVersion   MixedVersions   TSPort   GSPort
----   -------   -----------------   -------------   ------   ------
test   Offline   2.4.7.1             No              12347    12348
```

4. Installieren des DB2-Fixpacks auf dem Bereitschaftsknoten:

   Optional: Unter AIX ist möglicherweise die Installation der RSCT-Voraussetzungen für das betreffende Fixpack erforderlich.

```
root@grom03:# ./installFixPack -b /opt/ibm/db2/V9.5
DBI1017I  installFixPack aktualisiert das bzw. die an der Speicherposition
          '/opt/ibm/db2/V9.5' installierte(n) DB2-Produkt(e).
```

   Die Installation des DB2-Fixpacks wird gestartet.

5. Starten des Knotens und Versetzen der Ressourcengruppe in den Onlinemodus:

   Wenn die Installation erfolgreich abgeschlossen ist.

```
root@grom03:# startrpdomain test

root@grom03:# lsrpdomain
Name   OpState   RSCTActiveVersion   MixedVersions   TSPort   GSPort
-----  --------  ------------------  --------------  -------  -------
test   Online    2.4.7.1             Yes             12347    12348

root@grom03:# chrg -o Online db2_stevera_grom03_0-rg
```

6. Verifizieren, dass das Fixpack angewendet wurde und sich HADR wieder im Peerzustand befindet:

```
stevera@grom03% db2level

stevera@grom03% db2pd –hadr –db SVTDB
```

7. TAKEOVER-Operation durchführen:

   Für ein Upgrade des anderen Knotens (in diesem Fall 'grom04') führen Sie die TAKEOVER-Operation so aus, dass sich die HADR-Primärdatenbank auf dem Knoten 'grom03' befindet.

```
root@grom03:# su - stevera

stevera@grom03% db2 takeover hadr on db SVTDB
DB20000I  Der Befehl TAKEOVER HADR ON DATABASE wurde erfolgreich ausgeführt.

root@grom03:# lssam
Online IBM.ResourceGroup:db2_stevera_grom03_0-rg Nominal=Online
        '- Online IBM.Application:db2_stevera_grom03_0-rs
                '- Online IBM.Application:db2_stevera_grom03_0-rs:grom03
Online IBM.ResourceGroup:db2_stevera_grom04_0-rg Nominal=Online
        '- Online IBM.Application:db2_stevera_grom04_0-rs
                '- Online IBM.Application:db2_stevera_grom04_0-rs:grom04
Online IBM.ResourceGroup:db2_stevera_stevera_SVTDB-rg Nominal=Online
        |- Online IBM.Application:db2_stevera_stevera_SVTDB-rs
                |- Online IBM.Application:db2_stevera_stevera_SVTDB-rs:grom03
                '- Offline IBM.Application:db2_stevera_stevera_SVTDB-rs:grom04
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:grom03
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:grom04
```

8. Durchführen eines Upgrades auf dem Knoten 'grom04':

```
root@grom03:# ssh root@grom04

root@grom04:# chrg -o Offline db2_stevera_grom04_0-rg
```

```
root@grom04:# lssam –g db2_stevera_grom04_0-rg
Offline IBM.ResourceGroup:db2_stevera_grom04_0-rg Nominal=Offline
        '- Offline IBM.Application:db2_stevera_grom04_0-rs
                '- Offline IBM.Application:db2_stevera_grom04_0-rs:grom04

root@grom04:# stoprpnode grom04
```

Optional: Unter AIX ist möglicherweise die Installation der RSCT-Voraussetzungen für das betreffende Fixpack erforderlich.

```
root@grom04:# ./installFixPack -b /opt/ibm/db2/V9.5
DBI1017I  installFixPack aktualisiert das bzw. die an der Speicherposition
          '/opt/ibm/db2/V9.5' installierte(n) DB2-Produkt(e).
```

Die Installation des DB2-Fixpacks wird gestartet.

Wenn die Installation erfolgreich abgeschlossen ist.

```
root@grom04:# lsrpdomain
Name   OpState   RSCTActiveVersion   MixedVersions   TSPort   GSPort
-----  --------  ------------------  --------------  -------  -------
test   Offline   2.4.7.1             Yes             12347    12348

root@grom04:# startrpdomain test

root@grom04:# lsrpdomain
Name   OpState   RSCTActiveVersion   MixedVersions   TSPort   GSPort
-----  --------  ------------------  --------------  -------  -------
test   Online    2.4.7.1             Yes             12347    12348

root@grom04:# chrg -o Online db2_stevera_grom04_0-rg
```

9. Verifizieren, dass das Fixpack angewendet wurde (durch Ausführen von 'db2level') und dass sich HADR im Peerzustand befindet (durch Ausführen von 'db2pd –hadr –db svtdb'):

```
root@grom04:# su - stevera

stevera@grom04% db2pd -hadr -db svtdb

Database Partition 0 -- Database SVTDB -- Active -- Up 0 days 00:00:05

HADR-Information:
Role     State    SyncMode    HeartBeatsMissed    LogGapRunAvg (bytes)
-----    ------   ---------   -----------------   --------------------
Standby  Peer     Sync        0                   0

ConnectStatus   ConnectTime                          Timeout
--------------  ------------                         --------
Connected       Tue May 5 13:20:58 2009 (1241544058)  120

PeerWindowEnd                          PeerWindow
-------------                          -----------
Tue May  5 13:25:58 2009 (1241544358)  300

LocalHost                              LocalService
----------                             -------------
grom04                                 55555

RemoteHost                             RemoteService      RemoteInstance
-----------                            --------------     ---------------
grom03                                 55555              stevera

PrimaryFile    PrimaryPg    PrimaryLSN
------------   ----------   -----------
S0000001.LOG   1            0x0000000003389487

StandByFile    StandByPg    StandByLSN          StandByRcvBufUsed
------------   ----------   -----------         -----------------
S0000001.LOG   1            0x0000000003389487  0%
```

```
root@grom04:# lssam
Online IBM.ResourceGroup:db2_stevera_grom03_0-rg Nominal=Online
        '- Online IBM.Application:db2_stevera_grom03_0-rs
                '- Online IBM.Application:db2_stevera_grom03_0-rs:grom03
Online IBM.ResourceGroup:db2_stevera_grom04_0-rg Nominal=Online
        '- Online IBM.Application:db2_stevera_grom04_0-rs
                '- Online IBM.Application:db2_stevera_grom04_0-rs:grom04
Online IBM.ResourceGroup:db2_stevera_stevera_SVTDB-rg Nominal=Online
        |- Online IBM.Application:db2_stevera_stevera_SVTDB-rs
                |- Online IBM.Application:db2_stevera_stevera_SVTDB-rs:grom03
                '- Offline IBM.Application:db2_stevera_stevera_SVTDB-rs:grom04
        '- Online IBM.ServiceIP:db2ip_9_26_124_22-rs
                |- Online IBM.ServiceIP:db2ip_9_26_124_22-rs:grom03
                '- Offline IBM.ServiceIP:db2ip_9_26_124_22-rs:grom04
```

10. Migration der TSA-Domäne:

```
root@grom04:# export CT_MANAGEMENT_SCOPE=2

root@grom04:# runact -c IBM.PeerDomain CompleteMigration Options=0
Resource Class Action Response for CompleteMigration

root@grom04:# samctrl -m

Ready to Migrate! Are you Sure? [Y|N]:.

Y
```

11. Sicherstellen, dass der Wert für 'MixedVersions' nicht mehr 'Yes' für die Clusterkomponente lautet:

```
root@grom04:# lsrpdomain
Name   OpState   RSCTActiveVersion   MixedVersions   TSPort   GSPort
-----  --------  ------------------  --------------  -------  -------
test   Online    2.5.1.2             No              12347    12348
```

12. Sicherstellen, dass die AVN (Active Version Number) mit der IVN (Installed Version Number) für den HA-Manager übereinstimmt:

```
root@grom04:# lssrc –ls IBM.RecoveryRM |grep VN
   Our IVN         : 2.2.0.7
   Our AVN         : 2.2.0.7
```

13. Optional: Durchführen einer TAKEOVER-Operation als Instanzeigner 'stevera' auf der 'grom04'-Maschine, um 'grom04' zur HADR-Primärdatenbank zu machen (wie ursprünglich).

Verwenden einer geteilten Spiegeldatenbank zum Klonen einer Datenbank

Verwenden Sie zum Erstellen einer Klondatenbank die folgende Prozedur. Klondaten werden allgemein für Aktivitäten verwendet, bei denen nur Lesezugriff erforderlich ist (z. B. das Ausführen von Berichten), es ist jedoch auch möglich, in Klondatenbanken zu schreiben.

Sie können die geklonte Datenbank weder sichern, noch ihr Image auf dem Originalsystem wiederherstellen oder es mithilfe der Protokolldateien, die auf dem Originalsystem generiert wurden, aktualisierend wiederherstellen. Sie können die Option AS SNAPSHOT verwenden; allerdings wird dadurch lediglich eine unmittelbare Kopie der Datenbank bereitgestellt, wenn die E/A ausgesetzt wird; sämtliche weitere ausstehende und nicht festgeschriebene Arbeit wird rückgängig gemacht, nachdem der Befehl db2inidb für den Klon ausgeführt wurde.

Gehen Sie wie folgt vor, um eine Datenbank zu klonen:

1. Setzen Sie für die primäre Datenbank die E/A aus:

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

Führen Sie keine weiteren Dienstprogramme oder Tools aus, während sich die Datenbank im ausgesetzten Status befindet. Das Erstellen einer Datenbankkopie sollte die einzige Aktivität sein.

2. Verwenden Sie geeignete Befehle auf Betriebssystemebene, um die Spiegeldatenbank(en) aus der primären Datenbank zu erstellen.

   Hinweis: Stellen Sie sicher, dass Sie das gesamte Datenbankverzeichnis einschließlich des Datenträgerverzeichnisses kopieren. Sie müssen auch das Protokollverzeichnis und alle Containerverzeichnisse kopieren, die außerhalb des Datenbankverzeichnisses vorhanden sind. Eine Zusammenstellung dieser Informationen finden Sie in der Verwaltungssicht DBPATHS, in der alle Dateien und Verzeichnisse der Datenbank angezeigt werden, die geteilt werden müssen.

3. Nehmen Sie den E/A-Betrieb auf der primären Datenbank wieder auf:

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

4. Katalogisieren Sie die gespiegelte Datenbank auf dem sekundären System.

   Hinweis: Eine gespiegelte Datenbank kann standardmäßig nicht in demselben System vorhanden sein wie die Primärdatenbank. Sie muss sich auf einem sekundären System mit der gleichen Verzeichnisstruktur befinden und denselben Instanznamen als primäre Datenbank verwenden. Wenn die gespiegelte Datenbank auf demselben System als Primärdatenbank vorhanden sein muss, können Sie das Dienstprogramm db2relocatedb oder die Option RELOCATE USING des Befehls db2inidb für diese Aufgabe verwenden.

5. Starten Sie die Datenbankinstanz auf dem sekundären System:

   ```
   db2start
   ```

6. Initialisieren Sie die gespiegelte Datenbank auf dem sekundären System:

   ```
   db2inidb aliasname-der-datenbank as snapshot
   ```

   Falls erforderlich, geben Sie die Option RELOCATE USING des Befehls d2inidb an, um die Klondatenbank zu verlagern:

   ```
   db2inidb aliasname-der-datenbank as snapshot relocate using relocatedbcfg.txt
   ```

   'relocatedbcfg.txt' ist die Datei, die die zum Verlagern der Datenbank erforderlichen Informationen enthält.

   Hinweise:

   a. Mit diesem Befehl werden Transaktionen rückgängig gemacht, die während des Teilens gerade verarbeitet werden, und es wird eine neue Protokollkettenreihenfolge begonnen, sodass Protokolle aus der primären Datenbank nicht für die Klondatenbank wiederholt werden können.

   b. Das Datenbankverzeichnis (einschließlich des Datenträgerverzeichnisses), das Protokollverzeichnis und die Containerverzeichnisse müssen an die gewünschte Position versetzt werden, bevor Sie die Option RELOCATE USING verwenden.

Szenario: Ändern der Systemuhr

Beim Einstellen oder Ändern der Systemuhr muss DB2 nicht gestoppt werden. DB2 übernimmt problemlos zwei Mal pro Jahr die erfolgreiche Sommerzeitumstellung auf der gesamten Welt. Konfigurationen, die zum systemübergreifenden Synchronisieren von Uhren NTP verwenden, werden ebenfalls vollständig unterstützt.

Informationen zu dieser Task

Beim Ändern der Systemzeit sind einige empfohlene Methoden zu berücksichtigen.

Einschränkungen

Das Ändern der Systemuhr wirkt sich in den meisten Szenarios in keiner Weise auf andere Prozesse aus.

Bei starken Zeitverschiebungen sollten Sie sich jedoch zwei Situationen bewusst sein.

- Beim Ausführen einer punktuellen Recovery müssen Sie auf alle signifikanten Zeitverschiebungen achten.
- Die Funktionsdefinitionen enthalten Zeit und Datum ihrer Erstellung im Format einer Zeitmarke. Bei Funktionsaufruf versucht DB2, die Funktionsdefinition aufzulösen. Im Rahmen der Funktionsauflösung wird der zur Erstellungszeit in der Funktionsdefinition protokollierte Zeitmarkenwert geprüft. Wenn Sie diese Systemuhr auf eine Zeit vor der Erstellung der Funktionen zurücksetzen, löst DB2 keine Referenzen auf diese Funktionen auf.

Vorgehensweise

Empfohlene Methoden zur Vermeidung dieser beiden Situationen:

1. Wenn Sie die Zeit vorstellen, fahren Sie mit Schritt 3 fort.
2. Wenn Sie die Zeit um X Minuten zurückstellen, gehen Sie wie folgt vor:
   a. Wählen Sie zum Ausführen der Änderung einen Zeitpunkt aus, bei dem feststeht, dass in den vergangenen X Minuten keine neuen Funktionen erstellt wurden und dass in den folgenden X Minuten keine Aktualisierungstransaktionen erfolgen werden.
   b. Wenn Sie keinen geeigneten Zeitpunkt wie unter a. beschrieben finden, können Sie die Systemuhr dennoch um X Minuten zurückstellen, während DB2 online ist. Dabei müssen Sie allerdings mit den folgenden Auswirkungen rechnen:
      - Sie können möglicherweise nicht die punktuelle Recovery verwenden, um eine Wiederherstellung auf einen Zeitpunkt innerhalb dieser X Minuten durchzuführen. Das heißt, ein Teil der innerhalb dieser X Minuten ausgeführten Aktualisierungstransaktionen kann möglicherweise nicht wiederhergestellt werden.
      - Funktionen, die innerhalb von X Minuten vor der Änderung erstellt wurden, können X Minuten nach der Änderung nicht aufgelöst werden.
3. Ändern Sie die Systemuhr.

Ergebnisse

Wenn Sie die empfohlenen Methoden wie beschrieben anwenden, vermeiden Sie beim Ändern der Systemuhr alle potenziellen Probleme hinsichtlich punktueller Recovery oder Funktionsauflösung.

Synchronisation der Primär- und der Bereitschaftsdatenbank

Eine der Strategien für hohe Verfügbarkeit ist es, über eine Primärdatenbank und eine sekundäre oder Bereitschaftsdatenbank zu verfügen; letztere übernimmt die anfallenden Operationen, falls die Primärdatenbank ausfällt. Muss die Bereitschaftsdatenbank die Datenbankoperationen für eine ausgefallene Primärdatenbank übernehmen, müssen in ihr exakt die gleichen Daten enthalten sein, sämtliche unvollständigen Transaktionen müssen ihr bekannt sein, und sie muss ansonsten auf exakt die gleiche Art die Datenbankverarbeitung fortführen, als ob der primäre Datenbankserver nicht ausgefallen wäre. Der fortlaufende Prozess der Aktualisierung der Bereitschaftsdatenbank, sodass sie eine Kopie der Primärdatenbank darstellt, wird als Synchronisation bezeichnet.

Sie müssen zunächst die folgenden Aufgaben erledigen, bevor Sie die primäre und Bereitschaftsdatenbank synchronisieren können:

- Erstellen und Konfigurieren der primären und der Bereitschaftsdatenbank.
- Konfigurieren der Datenübertragung zwischen der primären und der Bereitschaftsdatenbank.
- Auswählen einer Synchronisationsstrategie (z. B. Protokollübertragung, Protokollspiegelung, ausgesetzte E/A und Plattenspiegelung oder HADR).

  Es gibt verschiedene Strategien, um den primären Datenbankserver und den Bereitschaftsdatenbankserver synchronisiert zu halten:

  – Das Übertragen von Protokollen von der Primärdatenbank an die Bereitschaftsdatenbank sowie anschließend eine aktualisierende Recovery der Protokolle in der Bereitschaftsdatenbank
  – Das gleichzeitige Schreiben von Datenbankprotokollen in die Primär- und die Bereitschaftsdatenbank, auch als Protokollspiegelung bezeichnet
  – Verwenden der Unterstützung für ausgesetzte E/A zusammen mit der Plattenspiegelung, um regelmäßig eine Kopie der Primärdatenbank zu erstellen, den Spiegel zu teilen und die Kopie als neuen Bereitschaftsdatenbankserver zu initialisieren
  – Das Verwenden einer Funktion für die Verfügbarkeit, wie z. B. DB2 High Availability Disaster Recovery (HADR), sodass die Primär- und die Bereitschaftsdatenbank immer synchronisiert sind.

1. Wenn Sie Protokolle verwenden, um die Primärdatenbank und die sekundäre bzw. die Bereitschaftsdatenbank zu synchronisieren, konfigurieren Sie die DB2-Datenbank so, dass sie die erforderliche Protokollierungsverwaltung für Sie ausführt. Beispiel: Wenn Sie möchten, dass die DB2-Datenbank die Protokolle spiegelt, setzen Sie den Konfigurationsparameter MIRRORLOGPATH auf die Speicherposition, in der die Kopie der Protokolle gespeichert werden soll.
2. Wenn Sie die Funktionalität für ausgesetzte E/A Ihrer DB2-Datenbank verwenden, um eine gespiegelte Platte der Primärdatenbank zu teilen, müssen Sie wie folgt vorgehen:
   a. Initialisieren Sie die Plattenspiegelung für die Primärdatenbank.
   b. Wenn Sie den Spiegel der Primärdatenbank teilen müssen, befolgen Sie die Anweisungen im Abschnitt „Verwenden einer geteilten Spiegeldatenbank als Bereitschaftsdatenbank".
3. Wenn Sie die HADR-Funktion verwenden, um die Synchronisation der Primär- und der Bereitschaftsdatenbank zu verwalten, konfigurieren Sie die DB2-Datenbank für HADR und richten Sie es so ein, dass die DB2-Datenbank das Synchronisieren der Primär- und der Bereitschaftsdatenbank für Sie übernimmt.

Replizierte Operationen von DB2 High Availability Disaster Recovery (HADR)

DB2 High Availability Disaster Recovery (HADR) verwendet Datenbankprotokolle, um Daten aus der Primärdatenbank in der Bereitschaftsdatenbank zu replizieren. Einige Aktivitäten können dazu führen, dass die Bereitschaftsdatenbank nicht auf dem aktuellen Stand der Primärdatenbank bleibt, da die Protokolle auf der Bereitschaftsdatenbank angewendet werden. Einige Aktivitäten werden so ausführlich protokolliert, dass die dadurch generierte umfangreiche Zahl an Protokolldateien Speicherprobleme verursachen kann. Obwohl das Replizieren von Daten in der Bereitschaftsdatenbank mithilfe von Protokollen zentraler Bestandteil von Verfügbarkeitsstrategien ist, kann die Protokollierung an sich potenziell negative Auswirkungen auf die Verfügbarkeit Ihrer Lösung haben. Gestalten Sie Ihre Verwaltungs-

strategie so optimal wie möglich, und konfigurieren Sie Ihr System so, dass negative Auswirkungen der Protokollierung minimiert werden und die Protokollierung Ihre Transaktionsdaten schützt.

Wenn Sie HADR verwenden, werden die folgenden Operationen aus der Primärdatenbank in die Bereitschaftsdatenbank repliziert:

- Datendefinitionssprache (DDL - Data Definition Language)
- Datenbearbeitungssprache (DML - Data Manipulation Language)
- Pufferpooloperationen
- Tabellenbereichsoperationen
- Onlinereorganisation
- Offlinereorganisation
- Metadaten für gespeicherte Prozeduren und benutzerdefinierte Funktionen (UDF - User-defined Function), jedoch ohne zugehörige Objekt- oder Bibliotheksdateien

Während einer Onlinereorganisation werden alle Operationen detailliert protokolliert. Daher kann HADR die Operation replizieren, ohne dass die Bereitschaftsdatenbank weiter in Rückstand geraten würde, als dies bei herkömmlichen Datenbankaktualisierungen der Fall wäre. Dieses Verhalten kann potenziell jedoch große Auswirkungen auf das System haben, da eine große Anzahl Protokollsätze generiert wird.

Während Offlinereorganisationen nicht so umfassend protokolliert werden wie Onlinereorganisationen, werden Operationen in der Regel für hunderte oder tausende von betroffenen Zeilen protokolliert. Dies bedeutet, dass die Bereitschaftsdatenbank in Rückstand geraten kann, da sie auf jeden Protokollsatz wartet und anschließend gleichzeitig zahlreiche Aktualisierungen wiederholt. Wenn es sich bei Offlinereorganisation nicht um eine Clusterreorganisation handelt, wird nach der gesamten Reorganisation ein einzelner Protokollsatz generiert. Dieser Modus hat die größte Auswirkung darauf, ob und wie die Bereitschaftsdatenbank auf dem Stand der Primärdatenbank gehalten werden kann. Die Bereitschaftsdatenbank führt die gesamte Reorganisation aus, nachdem sie den Protokollsatz von der Primärdatenbank erhalten hat.

HADR repliziert weder gespeicherte Prozeduren noch UDF-Objektdateien oder Bibliotheksdateien. Sie müssen die Dateien in der Primärdatenbank und in der Bereitschaftsdatenbank in identischen Pfaden erstellen. Wenn die Bereitschaftsdatenbank das Referenzobjekt oder die Bibliotheksdatei nicht finden kann, wird der Aufruf der gespeicherten Prozedur oder UDF in der Bereitschaftsdatenbank fehlschlagen.

Nicht replizierte Operationen von DB2 High Availability Disaster Recovery (HADR)

DB2 High Availability Disaster Recovery (HADR) verwendet Datenbankprotokolle, um Daten aus der Primärdatenbank in die Bereitschaftsdatenbank zu replizieren. Nicht protokollierte Operationen sind in der Primärdatenbank zwar zulässig, sie werden aber nicht in die Bereitschaftsdatenbank repliziert. Wenn Sie möchten, dass nicht protokollierte Operationen wie z. B. Aktualisierungen der Verlaufsdatei in der Bereitschaftsdatenbank abgebildet werden, sind dafür zusätzliche Schritte erforderlich.

Die folgenden Beispiele beschreiben Fälle, in denen Operationen in der Primärdatenbank nicht in die Bereitschaftsdatenbank repliziert werden:

- Tabellen, die mit der Option NOT LOGGED INITIALLY erstellt werden, werden nicht repliziert. Versuche, auf solche Tabellen zuzugreifen, nachdem eine HADR-Bereitschaftsdatenbank die Funktion als primäre Datenbank übernommen hat, enden mit einem Fehler.
- Alle protokollierten LOB-Spalten werden repliziert. Nicht protokollierte LOB-Spalten werden nicht repliziert. Der Speicherbereich für sie wird in der Bereitschaftsdatenbank jedoch zugeordnet, indem binäre Nullen als Werte für die Spalte verwendet werden.
- Aktualisierungen an der Datenbankkonfiguration, die mit den Befehlen UPDATE DATABASE CONFIGURATION und UPDATE DATABASE MANAGER CONFIGURATION ausgeführt werden, werden nicht repliziert.
- Datenbankkonfigurationsparameter und Datenbankmanagerkonfigurationsparameter werden nicht repliziert.
- Bei benutzerdefinierten Funktionen (UDFs) werden Änderungen an Objekten, die für die Datenbank extern sind (z. B. zugehörige Objekte und Bibliotheksdateien) nicht repliziert. Solchen Änderungen muss in der Bereitschaftsdatenbank mit anderen Mitteln Rechnung getragen werden.
- Die Datei des Recoveryprotokolls (db2rhist.asc) und daran vorgenommene Änderungen werden nicht automatisch von der Primärdatenbank an die Bereitschaftsdatenbank übertragen.

  Sie können eine Anfangskopie der Datei des Recoveryprotokolls (die Sie aus dem Backup-Image der Primärdatenbank abrufen) in die Bereitschaftsdatenbank kopieren, indem Sie den Befehl RESTORE DATABASE mit der Option REPLACE HISTORY FILE absetzen:

  ```
  RESTORE DB KELLY REPLACE HISTORY FILE
  ```

  Nachdem HADR initialisiert wurde und nachfolgend in der Primärdatenbank Backup-Aktivitäten erfolgt sind, befindet sich die Datei des Recoveryprotokolls in der Bereitschaftsdatenbank nicht mehr auf dem aktuellen Stand. Eine Kopie der Datei des Recoveryprotokolls wird jedoch in jedem Backup-Image gespeichert. Sie können die Datei des Recoveryprotokolls in der Bereitschaftsdatenbank aktualisieren, indem Sie die Datei des Recoveryprotokolls mithilfe des folgenden Befehls aus einem Backup-Image extrahieren:

  ```
  RESTORE DB KELLY HISTORY FILE
  ```

  Verwenden Sie nicht die regulären Betriebssystembefehle, um die Datei des Recoveryprotokolls im Datenbankverzeichnis aus der Primärdatenbank in die Bereitschaftsdatenbank zu kopieren. Die Datei des Recoveryprotokolls kann beschädigt werden, wenn die Primärdatenbank die Datei gerade ändert, wenn die Kopie ausgeführt wird.

  Wenn eine Übernahmeoperation ausgeführt wird und die Bereitschaftsdatenbank über eine aktuelle Datei des Recoveryprotokolls verfügt, generieren Backup- und Restoreoperationen in der neuen Primärdatenbank neue Sätze in der Datei des Recoveryprotokolls, die sich nahtlos in die von der ursprünglichen Primärdatenbank generierten Sätze einfügen. Wenn die Datei des Recoveryprotokolls nicht auf dem neuesten Stand ist oder Einträge fehlen, kann eventuell kein automatischer inkrementeller Restore ausgeführt werden. In diesem Fall ist ein manueller inkrementeller Restore erforderlich.

Status von DB2-HADR-Bereitschaftsdatenbanken (High Availability Disaster Recovery)

Die Bereitschaftsdatenbank befindet sich immer in einem von fünf möglichen Status: Lokales Catch-up, Fernes Catch-up anstehend, Fernes Catch-up, Peer und Unterbrochener Peer. Der Status, in dem sich die Bereitschaftsdatenbank befindet, bestimmt, welche Operationen sie ausführen kann. Mit dem Befehl GET SNAPSHOT können Sie feststellen, in welchem Status sich Ihre Bereitschaftsdatenbank gerade befindet.

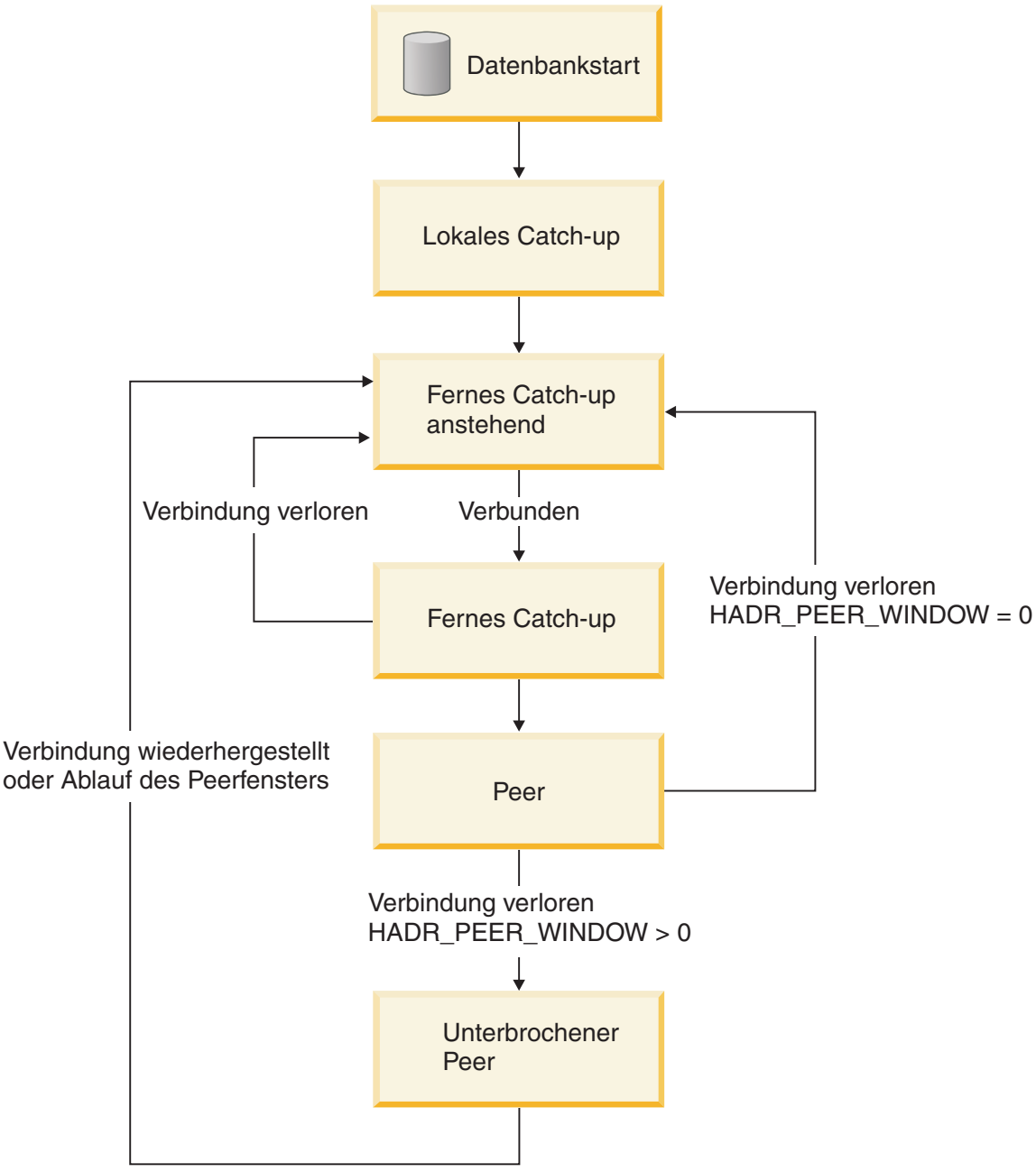


Abbildung 9. Status der Bereitschaftsdatenbank

Start der Datenbank, Lokales Catch-up und Fernes Catch-up anstehend

Wenn Sie die HADR-Funktion verwenden und die Bereitschaftsdatenbank gestartet wird, erhält sie den Status *Lokales Catch-up* und versucht, die Protokolldateien in ihrem lokalen Protokollpfad zu lesen. Wenn die Datenbank keine Protokolldatei in ihrem lokalen Protokollpfad findet und eine Protokollarchivierungsmethode angegeben wurde, wird die Protokolldatei unter Verwendung der angegebenen Methode abgerufen. Nachdem die Protokolldateien gelesen wurden, werden sie auf die Bereitschaftsdatenbank angewendet. Während dieser Zeit ist keine Verbindung zur Primärdatenbank erforderlich. Wenn keine Verbindung vorhanden ist, versucht die Bereitschaftsdatenbank jedoch, eine Verbindung zur Primärdatenbank herzustellen. Wird das Ende der lokalen Protokolldateien erreicht, wechselt die Bereitschaftsdatenbank in den Status *Fernes Catch-up anstehend*.

Wenn nach dem Wechseln der Bereitschaftsdatenbank in den Status 'Fernes Catch-up anstehend' Protokolldateien verfügbar werden, können Sie die Bereitschaftsdatenbank herunterfahren und erneut starten, um zu veranlassen, dass sie wieder den Status 'Lokales Catch-up' erhält. Diese Vorgehensweise ist nützlich, wenn ein lokaler Zugriff auf diese Protokolldateien auf der Bereitschaftsdatenbank effizienter ist, als die Dateien mit HADR über das Netzwerk aus der Primärdatenbank zu kopieren.

Fernes Catch-up anstehend, Fernes Catch-up, Peer

Die Bereitschaftsdatenbank behält den Status 'Fernes Catch-up anstehend', bis eine Verbindung zur Primärdatenbank aufgebaut ist. Anschließend wechselt sie in den Status 'Fernes Catch-up'. Währenddessen liest die Primärdatenbank Protokolldaten aus ihrem Protokollpfad oder unter Verwendung einer Protokollarchivierungsmethode und sendet diese Protokolldaten an die Bereitschaftsdatenbank. Die Bereitschaftsdatenbank empfängt die Protokolldaten und wendet diese an. Die Primär- und die Bereitschaftsdatenbank wechseln in den Peerstatus, wenn die Bereitschaftsdatenbank alle Protokolldateien empfangen hat, die sich auf der Platte der Maschine mit der Primärdatenbank befinden.

Im Peerstatus werden Protokollseiten jedes Mal an die Bereitschaftsdatenbank übertragen, wenn die Primärdatenbank eine Protokollseite auf Platte schreibt. Die Protokollseiten werden in die lokalen Protokolldateien in der Bereitschaftsdatenbank geschrieben, um sicherzustellen, dass die Protokolldateifolgen der Primärdatenbank und die Bereitschaftsdatenbank identisch sind. Die Protokollseiten können anschließend auf die Bereitschaftsdatenbank angewendet werden.

Geht die Verbindung zwischen der Primärdatenbank und der Bereitschaftsdatenbank verloren, wenn sich die Datenbanken im Status 'Fernes Catch-up' befinden, wechselt die Bereitschaftsdatenbank in den Status 'Fernes Catch-up anstehend'. Geht die Verbindung zwischen der Primärdatenbank und der Bereitschaftsdatenbank verloren, wenn sich die Datenbanken im Peerstatus befinden und der Datenbankkonfigurationsparameter HADR_PEER_WINDOW entweder nicht oder auf Null gesetzt ist, wechselt die Bereitschaftsdatenbank in den Status 'Fernes Catch-up anstehend'. Geht die Verbindung zwischen der Primärdatenbank und der Bereitschaftsdatenbank jedoch verloren, wenn sich die Datenbanken im Peerstatus befinden und der Datenbankkonfigurationsparameter HADR_PEER_WINDOW auf einen Wert ungleich Null gesetzt ist, wechselt die Bereitschaftsdatenbank in den Status 'Unterbrochener Peer'.

Unterbrochener Peer

Wenn Sie den Datenbankkonfigurationsparameter HADR_PEER_WINDOW für einen Zeitwert größer Null konfigurieren und die Primärdatenbank die Verbindung mit der Bereitschaftsdatenbank verliert, verhält sich die Primärdatenbank weiterhin so, als ob sich die Primär- und die Bereitschaftsdatenbank für den konfigurierten Zeitraum im Peerstatus befinden würden. Wenn die Verbindung zwischen der Primär- und der Bereitschaftsdatenbank getrennt wird und sich die Datenbanken weiterhin so verhalten, als ob sie sich im Peerstatus befinden würden, wird dieser Status als 'Unterbrochener Peer' bezeichnet. Der Zeitraum, in dem die Primärdatenbank im Status 'Unterbrochener Peer' verbleibt, nachdem die Verbindung zur Bereitschaftsdatenbank getrennt wurde, wird als Peerfenster bezeichnet. Wenn die Verbindung zur Bereitschaftsdatenbank wiederhergestellt wird oder das Peerfenster abläuft, verlässt die Bereitschaftsdatenbank den Status 'Unterbrochener Peer'.

Die Konfiguration eines Peerfensters hat den Vorteil, dass das Risiko eines Transaktionsverlusts niedriger ist, wenn mehrere Fehler auftreten oder Fehler weitergegeben werden. Ohne das Peerfenster verlässt die Primärdatenbank den Peerstatus, wenn die Verbindung mit der Bereitschaftsdatenbank verloren geht. Nach dem Trennen der Verbindung verarbeitet die Primärdatenbank Transaktionen unabhängig von der Bereitschaftsdatenbank. Wenn ein Fehler in der Primärdatenbank auftritt, während sich die Datenbank wie beschrieben nicht im Peerstatus befindet, können Transaktionen verloren gehen, weil diese auf der Bereitschaftsdatenbank nicht repliziert wurden. Bei einem konfigurierten Peerfenster betrachtet die Primärdatenbank eine Transaktion erst dann als festgeschrieben, wenn sie eine Empfangsbestätigung von der Bereitschaftsdatenbank erhält, dass die Protokolle entweder in den Hauptspeicher auf dem Bereitschaftssystem oder in Protokolldateien auf der Bereitschaftsdatenbank geschrieben wurden (abhängig vom HADR-Synchronisationsmodus).

Die Konfiguration eines Peerfensters hat den Nachteil, dass die Ausführung von Transaktionen auf der Primärdatenbank länger dauert oder sogar das Zeitlimit überschreiten kann, während die Primärdatenbank im Peerfenster auf eine Wiederherstellung der Verbindung mit der Bereitschaftsdatenbank oder den Ablauf des Peerfensters wartet.

Mit dem Befehl GET SNAPSHOT oder dem Dienstprogramm db2pd in Verbindung mit dem Parameter -hadr können Sie die Größe des Peerfensters ermitteln, die dem Wert des Datenbankkonfigurationsparameters HADR_PEER_WINDOW entspricht.

Status von Bereitschaftsdatenbanken - Auswirkungen und Einschränkungen bezüglich der Synchronisation von Primär- und Bereitschaftsdatenbank

Eine Methode zum Synchronisieren von Primär- und Bereitschaftsdatenbank besteht darin, die Protokolldateien der Primärdatenbank manuell in den Protokollpfad der Bereitschaftsdatenbank zu kopieren, um sie für lokales Catch-up bereitzustellen. Wenn Sie die Primär- und die Bereitschaftsdatenbank synchronisieren, indem Sie die Protokolle der Primärdatenbank manuell in den Protokollpfad der Bereitschaftsdatenbank kopieren, müssen Sie die Protokolldateien der Primärdatenbank vor dem Starten der Bereitschaftsdatenbank kopieren.

Das hat folgende Gründe:

1. Wenn das Ende der lokalen Protokolldateien erreicht wird, wechselt die Bereitschaftsdatenbank in den Status 'Fernes Catch-up anstehend' und versucht erst dann wieder, auf die lokalen Protokolldateien zuzugreifen, wenn die Bereitschaftsdatenbank erneut gestartet wird.
2. Wenn die Bereitschaftsdatenbank in den Status 'Fernes Catch-up' wechselt, überschneidet sich außerdem das Kopieren von Protokolldateien in den Protokollpfad mit dem Schreiben von lokalen Protokolldateien der Bereitschaftsdatenbank.

Bestimmen des Status von HADR-Bereitschaftsdatenbanken mit dem Befehl GET SNAPSHOT

Sie können den Status einer DB2 HADR-Bereitschaftsdatenbank (HADR = High Availability Disaster Recovery) bestimmen, indem Sie den Befehl GET SNAPSHOT mit der Option DATABASE ON absetzen.

Sie können den Befehl GET SNAPSHOT auf der Primär- oder der Bereitschaftsdatenbank absetzen, um den Status einer HADR-Bereitschaftsdatenbank in einem HADR-Datenbankpaar (Primär- und Bereitschaftsdatenbank) zu bestimmen.

- Wenn der Befehl GET SNAPSHOT auf der Bereitschaftsdatenbank abgesetzt wird, wird der Status der Bereitschaftsdatenbank im Feld `Status` der Ausgabe angezeigt.
- Wenn der Befehl GET SNAPSHOT auf einer Primärdatenbank abgesetzt wird, die mit der Bereitschaftsdatenbank verbunden ist, wird der Status der Bereitschaftsdatenbank im Feld `Status` der Ausgabe angezeigt.
- Wenn Sie den Befehl GET SNAPSHOT auf einer Primärdatenbank absetzen, die nicht mit der Bereitschaftsdatenbank verbunden ist, wird im Feld `Status` der Ausgabe der Status `Unterbrochen` angezeigt.

Den Status einer Bereitschaftsdatenbank MUSIC beispielsweise können Sie mit dem folgenden Befehl anzeigen:

```
get snapshot for database on music
```

Es folgt eine Beispielausgabe für den Abschnitt 'HADR-Status', der vom Befehl GET SNAPSHOT zurückgegeben wird:

```
HADR-Status

    Rolle                       = Primär
    Status                      = Peer
    Synchronisationsmodus       = Synchron
    Verbindungsstatus           = Verbunden, 11/03/2002 12:23:09.35092
    Fehlende Überwachungssignale = 0
    Lokaler Host                = host1.ibm.com
    Lokaler Service             = hadr_service
    Ferner Host                 = host2.ibm.com
    Ferner Service              = hadr_service
    Ferne Instanz               = dbinst2
    Zeitlimit (Sekunden)        = 120
    Position des Primärprotokolls (Datei, Seite, Protokollfolgenummer)
        = S0001234.LOG, 12, 0000000000BB800C
    Position des Bereitschaftsprotokolls (Datei, Seite, Protokollfolgenummer)
        = S0001234.LOG, 12, 0000000000BB800C
    Von benutzten Seiten belegter Protokollspeicherbereich (Byte)
        = 8723
```

Beim Überprüfen der Ausgabe des Befehls GET SNAPSHOT stellen Sie möglicherweise eine Protokollabstimmungsdiskrepanz fest. Wenn nämlich eine Protokolldatei abgeschnitten wird, sei es infolge einer expliziten Abschneideoperation für das Protokoll oder infolge eines Stoppens und Neustartens der Primärdatenbank, wechselt die Primärdatenbank zum Anfang der nächsten Protokolldatei. Die Bereitschaftsdatenbank bleibt jedoch am Ende der letzten Protokolldatei. Sobald die Primärdatenbank ein Protokoll schreibt, wird das Protokoll repliziert und die Bereitschaftsdatenbank aktualisiert ihre Protokollposition.

Verwaltung von DB2 High Availability Disaster Recovery (HADR)

Zur Verwaltung von DB2 High Availability Disaster Recovery (HADR) gehört das Konfigurieren und Verwalten des Status Ihres HADR-Systems.

Die HADR-Verwaltung umfasst u. a. die folgenden Tasks:

- „Initialisieren von High Availability Disaster Recovery (HADR)" auf Seite 27
- „Stoppen von DB2 High Availability Disaster Recovery (HADR)" auf Seite 176
- „Tauschen von Datenbankrollen bei High Availability Disaster Recovery (HADR)" auf Seite 207
- „Ausführen einer HADR-Funktionsübernahmeoperation" auf Seite 204
- „Überwachen von HADR (High Availability Disaster Recovery)" auf Seite 201
- Überprüfen oder Ändern der HADR-bezogenen Datenbankkonfigurationsparameter.
- Katalogisieren einer HADR-Datenbank (falls erforderlich).

Zur Verwaltung von HADR stehen die folgenden Möglichkeiten zur Verfügung:

- Befehlszeilenprozessor
- GUI-Tools der Steuerzentrale
- DB2-Administrator-API

DB2-HADR-Befehle

Die Funktion DB2 High Availability Disaster Recovery (HADR) umfasst u.a. komplexe Protokollierung, die Funktionsübernahme (Failover) und eine Recovery-Funktionalität für hoch verfügbare DB2 Datenbanklösungen. Trotz der Komplexität der Funktionalität, die HADR bietet, gibt es nur einige wenige Aktionen, deren Ausführung Sie HADR direkt befehlen müssen: HADR starten und stoppen sowie die Bereitschaftsdatenbank veranlassen, die Operationen als Primärdatenbank zu übernehmen.

Zur Verwaltung von High Availability Disaster Recover (HADR) werden drei Befehle zur Verfügung gestellt:

- START HADR (HADR starten)
- STOP HADR (HADR stoppen)
- TAKEOVER HADR (HADR übernehmen)

Zum Aufrufen dieser Befehle können Sie den Befehlszeilenprozessor oder die Administrator-API verwenden. Darüber hinaus können Sie diese Befehle über die grafischen Benutzerschnittstellen (GUIs) des Fensters **HADR verwalten** der Steuerzentrale aufrufen. Gehen Sie wie folgt vor, um das Fenster 'HADR verwalten' über die Steuerzentrale zu öfnen: Klicken Sie eine Datenbank mit der rechten Maustaste an, und klicken Sie anschließend 'High Availability Disaster Recovery-—>Verwalten' an.

Durch Ausführen des Befehls START HADR mit der Option AS PRIMARY oder AS STANDBY wird die Rolle der Datenbank in die angegebene Rolle geändert, sofern die Datenbank diese Rolle nicht bereits hat. Darüber hinaus aktiviert dieser Befehl die Datenbank, wenn sie nicht bereits aktiviert ist.

Der Befehl STOP HADR ändert eine HADR-Datenbank (Primär- oder Bereitschaftsdatenbank) in eine Standarddatenbank. Alle auf HADR bezogenen Datenbankkonfigurationsparameter bleiben unverändert, sodass sich die Datenbank problemlos als HADR-Datenbank reaktivieren lässt.

Der Befehl TAKEOVER HADR, der nur für die Bereitschaftsdatenbank ausgeführt werden kann, ändert die Bereitschaftsdatenbank in eine Primärdatenbank. Wenn Sie die Option BY FORCE nicht angeben, tauschen die Primärdatenbank und die Bereitschaftsdatenbank ihre Rollen. Wenn Sie die Option BY FORCE angeben, wechselt die Bereitschaftsdatenbank einseitig ihre Rolle und wird zur Primärdatenbank. In diesem Fall versucht die Bereitschaftsdatenbank die Transaktionsverarbeitung in der alten Primärdatenbank zu stoppen. Es gibt jedoch keine Garantie, dass die Transaktionsverarbeitung tatsächlich gestoppt wird. Verwenden Sie die Option BY FORCE zur Erzwingung einer Übernahmeoperation nur unter Bedingungen, in denen die Funktionsübernahme erforderlich ist. Stellen Sie so weit wie möglich sicher, dass die aktuelle Primärdatenbank definitiv ausgefallen ist, oder fahren Sie sie selbst herunter, bevor Sie den Befehl TAKEOVER HADR mit der Option BY FORCE ausführen.

Rollenwechsel einer HADR-Datenbank

Eine Datenbank kann dynamisch und wiederholt zwischen der Rolle der Primärdatenbank und einer Standarddatenbank umgeschaltet werden. Wenn die Datenbank online oder offline ist, können Sie sowohl den Befehl START HADR mit der Option AS PRIMARY als auch den Befehl STOP HADR ausführen.

Sie können eine Datenbank zwischen der Rolle als Bereitschaftsdatenbank und der Rolle als Standarddatenbank statisch umschalten. Mehrfach ist dies jedoch nur möglich, wenn die Datenbank im Status *aktualisierende Recovery anstehend* verbleibt. Sie können den Befehl START HADR mit der Option AS STANDBY ausführen, um eine Standarddatenbank in eine Bereitschaftsdatenbank umzuwandeln, während die Datenbank offline ist und sich im Status *aktualisierende Recovery anstehend* befindet. Verwenden Sie den Befehl STOP HADR, um eine Bereitschaftsdatenbank in eine Standarddatenbank umzuwandeln, während die Datenbank offline ist. Die Datenbank bleibt nach der Ausführung des Befehls STOP HADR im Status *aktualisierende Recovery anstehend*. Durch einen nachfolgend ausgeführten Befehl START HADR mit der Option AS STANDBY wird die Datenbank wieder in eine Bereitschaftsdatenbank geändert. Wenn Sie den Befehl ROLLFORWARD DATABASE mit der Option STOP ausführen, nachdem HADR in einer Bereitschaftsdatenbank gestoppt wurde, können Sie die Datenbank nicht wieder zu einer Bereitschaftsdatenbank machen. Da sich die Datenbank nicht mehr im Status *aktualisierende Recovery anstehend* befindet, können Sie sie wieder als Standarddatenbank verwenden. Dies wird als Erfassung einer Momentaufnahme der Bereitschaftsdatenbank bezeichnet. Nach der Umwandlung einer vorhandenen Bereitschaftsdatenbank in eine Standarddatenbank sollten Sie aus Gründen der hohen Verfügbarkeit in Betracht ziehen, eine neue Bereitschaftsdatenbank zu erstellen.

Zum Vertauschen der Rollen von Primär- und Bereitschaftsdatenbank führen Sie eine TAKEOVER-Operation ohne die Option BY FORCE aus.

Zur einseitigen Änderung der Bereitschaftsdatenbank in eine Primärdatenbank (d. h. ohne die Primärdatenbank zur Bereitschaftsdatenbank zu machen), verwenden Sie den Befehl TAKEOVER mit der Option BY FORCE. Nachfolgend kann die alte Primärdatenbank möglicherweise als neue Bereitschaftsdatenbank wieder integriert werden.

Die HADR-Rolle ist persistent. Wenn eine HADR-Rolle eingerichtet ist, bleibt sie als Merkmal der Datenbank bestehen, selbst wenn die DB2-Instanz zwischendurch wiederholt gestoppt und erneut gestartet bzw. die DB2-Datenbank inaktiviert und aktiviert wird.

Das Starten der Bereitschaftsdatenbank erfolgt asynchron

Wenn Sie den Befehl START HADR mit der Option AS STANDBY ausführen, gibt der Befehl die Steuerung zurück, sobald die relevanten EDUs (Engine Dispatchable Units, zuteilbare Einheiten der Steuerkomponente) erfolgreich gestartet wurden. Der Befehl wartet nicht darauf, dass die Bereitschaftsdatenbank eine Verbindung zur Primärdatenbank herstellt. Vielmehr wird die Primärdatenbank erst dann als gestartet betrachtet, wenn sie eine Verbindung zu einer Bereitschaftsdatenbank herstellt (sofern der Befehl START HADR in der Primärdatenbank nicht mit der Option BY FORCE ausgeführt wird). Wenn die Bereitschaftsdatenbank einen Fehler feststellt, zum Beispiel wenn die Verbindung von der Primärdatenbank verweigert wird, wurde der Befehl START HADR mit der Option AS STANDBY möglicherweise bereits erfolgreich beendet. Infolgedessen ist keine Benutzereingabeaufforderung vorhanden, an die HADR einen Fehleranzeiger zurückmelden könnte. Die HADR-Bereitschaftsdatenbank schreibt in solchen Fällen eine Nachricht in das DB2-Diagnoseprotokoll und fährt sich selbst herunter. Sie sollten den Status der HADR-Bereitschaftsdatenbank überwachen, um sicherzustellen, dass sie erfolgreich eine Verbindung zur HADR-Primärdatenbank herstellt.

Protokollanwendungsfehler, d. h. Fehler, die von der Bereitschaftsdatenbank bei der nachvollziehenden Anwendung von Protokollsätzen festgestellt werden, setzen die Bereitschaftsdatenbank ebenfalls außer Gefecht. Solche Fehler können zum Beispiel auftreten, wenn zur Erstellung eines Pufferpools nicht genügend Speicher verfügbar ist oder wenn bei der Erstellung eines Tabellenbereichs der Pfad nicht gefunden wird. Daher sollten Sie den Status der Bereitschaftsdatenbank kontinuierlich überwachen.

Führen Sie HADR-Befehle nicht über einen Client aus, der mit einem zur Clientweiterleitung eingerichteten Datenbankaliasnamen arbeitet

Wenn die automatische Clientweiterleitung eingerichtet ist, ist für den Datenbankserver ein alternativer Server vordefiniert, sodass Clientanwendungen bei minimaler Unterbrechung zwischen der Arbeit mit dem ursprünglichen Datenbankserver und dem alternativen Server umschalten können. Wenn ein Client in einer solchen Umgebung eine Verbindung über TCP herstellt, kann die tatsächliche Verbindung entweder zur ursprünglichen Datenbank oder zur alternativen Datenbank erfolgen. HADR-Befehle sind so implementiert, dass sie die Zieldatenbank durch die reguläre Verbindungslogik des Clients ermitteln. Wenn für die Zieldatenbank eine alternative Datenbank definiert ist, ist es daher schwierig, die Datenbank zu bestimmen, an der der Befehl tatsächlich ausgeführt wird.

Während es für einen SQL-Client keinen Unterschied macht, mit welcher Datenbank er verbunden wird, müssen HADR-Befehl auf eine bestimmte Datenbank angewendet werden. Um dieser Einschränkung Rechnung zu tragen, sollten HADR-Befehle lokal auf der Servermaschine ausgeführt werden, sodass die Clientweiterleitung umgangen wird (die Clientweiterleitung betrifft nur TCP/IP-Verbindungen).

HADR-Befehle müssen auf einem Server mit einer gültigen Lizenz ausgeführt werden

Die Befehle START HADR, STOP HADR und TAKEOVER HADR setzen voraus, dass eine gültige HADR-Lizenz auf dem Server installiert ist, auf dem sie ausgeführt werden. Wenn die Lizenz nicht vorhanden ist, schlagen diese Befehle fehl und geben einen befehlsspezifischen Fehlercode (SQL1767N, SQL1769N bzw. SQL1770N) zusammen mit dem Ursachencode 98 zurück. Zur Behebung des Problems müssen Sie entweder eine gültige HADR-Lizenz mithilfe des Befehls db2licm installieren oder eine Version des Servers installieren, deren Lieferumfang eine gültige HADR-Lizenz enthält.

Kapitel 5. Feststellen und Handhaben von Systemausfällen in einer Hochverfügbarkeitslösung

Die Implementierung einer Lösung mit hoher Verfügbarkeit verhindert nicht, dass Hardware- oder Softwarefehler auftreten. Das Vorhandensein redundanter Systeme und ein Funktionsübernahmemechanismus stellen jedoch sicher, dass Ihre Lösung Fehler feststellt und handhaben kann, und dass Verarbeitungsprozesse weitergeleitet werden, sodass Benutzeranwendungen immer noch funktionieren können.

Wenn ein Fehler auftritt, muss Ihre Datenbank wie folgt vorgehen:

1. Feststellen des Fehlers.

   Die Software für die Funktionsübernahme kann die Überwachungssignalfunktion verwenden, um die Verfügbarkeit von Systemkomponenten zu bestätigen. Ein Überwachungssignalmonitor überwacht die reguläre Kommunikation aller Komponenten im System. Wenn der Überwachungssignalmonitor von einer Komponente keine Signale mehr empfängt, signalisiert der Überwachungssignalmonitor dem System, dass die Komponente fehlgeschlagen ist.
2. Handhabung des Fehlers: Funktionsübernahme.
   a. Eine sekundäre Komponente muss identifiziert, online geschaltet und initialisiert werden, damit sie die Operationen der fehlgeschlagenen Komponente übernehmen kann.
   b. Verarbeitungsprozesse an die sekundäre Komponente weiterleiten.
   c. Entfernen der fehlgeschlagenen Komponente aus dem System.
3. Durchführen einer Recovery nach dem Fehler.

   Schlägt ein primärer Datenbankserver fehl, ist die oberste Priorität, die Clients an einen alternativen Server weiterzuleiten oder die Funktionsübernahme durch eine Bereitschaftsdatenbank zu beginnen, sodass die Clientanwendungen in ihrem Betrieb so wenig wie möglich unterbrochen werden. War die Funktionsübernahme erfolgreich, müssen Sie den Fehler auf dem fehlgeschlagenen Datenbankserver reparieren, sodass dieser wieder in die Lösung integriert werden kann. "Reparatur" kann ggf. einfach ein Neustart des Servers sein.
4. Rückkehr zum normalen Betrieb.

   Sobald das fehlgeschlagene Datenbanksystem repariert ist, müssen Sie es wieder in die Datenbanklösung integrieren. Sie können eine fehlgeschlagene Primärdatenbank als Bereitschaftsdatenbank für jene Datenbank reintegrieren, die die Funktion als Primärdatenbank übernommen hatte, als der Fehler auftrat. Sie könnten außerdem erzwingen, dass der reparierte Datenbankserver erneut seine vorherige Funktion als primärer Datenbankserver übernimmt.

Die DB2-Datenbank kann einige dieser Schritte für Sie ausführen. Beispiel:

- Das Monitorelement für Überwachungssignale hadr_heartbeat von DB2 High Availability Disaster Recovery (HADR) kann feststellen, dass eine Primärdatenbank ausgefallen ist.
- Die DB2-Clientweiterleitung kann Workload von einem fehlgeschlagenen Datenbankserver an einen sekundären Datenbankserver weiterleiten.
- Der DB2-Fehlermonitor kann eine Datenbankinstanz erneut starten, die unerwartet beendet wurde.

Protokoll mit Benachrichtigungen für die Systemverwaltung

Der DB2-Datenbankmanager schreibt die folgenden Arten von Informationen in das Protokoll mit Benachrichtigungen für die Systemverwaltung: den Status von DB2-Dienstprogrammen wie REORG und BACKUP, Fehler in Clientanwendungen, Änderungen von Serviceklassen, Lizenzierungsvorgänge, Protokolldateipfade und Speicherprobleme, Überwachungs- und Indexierungsvorgänge sowie Tabellenbereichsprobleme. Ein Datenbankadministrator kann diese Informationen zum Diagnostizieren von Problemen, zur Optimierung der Datenbank oder einfach zur Überwachung der Datenbank verwenden.

Nachrichten des Protokolls mit Benachrichtigungen für die Systemverwaltung werden ebenfalls in 'db2diag.log' unter Verwendung eines standardisierten Nachrichtenformats protokolliert.

Hinweisnachrichten bieten zusätzliche Informationen als Ergänzung zu angegebenen SQLCODE-Werten. Der Typ von Ereignis sowie der Detaillierungsgrad der Informationen, die erfasst werden, werden durch den Konfigurationsparameter NOTIFYLEVEL bestimmt.

Feststellen einer ungeplanten Betriebsunterbrechung

Bevor Sie den Fehler oder Ausfall einer Komponente beheben können, müssen Sie zunächst feststellen, dass die betreffende Komponente fehlgeschlagen ist. DB2 Data Server verfügt über mehrere Tools zum Überwachen des ordnungsgemäßen Zustands einer Datenbank oder ggf. zum Feststellen, dass eine Datenbank fehlgeschlagen ist. Sie können diese Tools so konfigurieren, dass sie Sie benachrichtigen oder dass sie vordefinierte Maßnahmen ergreifen sollen, wenn sie einen Fehler feststellen.

Mit den beiden folgenden Tools können Sie feststellen, ob ein Fehler in Ihrer DB2-Datenbanklösung aufgetreten ist:

DB2-Fehlermonitorfunktion

Die DB2-Fehlermonitorfunktion sorgt dafür, dass die DB2-Datenbankinstanzen betriebsbereit sind. Wird eine DB2-Datenbankinstanz, der ein DB2-Fehlermonitor zugeordnet ist, unerwartet beendet, startet der DB2-Fehlermonitor die Instanz erneut. Wenn Ihre Datenbanklösung in einem Cluster implementiert ist, sollten Sie die Cluster-Management-Software so konfigurieren, dass fehlgeschlagene Datenbankinstanzen von ihr erneut gestartet werden, statt von dem DB2-Fehlermonitor.

Überwachung durch die Überwachungssignalfunktion in Clusterumgebungen

Die Cluster-Management-Software verwendet Überwachungssignalnachrichten zwischen den Knoten eines Clusters, um den Zustand der Knoten zu überwachen. Der Cluster-Manager stellt fest, dass ein Knoten fehlgeschlagen ist, wenn der Knoten nicht mehr antwortet oder keine Nachrichten mehr sendet.

Überwachung von DB2 High Availability Disaster Recovery-Datenbanken (HADR-Datenbanken)

Die HADR-Funktion hat ihren eigenen Überwachungssignalmonitor. Die Primärdatenbank und die Bereitschaftsdatenbank erwarten voneinander in regelmäßigen Intervallen Überwachungssignalnachrichten.

Überwachen von HADR (High Availability Disaster Recovery)

Sie können den Status Ihrer HADR-Datenbanken mithilfe der folgenden Methoden überwachen.

Dienstprogramm 'db2pd'
Dieses Dienstprogramm ruft Informationen aus den DB2-Speichergruppen ab. Wenn Sie zum Beispiel Informationen zu High Availability Disaster Recovery für die Datenbank MYDB anzeigen möchten, geben Sie den folgenden Befehl ein:

```
db2pd -db mydb -hadr
```

Befehl GET SNAPSHOT FOR DATABASE
Dieser Befehl sammelt Statusinformationen und formatiert die Ausgabe. Die zurückgelieferten Informationen stellen eine Momentaufnahme des Betriebsstatus des Datenbankmanagers zum Zeitpunkt der Befehlsausführung dar. HADR-Informationen werden in der Ausgabe unter der Überschrift *HADR-Status* angezeigt.

API 'db2GetSnapshot'
Diese Anwendungsprogrammierschnittstelle (API) sammelt Überwachungsdaten zum Datenbankmanager und gibt sie in einen dem Benutzer zugeordneten Datenpuffer aus. Die zurückgelieferten Informationen stellen eine Momentaufnahme des Betriebsstatus des Datenbankmanagers zum Zeitpunkt des API-Aufrufs dar.

HADR-Konfigurationsparameter sind nicht dynamisch:

Wenn Sie einen Parameter ändern, während die HADR-Datenbank online ist, werden die Änderungen sichtbar, wenn Sie den Befehl db2 get db cfg für die Datenbank ausführen. Die Änderungen werden jedoch erst wirksam, wenn Sie die Datenbank stoppen und erneut starten. Zum Abrufen der Parameter, die momentan wirksam sind, verwenden Sie den Befehl GET SNAPSHOT, das Tool 'db2pd' oder die Snapshot Monitor-API.

HADR-Datenbankrollen

Die aktuelle Rolle einer Datenbank wird durch den Datenbankkonfigurationsparameter *hadr_db_role* angezeigt. Gültige Werte für diesen Konfigurationsparameter sind PRIMARY, STANDBY oder STANDARD (der letzte Wert gibt an, dass die Datenbank keine HADR-Datenbank ist).

Status der Bereitschaftsdatenbank

Wenn eine Datenbank die Rolle der Bereitschaftsdatenbank besitzt, befindet sie sich gleichzeitig im Status *aktualisierende Recovery anstehend*. Dementsprechend zeigt die Konfiguration der Bereitschaftsdatenbank folgende Werte:

```
rollfwd_pending (Aktualisierende Recovery anstehend)    = DATABASE
restore_pending (Restore anstehend)                     = YES
```

Handhabung einer ungeplanten Betriebsunterbrechung

Wenn Ihre Datenbankmanagement-Software oder die Software für das Cluster-Management feststellt, dass ein Datenbankserver ausgefallen ist, muss Ihre Datenbanklösung auf diesen Ausfall so schnell und reibungslos wie möglich reagieren. Ihre Datenbanklösung muss versuchen, die Benutzeranwendungen vor dem Aus-

fall abzuschirmen, indem Verarbeitungsprozesse umgeleitet werden, falls möglich, und indem die Funktionsübernahme durch eine sekundäre bzw. Bereitschaftsdatenbank stattfindet, falls diese verfügbar ist.

Wenn Ihre Datenbank- oder Cluster-Management-Software feststellt, dass ein Datenbankserver ausgefallen ist, müssen Sie oder Ihre Cluster-Management-Software wie folgt vorgehen:

1. Ein sekundärer Datenbankserver muss identifiziert, online geschaltet und initialisiert werden, damit er die Operationen des fehlgeschlagenen Datenbankservers übernehmen kann.

   Wenn Sie DB2 High Availability Disaster Recover (HADR) verwenden, um den primären und den Bereitschaftsdatenbankserver zu verwalten, sorgt HADR dafür, dass die Bereitschaftsdatenbank immer mit der Primärdatenbank synchronisiert bleibt. Ebenso verwaltet HADR die Funktionsübernahme von der Primärdatenbank durch die Bereitschaftsdatenbank.
2. Verarbeitungsprozesse der Benutzeranwendungen müssen an den sekundären Datenbankserver weitergeleitet werden.

   Die DB2-Clientweiterleitung kann Clientanwendungen automatisch von einem fehlgeschlagenen Datenbankserver an einen sekundären Datenbankserver weiterleiten, welcher zuvor zu diesem Zweck identifiziert und konfiguriert worden ist.
3. Die fehlgeschlagenen Datenbankserver müssen aus dem System entfernt und repariert werden.

   Wenn die Benutzeranwendungen an einen sekundären oder einen Bereitschaftsdatenbankserver weitergeleitet wurden, kann der fehlgeschlagene Datenbankserver etwaige Anforderungen von Clientanwendungen erst dann wieder handhaben, wenn er neu gestartet oder anderweitig repariert worden ist. Wenn z. B. die Ursache des Fehlers in der Primärdatenbank das unerwartete Beenden einer Datenbankinstanz war, startet die DB2-Fehlermonitorfunktion die Instanz automatisch neu.

Automatische Clientweiterleitung - Beispiele

Die DB2 Data Server-Clientweiterleitung kann Clientanwendungen automatisch von einem fehlgeschlagenen Datenbankserver an einen sekundären Datenbankserver weiterleiten, welcher zuvor zu diesem Zweck identifiziert und konfiguriert worden ist. Sie können ohne großen Aufwand Clientanwendungen erstellen, die diese DB2 Data Server-Funktionalität testen und veranschaulichen.

Das folgende Beispiel zeigt eine automatische Clientweiterleitung für eine Clientanwendung (nur in Pseudocode dargestellt):

```
        int checkpoint = 0;

                check_sqlca(unsigned char *str, struct sqlca *sqlca)
        {
           if (sqlca–>sqlcode == -30081)
           {
                       // Bei Kommunikationsverlust Anwendung sofort beenden
                       exit(1);
           }
                    else

                                // Fehler ausgeben
              printf(...);
 if (sqlca–>sqlcode == -30108)
          {
             // Bei erneutem Verbindungsaufbau fehlgeschlagene Transaktion wiederholen
```

```
                        if (checkpoint == 0)
        {
                                goto checkpt0;
        }
           else if (checkpoint == 1)
        {
                              goto checkpt1;
        }
                        else if (checkpoint == 2)
        {
                                goto checkpt2;
        }
        ....
                        exit;
 }
 }
}

        main()
{
              connect to mydb;
              check_sqlca("connect failed", &sqlca);

        checkpt0:
              EXEC SQL set current schema XXX;
              check_sqlca("set current schema XXX failed", &sqlca);

   EXEC SQL create table t1...;
              check_sqlca("create table t1 failed", &sqlca);

              EXEC SQL commit;
              check_sqlca("commit failed", &sqlca);

              if (sqlca.sqlcode == 0)
   {
                    checkpoint = 1;
   }

        checkpt1:
              EXEC SQL set current schema YYY;
              check_sqlca("set current schema YYY failed", &sqlca);

   EXEC SQL create table t2...;
              check_sqlca("create table t2 failed", &sqlca);

              EXEC SQL commit;
              check_sqlca("commit failed", &sqlca);

              if (sqlca.sqlcode == 0)
   {
                    checkpoint = 2;
   }
...
}
```

Auf dem Clientsystem wird die Datenbank mit dem Namen „mydb" katalogisiert, die auf einen Knoten „hornet" verweist, wobei „hornet" ebenfalls im Knotenverzeichnis (Hostname „hornet" mit Portnummer 456) katalogisiert wird.

Beispiel 1 (mit einer Nicht-HADR-Datenbank)

Auf dem Server „hornet" (Hostname ist gleich „hornet" mit einer Portnummer) wird eine Datenbank mit dem Namen „mydb" erstellt. Darüber hinaus wird die Datenbank „mydb" auch auf dem alternativen Server (Hostname „montero" mit

Portnummer 456) erstellt. Sie müssen außerdem den alternativen Server für die Datenbank „mydb" auf dem Server „hornet" wie folgt aktualisieren:

```
db2 update alternate server for database mydb using hostname montero port 456
```

In der obigen Beispielanwendung und bei nicht eingerichteter Funktion zur automatischen Clientweiterleitung wird die Anwendung im Fall eines Kommunikationsfehlers in der Anweisung `create table t1` beendet. Bei eingerichteter Funktion zur automatischen Clientweiterleitung versucht der DB2-Datenbankmanager die Verbindung zum Host „hornet" (mit Port 456) erneut herzustellen. Wenn dieser noch nicht wieder funktioniert, versucht der DB2-Datenbankmanager den alternativen Serverstandort (Hostname „montero" mit Port 456). Unter der Voraussetzung, dass es keinen Kommunikationsfehler in der Verbindung zum alternativen Serverstandort gibt, kann die Anwendung anschließend mit der Ausführung nachfolgender Anweisungen fortfahren (und die fehlgeschlagene Transaktion erneut ausführen).

Beispiel 2 (mit einer HADR-Datenbank)

Auf dem Server „hornet" (Hostname ist gleich „hornet" mit einer Portnummer) wird die Primärdatenbank „mydb" erstellt. Eine Bereitschaftsdatenbank wird außerdem auf Host „montero" mit Port 456 erstellt. Informationen zur Konfiguration von HADR für die Primär- und die Bereitschaftsdatenbank finden Sie in *Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz*. Sie müssen außerdem den alternativen Server für die Datenbank „mydb" wie folgt aktualisieren:

```
db2 update alternate server for database mydb using hostname montero port 456
```

In der obigen Beispielanwendung und bei nicht eingerichteter Funktion zur automatischen Clientweiterleitung wird die Anwendung im Fall eines Kommunikationsfehlers in der Anweisung `create table t1` beendet. Bei eingerichteter Funktion zur automatischen Clientweiterleitung versucht das DB2-Datenbanksystem die Verbindung zum Host „hornet" (mit Port 456) erneut herzustellen. Wenn dieser noch nicht wieder funktioniert, versucht das DB2-Datenbanksystem den alternativen Serverstandort (Hostname „montero" mit Port 456). Unter der Voraussetzung, dass es keinen Kommunikationsfehler in der Verbindung zum alternativen Serverstandort gibt, kann die Anwendung anschließend mit der Ausführung nachfolgender Anweisungen fortfahren (und die fehlgeschlagene Transaktion erneut ausführen).

Ausführen einer HADR-Funktionsübernahmeoperation

Wenn Sie die aktuelle Bereitschaftsdatenbank als neue Primärdatenbank einrichten wollen, da die aktuelle Primärdatenbank nicht verfügbar ist, können Sie eine Funktionsübernahme ausführen.

Warnung:

Diese Prozedur kann zu Datenverlusten führen. Lesen Sie die folgenden Informationen, bevor Sie diese Notfallprozedur ausführen:

- Stellen Sie sicher, dass die Primärdatenbank keine Datenbanktransaktionen mehr verarbeitet. Wenn die Primärdatenbank immer noch aktiv ist, jedoch mit der Bereitschaftsdatenbank nicht mehr kommunizieren kann, könnte eine erzwungene Übernahmeoperation (Befehl TAKEOVER HADR mit Option BY FORCE) zu zwei Primärdatenbanken führen. Wenn zwei Primärdatenbanken vorhanden sind, enthalten diese Datenbanken jeweils unterschiedliche Daten, und eine automatische Synchronisierung der Datenbanken ist nicht mehr möglich.
  - Deaktivieren Sie die Primärdatenbank, oder stoppen Sie ihre Instanz, sofern möglich. (Wenn das primäre System blockiert, ausgefallen oder aus anderen

Gründen nicht verfügbar ist, ist dies eventuell nicht möglich.) Nach einer Übernahmeoperation übernimmt die ausgefallene Datenbank, wenn sie später erneut gestartet wird, nicht automatisch die Rolle der Primärdatenbank.

- Die Wahrscheinlichkeit und der Umfang eines Transaktionsverlusts hängt von Ihrer spezifischen Konfiguration und den Umständen ab:
  - Wenn die Primärdatenbank ausfällt, während sie sich im Status 'Peer' oder 'Getrennter Peer' befindet und der Synchronisationsmodus SYNC (synchron) verwendet wird, gehen in der Bereitschaftsdatenbank keine Transaktionen verloren, die einer Anwendung vor dem Ausfall der Primärdatenbank als festgeschrieben gemeldet wurden.
  - Wenn die Primärdatenbank ausfällt, während sie sich im Status 'Peer' oder 'Getrennter Peer' befindet und der Synchronisationsmodus NEARSYNC (fast synchron) verwendet wird, gehen in der Bereitschaftsdatenbank nur dann von der Primärdatenbank festgeschriebene Transaktionen verloren, wenn die Primärdatenbank und die Bereitschaftsdatenbank gleichzeitig ausfallen.
  - Wenn die Primärdatenbank ausfällt, während sie sich im Status 'Peer' oder 'Getrennter Peer' befindet und der Synchronisationsmodus ASYNC (asynchron) verwendet wird, können in der Bereitschaftsdatenbank von der Primärdatenbank festgeschriebene Transaktionen verloren gehen, wenn die Bereitschaftsdatenbank vor dem Ausführen der Übernahmeoperation nicht alle Protokollsätze empfangen hat. In der Bereitschaftsdatenbank können von der Primärdatenbank festgeschriebene Transaktionen außerdem verloren gehen, wenn die Primärdatenbank und die Bereitschaftsdatenbank gleichzeitig ausfallen.
  - Wenn die Primärdatenbank ausfällt, während sie sich im Status *Fernes Catch-up anstehend* befindet, gehen Transaktionen, die von der Bereitschaftsdatenbank nicht empfangen und verarbeitet wurden, verloren.

    Anmerkung: Die in der Momentaufnahme der Datenbank angezeigte Protokollabstimmungsdiskrepanz stellt die Diskrepanz zu dem Zeitpunkt dar, an dem die Primärdatenbank und die Bereitschaftsdatenbank zuletzt miteinander kommunizierten. Die Primärdatenbank kann nach diesem Zeitpunkt eine große Anzahl Transaktionen verarbeitet haben.
- Stellen Sie sicher, dass jede Anwendung, die eine Verbindung zur neuen Primärdatenbank herstellt (bzw. durch die Clientweiterleitung an die neue Primärdatenbank weitergeleitet wird), darauf vorbereitet ist, die folgenden Fälle aufzufangen:
  - Bei der Funktionsübernahme kommt es zu Datenverlust. Die neue Primärdatenbank enthält nicht alle Transaktionen, die in der alten Primärdatenbank festgeschrieben wurden. Dies ist möglich, selbst wenn der Konfigurationsparameter HADR_SYNCMODE auf SYNC gesetzt ist. Da eine HADR-Bereitschaftsdatenbank Protokolle sequenziell anwendet, können Sie von der Annahme ausgehen, dass, wenn eine Transaktion in einer SQL-Sitzung in der neuen Primärdatenbank festgeschrieben (COMMIT) wird, alle vorherigen Transaktionen in derselben Sitzung ebenfalls in der neuen Primärdatenbank wurden. Die COMMIT-Folge der Transaktionen über mehrere Sitzungen hinweg lässt sich nur durch eine detaillierte Analyse des Protokolldatenstroms bestimmen.
  - Es ist möglich, dass eine Transaktion an die ursprüngliche Primärdatenbank abgesetzt, in der ursprünglichen Primärdatenbank festgeschrieben und in die neue Primärdatenbank (d. h. die ursprüngliche Bereitschaftsdatenbank) repliziert wird, jedoch nicht als festgeschrieben gemeldet wird, weil die ursprüngliche Primärdatenbank abstürzt, bevor sie an den Client melden kann, dass die Transaktion festgeschrieben wurde. Jede Anwendung, die Sie schrei-

ben, sollte solche Fälle verarbeiten können, in denen Transaktionen, die an die ursprüngliche Primärdatenbank abgesetzt wurden, jedoch nicht als in der ursprünglichen Primärdatenbank festgeschrieben gemeldet werden, in der neuen Primärdatenbank (der ursprünglichen Bereitschaftsdatenbank) festgeschrieben werden.
  - Manche Operationen werden nicht repliziert. Dazu gehören Änderungen an der Datenbankkonfiguration und an externen UDF-Objekten.
- Der Befehl TAKEOVER HADR kann nur für die Bereitschaftsdatenbank abgesetzt werden.
- HADR tauscht keine Daten mit dem DB2-Fehlermonitor (db2fm) aus, der dazu verwendet werden kann, eine ausgefallene Datenbank automatisch erneut zu starten. Wenn der Fehlermonitor aktiviert ist, sollten Sie sich darüber im Klaren sein, dass der Fehlermonitor für eine vermutlich ausgefallene Primärdatenbank wahrscheinlich Aktionen ausführen wird.
- Eine Übernahmeoperation kann nur dann ausgeführt werden, wenn sich die Primärdatenbank und die Bereitschaftsdatenbank im Peerstatus bzw. die Bereitschaftsdatenbank im Status *Fernes Catch-up anstehend* befindet. Befindet sich die Bereitschaftsdatenbank in einem anderen Status, wird ein Fehler zurückgegeben.

  Anmerkung: Sie können eine Bereitschaftsdatenbank, die sich im Status *Lokales Catch-up anstehend* befindet, zur normalen Verwendung zur Verfügung stellen, indem Sie sie in eine Standarddatenbank konvertieren. Fahren Sie die Datenbank dazu mit dem Befehl DEACTIVATE DATABASE herunter, und setzen Sie anschließend den Befehl STOP HADR ab. Nachdem HADR gestoppt wurde, müssen Sie eine aktualisierende Recovery für die frühere Bereitschaftsdatenbank durchführen, bevor sie wieder verwendet werden kann. Eine Datenbank kann nicht wieder in ein HADR-Paar aufgenommen werden, nachdem sie von einer Bereitschaftsdatenbank in eine Standarddatenbank konvertiert wurde. Um HADR auf den beiden Servern erneut zu starten, führen Sie die Prozedur zum Initialisieren von HADR aus.
  Wenn Sie ein Peerfenster konfiguriert haben, beenden Sie die Primärdatenbank, bevor das Fenster abgelaufen ist, um im Falle einer damit zusammenhängenden Funktionsübernahme einen möglichen Transaktionsverlust zu vermeiden.

In einem Funktionsübernahmeszenario kann eine Übernahmeoperation über den Befehlszeilenprozessor (CLP), in der Steuerzentrale über das Fenster 'HADR verwalten' oder über die Anwendungsprogrammierschnittstelle (API) db2HADRTakeover ausgeführt werden.

Die folgende Prozedur zeigt, wie Sie über den CLP eine Funktionsübernahme in der Primärdatenbank oder der Bereitschaftsdatenbank einleiten:

1. Inaktivieren Sie die fehlgeschlagene Primärdatenbank vollständig. Wenn in einer Datenbank interne Fehler auftreten, kann sie mit normalen Befehlen zum Herunterfahren möglicherweise nicht vollständig inaktiviert werden. Sie müssen gegebenenfalls Betriebssystembefehle verwenden, um Ressourcen wie Prozesse, gemeinsam genutzten Speicher oder Netzverbindungen zu entfernen.
2. Setzen Sie den Befehl TAKEOVER HADR mit der Option BY FORCE für die Bereitschaftsdatenbank ab. Im folgenden Beispiel wird die Funktionsübernahme für die Datenbank LEAFS ausgeführt:

   ```
   TAKEOVER HADR ON DB LEAFS BY FORCE
   ```

   Die Option BY FORCE ist erforderlich, da davon auszugehen ist, dass die Primärdatenbank offline ist.

Wenn die Primärdatenbank nicht vollständig inaktiviert ist, ist die Bereitschaftsdatenbank weiterhin mit ihr verbunden und sendet eine Anforderung zum Herunterfahren an die Primärdatenbank. Die Bereitschaftsdatenbank übernimmt unabhängig davon, ob sie eine Bestätigung erhält, dass die Primärdatenbank heruntergefahren wurde, die Rolle der Primärdatenbank.

Gehen Sie wie folgt vor, um das Fenster **HADR übernehmen** zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie die Datenbank finden, für die HADR verwaltet werden soll. Klicken Sie die Datenbank mit der rechten Maustaste an, und klicken Sie im Kontextmenü die Option **HADR (High Availability Disaster Recovery) → Verwalten** an. Das Fenster **HADR verwalten** wird geöffnet.
2. Klicken Sie **HADR übernehmen** an. Das Fenster **HADR übernehmen** wird geöffnet.
3. Wählen Sie die Option zum Ausführen einer Funktionsübernahme aus.
4. Wenn beide Datenbanken im HADR-Paar als Bereitschaftsdatenbank gestartet wurden, wählen Sie eine der Datenbanken für die Übernahme als Primärdatenbank aus.
5. Klicken Sie OK an. Das Fenster wird geschlossen. Möglicherweise wird ein Statusanzeigefeld für die Befehlsausführung geöffnet. Nach Abschluss der Befehlsausführung werden Sie benachrichtigt, ob der Befehl erfolgreich ausgeführt wurde oder nicht.
6. Aktualisieren Sie das Fenster **HADR verwalten**, um sicherzustellen, dass die Bereitschaftsdatenbank tatsächlich die Rolle als neue Primärdatenbank übernommen hat.
7. Wenn Sie die Funktion zur automatischen Clientweiterleitung nicht verwenden, leiten Sie die Clientanwendungen zur neuen Primärdatenbank weiter.

Detaillierte Informationen finden Sie in der Onlinehilfefunktion der Steuerzentrale.

Tauschen von Datenbankrollen bei High Availability Disaster Recovery (HADR)

Wenn Sie HADR verwenden, können Sie mit dem Befehl TAKEOVER HADR die Rollen der Primärdatenbank und der Bereitschaftsdatenbank vertauschen.

- Der Befehl TAKEOVER HADR kann nur für die Bereitschaftsdatenbank abgesetzt werden. Wenn beim Absetzen des Befehls keine Verbindung zwischen der Primärdatenbank und der Bereitschaftsdatenbank besteht, schlägt die Übernahmeoperation fehl.
- Der Befehl TAKEOVER HADR kann nur dann für einen Rollenwechsel der Primärdatenbank und der Bereitschaftsdatenbank verwendet werden, wenn sich die beiden Datenbanken im Peerstatus befinden. Wenn die Bereitschaftsdatenbank einen anderen Status hat, wird eine Fehlernachricht zurückgegeben.

Sie können die HADR-Datenbankrollen über den Befehlszeilenprozessor (CLP), in der Steuerzentrale über das Fenster 'HADR verwalten' oder über die Anwendungsprogrammierschnittstelle (API) db2HADRTakeover tauschen.

Wenn Sie über den CLP eine Übernahmeoperation für eine Bereitschaftsdatenbank einleiten wollen, setzen Sie den Befehl TAKEOVER HADR ohne die Option BY FORCE für die Bereitschaftsdatenbank ab.

Im folgenden Beispiel wird die Übernahmeoperation für die Bereitschaftsdatenbank LEAFS ausgeführt:

```
TAKEOVER HADR ON DB LEAFS
```

Direkt im Anschluss an eine Übernahmeoperation ist das Auftreten eines Fehlers wegen eines gefüllten Protokolls geringfügig wahrscheinlicher. Um die Wahrscheinlichkeit eines solchen Fehlers zu begrenzen, wird am Ende jeder TAKEOVER-Operation automatisch ein asynchrones Löschen (Flush) des Pufferpools gestartet. Die Wahrscheinlichkeit, dass ein Fehler wegen eines vollen Protokolls auftritt, sinkt mit dem Fortschreiten der asynchronen Flushoperation für den Pufferpool. Wenn darüber hinaus Ihre Konfiguration einen ausreichend großen Speicherbereich für die aktiven Protokolldateien bereitstellt, ist ein Fehler wegen eines vollen Protokolls noch unwahrscheinlicher. Wenn dennoch ein Fehler durch ein volles Protokoll verursacht wird, wird die aktuelle Transaktion abgebrochen und rückgängig gemacht.

Anmerkung: Durch die Ausführung des Befehls TAKEOVER HADR ohne die Option BY FORCE wird die Verbindung aller Anwendungen, die zurzeit mit der HADR-Primärdatenbank verbunden sind, zwangsweise getrennt. Diese Aktion ist dafür vorgesehen, in Koordination mit der automatischen Clientweiterleitung die Weiterleitung von Clients an die neue HADR-Primärdatenbank nach einem Rollwechsel zu unterstützen. Wenn jedoch die erzwungene Trennung von Anwendungen von der Primärdatenbank in Ihrer Umgebung zu Unterbrechungen führen kann, ist es für Sie vielleicht sinnvoller, eine eigene Prozedur zu implementieren, mit der Sie solche Anwendungen vor dem Rollenwechsel beenden und nach erfolgtem Rollenwechsel mit der neuen HADR-Primärdatenbank als Ziel erneut zu starten.

Gehen Sie wie folgt vor, um das Fenster **HADR übernehmen** zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis Sie die Datenbank finden, für die HADR verwaltet werden soll. Klicken Sie die Datenbank mit der rechten Maustaste an, und klicken Sie im Kontextmenü die Option **HADR (High Availability Disaster Recovery) → Verwalten** an. Das Fenster **HADR verwalten** wird geöffnet.
2. Stellen Sie sicher, dass sich die Datenbanken im Peerstatus befinden.
3. Klicken Sie **HADR übernehmen** an. Das Fenster **HADR übernehmen** wird geöffnet.
4. Wählen Sie die Option zum Wechseln der Rollen aus.
5. Wenn beide Datenbanken im HADR-Paar als Bereitschaftsdatenbank gestartet wurden, wählen Sie eine der Datenbanken für die Übernahme als Primärdatenbank aus.
6. Klicken Sie OK an. Das Fenster wird geschlossen. Möglicherweise wird ein Statusanzeigefeld für die Befehlsausführung geöffnet. Nach Abschluss der Befehlsausführung werden Sie benachrichtigt, ob der Befehl erfolgreich ausgeführt wurde oder nicht.
7. Aktualisieren Sie das Fenster **HADR verwalten**, um sicherzustellen, dass die Datenbanken wirklich ihre Rollen gewechselt haben.
8. Wenn Sie die Funktion zur automatischen Clientweiterleitung nicht verwenden, leiten Sie die Clientanwendungen zur neuen Primärdatenbank weiter.

   Weitere Informationen enthält die Kontexthilfefunktion der Steuerzentrale.

Reintegrieren einer Datenbank nach einer Übernahmeoperation

Wenn in einer HADR-Umgebung die Primärdatenbank fehlschlägt und Sie daher eine Übernahmeoperation ausgeführt haben, können Sie die fehlgeschlagene Datenbank wieder in den Onlinestatus versetzen und entweder als Bereitschaftsdatenbank verwenden oder sie in ihren früheren Status als Primärdatenbank zurückversetzen.

Gehen Sie wie folgt vor, um die fehlgeschlagene Primärdatenbank als neue Bereitschaftsdatenbank in das HADR-Paar zu integrieren:

1. Reparieren Sie das System, auf dem sich die ursprüngliche Primärdatenbank befand. Dies kann eine Reparatur von ausgefallener Hardware oder einen Neustart des fehlgeschlagenen Betriebssystems umfassen.
2. Starten Sie die fehlgeschlagene Primärdatenbank als Bereitschaftsdatenbank. Im folgenden Beispiel wird die Datenbank LEAFS als Bereitschaftsdatenbank gestartet:

   ```
   START HADR ON DB LEAFS AS STANDBY
   ```

   Anmerkung: Die Reintegration schlägt fehl, wenn die Protokolldatenströme der beiden Datenbankkopien nicht kompatibel sind. Insbesondere setzt HADR voraus, dass die ursprüngliche Primärdatenbank keine protokollierten Operationen anwendet hat, die in der ursprünglichen Bereitschaftsdatenbank noch nicht nachvollzogen waren, bevor diese die Rolle als neue Primärdatenbank übernahm. War dies der Fall, können Sie die ursprüngliche Primärdatenbank als Bereitschaftsdatenbank neu starten, indem Sie ein Backup-Image der neuen Primärdatenbank wiederherstellen oder eine geteilte Spiegeldatenbank initialisieren.

   Eine Erfolgsmeldung dieses Befehls bedeutet nicht, dass die Integration erfolgreich war. Sie bedeutet lediglich, dass die Datenbank gestartet wurde. Die Reintegration befindet sich noch in der Ausführung. Wenn die Reintegration nachfolgend fehlschlägt, fährt sich die Datenbank selbst herunter. Sie sollten die Status der Bereitschaftsdatenbank mithilfe des Befehls GET SNAPSHOT FOR DATABASE oder des Tools 'db2pd' überwachen, um sicherzustellen, dass die Bereitschaftsdatenbank online verbleibt und mit den normalen Statusübergängen fortfährt. Falls erforderlich, können Sie den Status der Datenbank anhand der Verwaltungsprotokolldatei 'db2diag.log' feststellen.

Nachdem die ursprüngliche Primärdatenbank dem HADR-Paar als Bereitschaftsdatenbank hinzugefügt wurde, können Sie eine Zurücksetzungsoperation ausführen, um die Rollen der Datenbanken wieder zu tauschen, sodass die ursprüngliche Primärdatenbank erneut als aktuelle Primärdatenbank genutzt werden kann. Setzen Sie den folgenden Befehl für die Bereitschaftsdatenbank ab, um diese Zurücksetzungsoperation auszuführen:

```
TAKEOVER HADR ON DB LEAFS
```

Anmerkung:

1. Wenn sich die HADR-Datenbanken nicht im Peerstatus befinden oder keine Verbindung zwischen ihnen besteht, schlägt dieser Befehl fehl.
2. Geöffnete Primärdatenbanksitzungen werden zwangsweise geschlossen, und unvollständige Transaktionen werden rückgängig gemacht.
3. Wenn Sie die Rollen der Primär- und der Bereitschaftsdatenbank tauschen, kann die Option BY FORCE des Befehls TAKEOVER HADR nicht angegeben werden.

Teil 2. Datenrecovery

Eine Recovery ist die Wiederherstellung einer Datenbank oder eines Tabellenbereichs nach einem Problem, zum Beispiel einem Datenträger- oder Speicherfehler, einem Stromausfall oder einem Anwendungsfehler. Wenn Sie Ihre Datenbank oder einzelne Tabellenbereiche mit Backup gesichert haben, können Sie diese erneut erstellen, wenn sie beschädigt oder aus einem anderen Grund unbrauchbar werden.

Drei Arten der Recovery stehen zur Verfügung:

- Eine Recovery nach einem Systemabsturz verhindert, dass eine Datenbank nach unerwartetem Abbruch von Transaktionen bzw. UOWs (Units of Work, Arbeitseinheiten) in einem inkonsistenten oder nicht verwendbaren Zustand verbleibt.
- Eine Versionsrecovery ist die Wiederherstellung einer früheren Version der Datenbank mithilfe eines Images der Datenbank, das im Rahmen einer Backup-Operation erstellt wurde.
- Eine aktualisierende Recovery dient zur erneuten Anwendung von Änderungen, die durch Transaktionen nach einem Backup festgeschrieben wurden.

Der DB2-Datenbankmanager startet eine Recovery nach einem Systemabsturz automatisch, um z. B. eine Datenbank nach einem Stromausfall wiederherzustellen.
Eine beschädigte Datenbank können Sie durch eine Versionsrecovery oder eine aktualisierende Recovery wiederherstellen.

Kapitel 6. Entwickeln einer Backup- und Recoverystrategie

Eine Datenbank kann aufgrund von Hardware- oder Softwarefehlern (oder beidem) unbrauchbar werden. Irgendwann treten möglicherweise Speicherprobleme, Stromausfälle oder Anwendungsfehler auf, und jedes Fehlerszenario erfordert unterschiedliche Recoveryaktionen. Schützen Sie Ihre Daten vor Verlust, indem Sie eine gut erprobte Recoverystrategie bereithalten. Bei der Entwicklung Ihrer Recoverystrategie sollten Sie unter anderem folgende Fragen berücksichtigen:

- Soll die Datenbank wiederherstellbar sein?
- Wie viel Zeit steht für die Recovery der Datenbank zur Verfügung?
- In welchen Abständen werden Backup-Operationen durchgeführt?
- Wie viel Speicher kann für Backupkopien und Archivprotokolldateien zugeordnet werden?
- Sind Backups auf Tabellenbereichsebene ausreichend, oder muss die gesamte Datenbank gesichert werden?
- Sollte ein Bereitschaftssystem manuell oder über HADR (High Availability Disaster Recovery) konfiguriert werden?

Eine Strategie zur Recovery der Datenbank sollte sicherstellen, dass alle Informationen verfügbar sind, wenn sie für eine Recovery der Datenbank benötigt werden. Sie sollte einen Zeitplan für regelmäßige Backups enthalten. Im Fall von Umgebungen mit partitionierten Datenbanken sollten auch nach einer Skalierung des Systems, d. h. nach dem Hinzufügen oder Löschen von Datenbankpartitionsservern oder -knoten, Backups durchgeführt werden. Ihre Gesamtstrategie sollte auch Prozeduren zum Wiederherstellen von Befehlsscripts, Anwendungen, benutzerdefinierten Funktionen, Code gespeicherter Prozeduren in Betriebssystembibliotheken sowie von Ladekopien enthalten.

In den folgenden Abschnitten werden verschiedene Recoverymethoden behandelt. Sie können bestimmen, welche Recoverymethode für Ihre Geschäftsumgebung am besten geeignet ist.

Das Konzept eines *Datenbank-Backups* ist mit jedem anderen Daten-Backup vergleichbar: Sie erstellen eine Kopie der Daten und speichern die Kopie anschließend auf einem anderen Datenträger für den Fall eines Ausfalls oder einer Beschädigung der Originaldaten. Den einfachsten Fall eines Backups bildet das Verfahren, bei dem zunächst die Datenbank gestoppt wird, um sicherzustellen, dass keine weiteren Transaktionen auftreten, und anschließend ein Backup-Image der Daten erstellt wird. Sie können die Datenbank dann erneut erstellen, falls sie beschädigt wird.

Das erneute Erstellen der Datenbank wird als *Recovery* bezeichnet. Eine *Versionsrecovery* ist die Wiederherstellung einer früheren Version der Datenbank mithilfe eines Images der Datenbank, das im Rahmen einer Backup-Operation erstellt wurde. Eine *aktualisierende Recovery* ist die erneute Anwendung von in den Datenbankprotokolldateien aufgezeichneten Transaktionen nach dem Restore eines Backup-Images einer Datenbank oder eines Tabellenbereichs.

Eine *Recovery nach einem Systemabsturz* ist die automatische Recovery der Datenbank, wenn ein Fehler auftritt, bevor alle Änderungen einer oder mehrerer UOWs (Transaktionen) beendet und festgeschrieben wurden. Dies geschieht durch den

Rollback der unvollständigen Transaktionen und Beenden der festgeschriebenen Aktionen, die sich noch im Hauptspeicher befanden, als der Systemabsturz auftrat.

Die Protokolldateien für die Recovery und die Datei des Recoveryprotokolls werden automatisch erstellt, wenn eine Datenbank erstellt wird (Abb. 10). Diese Protokolldateien sind wichtig, falls Sie verlorene oder beschädigte Daten wiederherstellen müssen.

Jede Datenbank enthält *Recoveryprotokolle*, die für die Recovery nach Anwendungs- oder Systemfehlern verwendet werden. In Kombination mit den Datenbankbackups dienen sie zur Recovery der Konsistenz der Datenbank bis exakt zu dem Zeitpunkt, an dem der Fehler auftrat.

Die *Datei des Recoveryprotokolls* enthält eine Zusammenfassung der Backupinformationen, die zur Bestimmung der Recoveryoptionen verwendet werden können, wenn die Datenbank insgesamt oder teilweise bis zu einem bestimmten Zeitpunkt wiederhergestellt werden muss. Sie dient zur Protokollierung recoveryrelevanter Ereignisse wie z. B. Backup- und Restoreoperationen.Diese Datei befindet sich im Datenbankverzeichnis.

Die *Protokolldatei der Tabellenbereichsänderungen*, die sich ebenfalls im Datenbankverzeichnis befindet, enthält Informationen, anhand deren bestimmt werden kann, welche Protokolldateien für die Recovery eines bestimmten Tabellenbereichs erforderlich sind.

Die Datei des Recoveryprotokolls und die Protokolldatei der Tabellenbereichsänderungen können nicht direkt geändert werden. Mit dem Befehl PRUNE HISTORY können Sie jedoch Einträge aus den Dateien löschen. Sie haben auch die Möglichkeit, mit dem Datenbankkonfigurationsparameter *rec_his_retentn* die Anzahl der Tage anzugeben, die diese Protokolldateien beibehalten werden sollen.

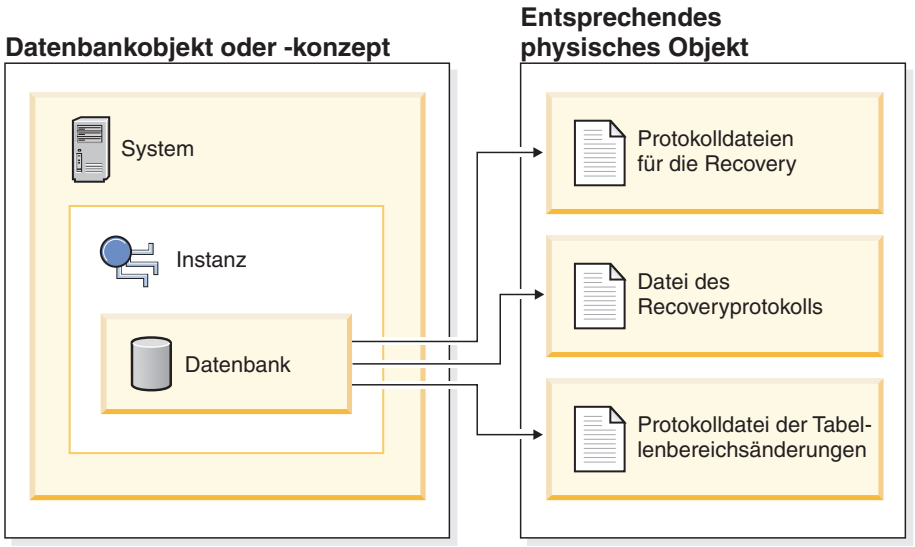


Abbildung 10. Datenbankrecoverydateien

Daten, die sich problemlos erneut erstellen lassen, können in einer nicht wiederherstellbaren Datenbank gespeichert werden. Dazu gehören Daten von einer externen Quelle, die für Anwendungen mit Lesezugriff verwendet werden, und Tabellen, die nicht oft aktualisiert werden und für die der geringe Protokollierungsaufwand nicht die zusätzliche Komplexität der Verwaltung von Protokolldateien sowie die aktualisierende Recovery nach einer Restoreoperation rechtfertigt. Wenn die Datenbankkonfigurationsparameter *logarchmeth1* und *logarchmeth2* beide auf „OFF" gesetzt sind, ist die Datenbank *nicht wiederherstellbar*. Dies bedeutet, dass lediglich die für die Recovery nach einem Systemabsturz erforderlichen Protokolle beibehalten werden. Diese Protokolle werden als *aktive Protokolldateien* bezeichnet und enthalten aktuelle Transaktionsdaten. Die Versionsrecovery mithilfe von *Offline-Backups* ist das primäre Mittel zur Recovery einer nicht wiederherstellbaren Datenbank. (Ein Offline-Backup bedeutet, dass während der Backup-Operation keine andere Anwendung die Datenbank verwenden kann.) Eine solche Datenbank kann nur offline wiederhergestellt werden. Sie wird in dem Status wiederhergestellt, den sie hatte, als das Backup-Image erstellt wurde. Eine aktualisierende Recovery wird nicht unterstützt.

Daten, die sich *nicht* einfach erneut erstellen lassen, sollten in einer wiederherstellbaren Datenbank gespeichert werden. Hierzu gehören Daten, deren Quelle nach dem Laden der Daten zerstört wird, Daten, die manuell in Tabellen eingegeben werden, sowie Daten, die nach dem Laden in die Datenbank von Anwendungsprogrammen oder Benutzern geändert werden. Bei *wiederherstellbaren Datenbanken* sind die Datenbankkonfigurationsparameter *logarchmeth1* oder *logarchmeth2* auf einen anderen Wert als „OFF" gesetzt. Aktive Protokolldateien sind weiterhin für die Datenbankrecovery nach einem Systemabsturz verfügbar. Es stehen jedoch auch die *Archivprotokolldateien* zur Verfügung, die festgeschriebene Transaktionsdaten enthalten. Eine solche Datenbank kann nur offline wiederhergestellt werden. Sie wird in dem Status wiederhergestellt, den sie hatte, als das Backup-Image erstellt wurde. Mithilfe der aktualisierenden Recovery (Rollforward Recovery) können Sie jedoch die Datenbank *aktualisierend wiederherstellen* (also über den Zeitpunkt hinaus, an dem das Backup-Image erstellt wurde), indem Sie die aktiven und archivierten Protokolle entweder bis zu einem bestimmten Zeitpunkt oder bis zum Ende der aktiven Protokolldateien anwenden.

Backup-Operationen für wiederherstellbare Datenbanken können entweder offline oder *online* durchgeführt werden (online heißt, dass andere Anwendungen während der Backup-Operation eine Verbindung zur Datenbank herstellen können). Operationen zum Online-Restore eines Tabellenbereichs und zur aktualisierenden Recovery werden nur für wiederherstellbare Datenbanken unterstützt. Wenn die Datenbank nicht wiederherstellbar ist, müssen Operationen zum Restore und zur aktualisierenden Recovery der Datenbank offline ausgeführt werden. Während eines Online-Backups stellt die aktualisierende Recovery sicher, dass *alle* Tabellenänderungen erfasst und erneut angewendet werden, wenn dieses Backup wiederhergestellt wird.

Bei einer wiederherstellbaren Datenbank können Sie anstelle der gesamten Datenbank auch einzelne Tabellenbereiche sichern, wiederherstellen und aktualisierend wiederherstellen. Wenn Sie einen Tabellenbereich online sichern, kann er weiterhin verwendet werden, und gleichzeitig durchgeführte Änderungen werden in den Protokollen aufgezeichnet. Wenn Sie einen Tabellenbereich online wiederherstellen oder online aktualisierend wiederherstellen, kann der Tabellenbereich selbst nicht mehr verwendet werden, bis die Operation beendet ist, Benutzer können jedoch weiterhin auf Tabellen in anderen Tabellenbereichen zugreifen.

Automatisierte Backup-Operationen

Da die Bestimmung, ob und wann Verwaltungsaktivitäten wie zum Beispiel Backup-Operationen auszuführen sind, zeitaufwendig sein kann, haben Sie die Möglichkeit, diese Aufgabe durch den Assistenten 'Automatische Verwaltung konfigurieren' erledigen zu lassen. Bei der automatischen Verwaltung definieren Sie Ihre Verwaltungszielsetzungen, einschließlich der möglichen Ausführungszeiten für die automatische Verwaltung. Anschließend ermittelt DB2 anhand dieser Informationen, ob Verwaltungsaktivitäten erforderlich sind und führt im nächsten verfügbaren Verwaltungsfenster (einem benutzerdefinierten Zeitraum zum Ausführen von automatischen Verwaltungsaktivitäten) nur die erforderlichen Verwaltungsaktivitäten aus.

Anmerkung: Nach der Konfiguration der automatischen Verwaltung haben Sie weiterhin die Möglichkeit, manuelle Backup-Operationen ausführen. DB2 führt automatische Backup-Operationen nur dann aus, wenn sie erforderlich sind.

Häufigkeit von Backups

In Ihrem Recoveryplan sollten regelmäßige Backup-Operationen vorgesehen sein, da das Sichern einer Datenbank Zeit und Systemressourcen in Anspruch nimmt. Ihr Plan kann eine Kombination aus Datenbankgesamtbackups und inkrementellen Backups enthalten.

Sie sollten in regelmäßigen Abständen Gesamtbackups der Datenbanken erstellen, selbst wenn Sie die Protokolle archivieren (wodurch eine aktualisierende Recovery ermöglicht wird). Für die Recovery einer Datenbank können Sie entweder ein Image eines Datenbankgesamtbackups verwenden, das Backup-Images aller Tabellenbereiche enthält, oder Sie können die Datenbank mithilfe ausgewählter Tabellenbereichsimages erneut erstellen (Rebuildfunktion). Images von Tabellenbereichsbackups sind auch sinnvoll für eine Recovery nach dem Ausfall einer einzelnen Platte oder nach einem Anwendungsfehler. In Umgebungen mit partitionierten Datenbanken müssen Sie jeweils nur die Tabellenbereiche wiederherstellen, die sich in den Datenbankpartitionen befinden, die ausgefallen sind. Es müssen nicht alle Tabellenbereiche oder Datenbankpartitionen wiederhergestellt werden.

Obwohl Datenbankgesamtbackups zur Datenbankwiederherstellung jetzt, da Sie eine Datenbank mithilfe von Tabellenbereichsimages wiederherstellen können, nicht länger erforderlich sind, ist es dennoch eine gute Praxis, gelegentlich Gesamtbackups der Datenbank zu erstellen.

Außerdem sollten Sie in Betracht ziehen, die Backup-Images und Protokolldateien nicht zu überschreiben, sondern mindestens zwei Gesamtbackup-Images der Datenbank mit zugehörigen Protokollen als weitere Vorsichtsmaßnahme zu sichern.

Wenn die erforderliche Zeit für die Anwendung der archivierten Protokolldateien bei der (aktualisierenden) Recovery einer sehr aktiven Datenbank wichtig ist, sollten Sie den Aufwand häufigerer Datenbank-Backups in Erwägung ziehen. Dadurch verringert sich die Anzahl der archivierten Protokolldateien, die Sie bei einer aktualisierenden Recovery anwenden müssen.

Sie können eine Backup-Operation starten, während die Datenbank *online* oder *offline* ist. Wenn die Datenbank online ist, können andere Anwendungen oder Prozesse Verbindungen zur Datenbank herstellen sowie Daten lesen und ändern, wäh-

rend die Backup-Operation ausgeführt wird. Wird die Backup-Operation offline ausgeführt, können andere Anwendungen *keine* Verbindung zur Datenbank herstellen.

Um den Zeitraum, in dem die Datenbank nicht verfügbar ist, möglichst kurz zu halten, sollten Sie Online-Backup-Operationen in Betracht ziehen. Online-Backup-Operationen werden nur unterstützt, wenn die aktualisierende Recovery aktiviert ist. Wenn die aktualisierende Recovery aktiviert ist und Sie über einen kompletten Satz von Recoveryprotokollen verfügen, können Sie die Datenbank im Bedarfsfall wiederherstellen. Sie können ein Online-Backup-Image nur dann für die Recovery verwenden, wenn Sie über die Protokolle verfügen, die sich über den Zeitraum erstrecken, über den die Backup-Operation ausgeführt wurde.

Offline-Backup-Operationen sind schneller als Online-Backup-Operationen, da zwischen den Datendateien keine Konkurrenzsituation auftritt.

Mit dem Backup-Dienstprogramm können Sie ausgewählte Tabellenbereiche sichern. Wenn Sie DMS-Tabellenbereiche verwenden, können Sie verschiedene Datentypen in speziellen Tabellenbereichen speichern, um die für Backup-Operationen benötigte Zeit zu verringern. Sie können die Tabellendaten in einem Tabellenbereich, die Langfeld- und LOB-Daten in einem anderen Tabellenbereich und die Indizes in einem dritten Tabellenbereich unterbringen. Wenn Sie diese Maßnahmen ergreifen, wird ein auftretender Plattenfehler vermutlich nur einen der Tabellenbereiche betreffen. Zum Restore oder zur aktualisierenden Recovery dieser Tabellenbereiche ist weniger Zeit erforderlich, als für den Restore eines einzelnen Tabellenbereichs, der alle Daten enthält.

Sie können auch Zeit sparen, indem Sie zu unterschiedlichen Zeitpunkten Backups unterschiedlicher Tabellenbereiche ausführen, solange die an ihnen vorgenommenen Änderungen nicht dieselben sind. Wenn beispielsweise Langfeld- oder LOB-Daten nicht so häufig geändert werden wie die übrigen Daten, müssen Sie diese Tabellenbereiche auch nicht so häufig sichern. Wenn die Langfeld- und LOB-Daten nicht zur Recovery benötigt werden, ist ein Backup dieser Tabellenbereiche möglicherweise gar nicht erforderlich. Wenn die LOB-Daten von einer getrennten Quelle reproduziert werden können, sollten Sie beim Erstellen oder Ändern einer Tabelle zum Hinzufügen von LOB-Spalten die Option NOT LOGGED verwenden.

Anmerkung: Der folgende Hinweis gilt für den Fall, dass Sie Ihre Langfelddaten, LOB-Daten und Indizes in separaten Tabellenbereichen speichern, aber sie nicht gemeinsam sichern: Wenn Sie einen Tabellenbereich sichern, der nicht alle Tabellendaten enthält, können Sie keine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt für diesen Tabellenbereich durchführen. Alle Tabellenbereiche, die irgendeine Art von Daten für eine Tabelle enthalten, müssen gleichzeitig bis zum selben Zeitpunkt aktualisierend wiederhergestellt werden.

Wenn Sie eine Tabelle reorganisieren, sollten Sie die betroffenen Tabellenbereiche nach Beendigung der Operation sichern. Wenn dann der Fall eintritt, dass die Tabellenbereiche wiederhergestellt werden müssen, muss die Datenreorganisation nicht mehr aktualisierend wiederhergestellt werden.

Die Zeit, die zur Recovery einer Datenbank benötigt wird, setzt sich aus zwei Perioden zusammen: der Zeit, die zum Restore der Backup-Kopie benötigt wird, sowie, wenn die Datenbank für die aktualisierende Recovery aktiviert ist, der Zeit, die zur Anwendung der Protokolle während der aktualisierenden Recovery benötigt wird. Bei der Formulierung eines Recoveryplans sollten Sie diesen Recoveryaufwand und seine Auswirkung auf den Geschäftsbetrieb berücksichtigen. Durch Testen des

Gesamtrecoveryplans können Sie ermitteln, ob die Zeit, die zur Recovery der Datenbank benötigt wird, in Anbetracht Ihrer Geschäftserfordernisse angemessen ist. Nach jedem Test stellen Sie möglicherweise fest, dass es vorteilhaft ist, die Häufigkeit der Backups zu erhöhen. Wenn Ihre Strategie eine aktualisierende Recovery vorsieht, wird dieses Vorgehen die Anzahl der zwischen Backups archivierten Protokolle reduzieren und auf diese Weise die für eine aktualisierende Recovery der Datenbank benötigte Zeit nach einem Restore von der Backupkopie verringern.

Aspekte des Speicherbedarfs für Recovery

Bei der Entscheidung für die zu verwendende Recoverymethode sollten Sie auch den erforderlichen Speicherplatz in Betracht ziehen.

Bei der Versionsrecovery wird Speicherplatz für die Backupkopie der Datenbank und für die wiederhergestellte Datenbank benötigt. Bei einer aktualisierenden Recovery wird Speicherplatz für die Backupkopie der Datenbank bzw. der Tabellenbereiche, für die wiederhergestellte Datenbank sowie für die archivierten Datenbankprotokolle benötigt.

Enthält eine Tabelle Langfeld- oder LOB-Spalten, ist es u. U. ratsam, die betreffenden Daten in einen separaten Tabellenbereich zu stellen. Dies hat Auswirkungen auf Ihre Überlegungen zum Speicherbedarf sowie auf Ihren Recoveryplan. Wenn Sie einen separaten Tabellenbereich für Langfeld- und LOB-Daten verwenden und Ihnen der Zeitaufwand für das Sichern der Langfeld- und LOB-Daten bekannt ist, können Sie einen Recoveryplan verwenden, bei dem dieser Tabellenbereich nur gelegentlich gesichert wird. Beim Erstellen oder Ändern einer Tabelle, die LOB-Spalten umfasst, können Sie zudem angeben, dass Änderungen an diesen Spalten nicht protokolliert werden sollen. Dadurch verringert sich die Größe der Protokolldateien und somit der erforderliche Protokollarchivierungsspeicher.

Um zu verhindern, dass ein Datenträgerfehler die Datenbank beschädigt und Ihnen die Wiederherstellung unmöglich macht, sollten Sie die Backupkopie der Datenbank, die Datenbankprotokolle sowie die Datenbank selbst auf unterschiedlichen Einheiten speichern. Aus diesem Grund wird ausdrücklich empfohlen, nach dem Erstellen der Datenbank die Datenbankprotokolle mithilfe des Konfigurationsparameters *newlogpath* auf eine separate Einheit umzuleiten.

Die Datenbankprotokolle können viel Speicherplatz in Anspruch nehmen. Wenn Sie beabsichtigen, eine aktualisierende Recovery durchzuführen, müssen Sie entscheiden, wie die archivierten Protokolle verwaltet werden. Ihnen stehen folgende Möglichkeiten zur Auswahl:

- Geben Sie unter Verwendung des Konfigurationsparameters LOGARCHMETH1 oder LOGARCHMETH2 eine Protokollarchivierungsmethode an.
- Kopieren Sie die Protokolldateien manuell auf eine Speichereinheit bzw. in ein Verzeichnis, die/das nicht mit dem Verzeichnispfad der Datenbankprotokolle übereinstimmt, nachdem die Protokolldateien nicht mehr zur Gruppe der aktiven Protokolldateien gehören.
- Kopieren Sie diese Protokolle mithilfe eines Benutzerexitprogramms auf eine andere Speichereinheit in Ihrer Umgebung.

Backupkomprimierung

Neben den Speichereinsparungen, die Sie durch die Zeilenkomprimierung in Ihrer aktiven Datenbank erzielen können, haben Sie auch die Möglichkeit, die Backupkomprimierung zu verwenden, um die Größe Ihrer Datenbankbackups zu reduzieren.

Während die Zeilenkomprimierung auf Tabellenbasis erfolgt, werden bei der Komprimierung von Backups *alle* Daten im Backup-Image komprimiert, einschließlich der Katalogtabellen, Indexobjekte, LOB-Objekte, Zusatzdatenbankdateien und Datenbankmetadaten.

Sie können die Backupkomprimierung auf Tabellen anwenden, die bereits die Zeilenkomprimierung verwenden. Dabei ist jedoch zu beachten, dass die Backupkomprimierung zusätzliche CPU-Ressourcen und zusätzliche Zeit erfordert. Möglicherweise reicht die Tabellenkomprimierung alleine aus, um eine Reduzierung Ihrer Backupspeicheranforderungen zu erzielen. Wenn Sie bereits die Zeilenkomprimierung verwenden, ist die Backupkomprimierung möglicherweise nur dann sinnvoll, wenn die Speicheroptimierung so wichtig ist, dass die zusätzliche Zeit zur Ausführung des Backups in Kauf genommen werden kann.

Tipp: Eventuell sollte die Backupkomprimierung nur in Tabellenbereichen erfolgen, die keine komprimierten Daten enthalten, wenn folgende Bedingungen zutreffen:

- Daten und Indexobjekte sind von LOB- und Langfelddaten getrennt und
- Sie verwenden die Zeilen- und Indexkomprimierung bei der Mehrheit Ihrer Datentabellen bzw. Indizes.

Um Ihre Backups zu komprimieren, verwenden Sie die Option COMPRESS im Befehl BACKUP DATABASE.

Zusammenhalten zusammengehöriger Daten

Im Lauf des Entwurfsprozesses für eine Datenbank entwickeln sich Erkenntnisse über die Beziehungen, die zwischen Tabellen bestehen. Solche Beziehungen können auf folgenden Ebenen bemerkbar machen:

- Auf der Anwendungsebene, wenn Transaktionen mehrere Tabellen aktualisieren
- Auf der Datenbankebene, wenn die referenzielle Integrität zwischen Tabellen zu gewährleisten ist oder sich Trigger einer Tabelle auf eine andere Tabelle auswirken

Diese Beziehungen sollten bei der Entwicklung eines Recoveryplans berücksichtigt werden. Wahrscheinlich ist es sinnvoll, voneinander abhängige Datengruppen gemeinsam zu sichern. Solche zusammengehörigen Datengruppen können entweder auf Tabellenbereichsebene oder auf Datenbankebene eingerichtet werden. Wenn die zusammengehörigen Datengruppen zusammen gesichert werden, können sie bis zu einem Punkt wiederhergestellt werden, an dem alle Daten konsistent sind. Dies ist besonders wichtig, wenn Sie in der Lage sein wollen, für Tabellenbereiche eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt auszuführen.

Backup- und Restoreoperationen zwischen unterschiedlichen Betriebssystemen und Hardwareplattformen

DB2-Datenbanksysteme unterstützen einige Backup- und Restoreoperationen zwischen unterschiedlichen Betriebssystemen und Hardwareplattformen.

Die unterstützten Plattformen für DB2-Backup- und Restoreoperationen können in eine von drei Familien gruppiert werden:

- Big Endian: Linux und UNIX
- Little Endian Linux und UNIX
- Windows

Ein Datenbank-Backup von einer Plattformfamilie kann auf einem anderen beliebigen System in derselben Plattformfamilie wiederhergestellt werden. Bei Windows-Betriebssystemen können Sie eine Datenbank, die in DB2 Universal Database (UDB) Version 8 erstellt wurde, in einem DB2-Datenbanksystem Version 9 wiederherstellen. Bei Linux- und UNIX-Betriebssystemen können Sie, sofern das Endian-Format (Big Endian oder Little Endian) der Backup- und Restore-Plattformen identisch ist, Backups, die unter DB2 UDB Version 8 erstellt wurden, unter DB2 Version 9 wiederherstellen.

In der folgenden Tabelle sehen Sie die einzelnen Linux- und UNIX-Plattformen, die DB2 unterstützt; außerdem wird angezeigt, ob die Plattformen das Endian-Format Big Endian oder Little Endian aufweisen:

Tabelle 10. Endian-Formate der unterstützten Linux- und UNIX-Betriebssysteme (von DB2 unterstützt)

Plattform	Endian-Format
AIX	Big Endian
HP unter IA64	Big Endian
Solaris x64	Little Endian
Solaris SPARC	Big Endian
Linux unter zSeries	Big Endian
Linux unter pSeries	Big Endian
Linux unter IA-64	Little Endian
Linux auf AMD64 und Intel EM64T	Little Endian
Linux (32-Bit) unter x86	Little Endian

Auf dem Zielsystem muss dieselbe (oder eine höhere) Version des DB2-Datenbankprodukts installiert sein wie auf dem Quellensystem. Sie können für ein Backup keinen Restore durchführen, das auf einer Version des Datenbankprodukts für ein System erstellt wurde, das eine frühere Version des Datenbankprodukts ausführt. Beispiel: Sie können ein DB2 UDB V8-Backup auf einem DB2 V9-Datenbanksystem wiederherstellen, nicht jedoch ein DB2 V9-Backup auf einem DB2 UDB V8-Datenbanksystem.

Anmerkung: Sie können eine Datenbank aus einem Backup-Image, das auf einer 32-Bit-Ebene erstellt wurde, in einer 64–Bit-Ebene wiederherstellen, nicht jedoch umgekehrt. Für das Backup und die Wiederherstellung Ihrer Datenbanken sollten Sie die DB2-Dienstprogramme Backup und Restore verwenden. Das Verschieben einer Dateigruppe von einer Maschine auf eine andere wird nicht empfohlen, da dies zu einer Beeinträchtigung der Datenbankintegrität führen kann.

In Fällen, bei denen bestimmte Backup- und Restorekombinationen nicht zulässig sind, können Sie Tabellen zwischen DB2-Datenbanken verschieben und dazu andere Methoden verwenden:

- db2move (Befehl)
- Dienstprogramm EXPORT, gefolgt vom Dienstprogramm IMPORT oder LOAD

Kapitel 7. Datei des Recoveryprotokolls

Mit jeder Datenbank wird eine Datei des Recoveryprotokolls (Recovery History File) erstellt. Diese Datei wird automatisch aktualisiert, wenn eine der folgenden Aktionen stattfindet:

- Sichern einer Datenbank oder von Tabellenbereichen
- Wiederherstellen einer Datenbank oder von Tabellenbereichen
- Aktualisierendes Wiederherstellen einer Datenbank oder von Tabellenbereichen
- Automatisches erneutes Erstellen einer Datenbank und Wiederherstellen mehrerer Images
- Erstellen eines Tabellenbereichs
- Ändern eines Tabellenbereichs
- Durchführen des Quiesce für einen Tabellenbereich
- Umbenennen eines Tabellenbereichs
- Löschen eines Tabellenbereichs
- Laden einer Tabelle
- Löschen einer Tabelle (wenn die Recovery gelöschter Tabellen aktiviert ist)
- Reorganisieren einer Tabelle
- Aufrufen der bedarfsgesteuerten Archivierung von Protokolldateien
- Schreiben in eine neue Protokolldatei (wenn die Protokollierung für Recovery verwendet wird)
- Archivieren einer Protokolldatei (wenn die Protokollierung für Recovery verwendet wird)
- Wiederherstellen einer Datenbank

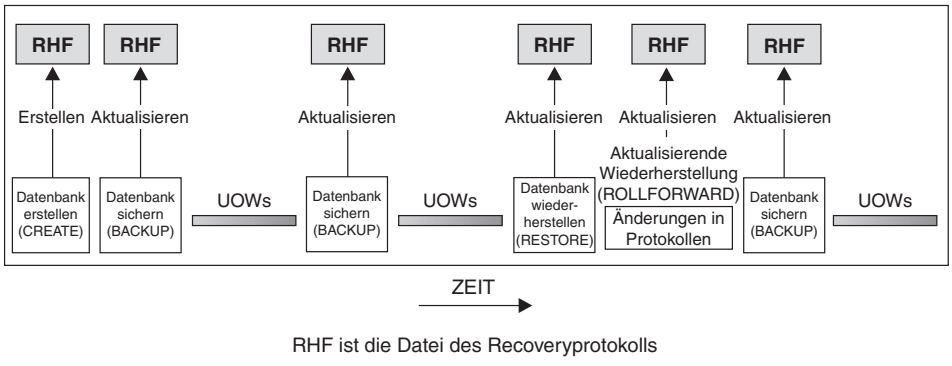


Abbildung 11. Erstellen und Aktualisieren der Datei des Recoveryprotokolls

Sie können die zusammengefassten Backup-Informationen in dieser Datei zur Recovery der gesamten Datenbank oder eines Teils der Datenbank bis zu einem bestimmten Zeitpunkt verwenden. Die Datei enthält folgende Informationen:

- Ein Identifikationsfeld zur eindeutigen Kennzeichnung der einzelnen Einträge
- Den Teil der Datenbank, der kopiert wurde, und Angaben zur Kopiermethode
- Den Zeitpunkt, an dem die Kopie erstellt wurde
- Die Speicherposition der Kopie (mit Informationen zur Einheit und logischen Möglichkeit für den Zugriff auf die Kopie)

- Den Zeitpunkt des letzten Restores
- Den Zeitpunkt, zu dem ein Tabellenbereich umbenannt wurde, sowie den vorherigen und den aktuellen Name des Tabellenbereichs
- Den Status der Backup-Operation: aktiv, inaktiv, abgelaufen oder gelöscht
- Die letzte Protokollfolgenummer, die vom Datenbank-Backup gespeichert oder von einer aktualisierenden Recovery verarbeitet wurde

Verwenden Sie den Befehl LIST HISTORY, wenn Sie die Einträge in der Datei des Recoveryprotokolls anzeigen wollen.

Jede Backup-Operation (sowohl für einen Tabellenbereich als auch für die gesamte Datenbank bzw. für ein inkrementelles Backup) schließt eine Kopie der Datei des Recoveryprotokolls mit ein. Die Datei des Recoveryprotokolls ist der Datenbank zugeordnet. Beim Löschen einer Datenbank wird auch die Datei des Recoveryprotokolls gelöscht. Beim Wiederherstellen einer Datenbank an einer neuen Position wird auch die Datei des Recoveryprotokolls wiederhergestellt. Beim Wiederherstellen wird die vorhandene Datei des Recoveryprotokolls nur dann überschrieben, wenn die auf dem Datenträger vorhandene Datei keine Einträge enthält. Ist dies der Fall, wird das Datenbankprotokoll aus dem Backup-Image wiederhergestellt.

Wenn die aktuelle Datenbank unbrauchbar oder nicht verfügbar ist und die zugehörige Datei des Recoveryprotokolls beschädigt oder gelöscht wurde, steht eine Option des Befehls RESTORE zur Verfügung, mit der lediglich die Datei des Recoveryprotokolls wiederhergestellt werden kann. Im Anschluss daran kann anhand der Informationen in der Datei des Recoveryprotokolls festgestellt werden, welches Backup zum Restore der Datenbank verwendet werden soll.

Die Größe dieser Datei wird mit dem Konfigurationsparameter *rec_his_retentn* gesteuert, der den Zeitraum (in Tagen) angibt, über den die Einträge in der Datei zu behalten sind. Auch wenn dieser Parameter auf den Wert null (0) gesetzt wurde, werden die Informationen zum aktuellen vollständigen Datenbank-Backup und der zugehörigen Restoregruppe beibehalten. (Diese Kopie kann nur über den Befehl PRUNE mit der Option FORCE gelöscht werden.) Der Aufbewahrungszeitraum beträgt standardmäßig 366 Tage. Mit der Angabe -1 kann für diesen Zeitraum eine unbegrenzte Anzahl von Tagen definiert werden. In diesem Fall muss der Befehl PRUNE für die Datei explizit verwendet werden.

Eintragsstatus der Datei des Recoveryprotokolls

Der Datenbankmanager erstellt in der Datei des Recoveryprotokolls Einträge für Ereignisse wie z. B. eine Backup- oder Restoreoperation, das Erstellen eines Tabellenbereichs usw. Jeder Eintrag in der Datei des Recoveryprotokolls hat einen zugehörigen Status: aktiv (active), inaktiv (inactive), abgelaufen (expired), gelöscht (deleted) oder nicht löschen (do_not_delete).

Der Datenbankmanager verwendet den Status eines Eintrags in der Datei des Recoveryprotokolls, um festzustellen, ob die zu diesem Eintrag gehörenden physischen Dateien notwendig sind, um die Datenbank wiederherzustellen. Als Teil der automatischen Bereinigung aktualisiert der Datenbankmanager den Status der Einträge in der Datei des Recoveryprotokolls.

Aktive Datenbankbackups

Ein aktives Datenbankbackup ist ein Backup, das wiederhergestellt und mit aktuellen Protokollen aktualisierend wiederhergestellt werden kann, um wieder den aktuellen Status der Datenbank zu erhalten.

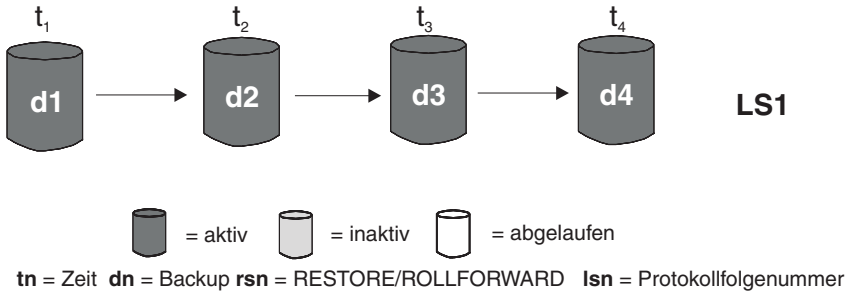


Abbildung 12. Aktive Datenbankbackups. Der Wert von *num_db_backups* wurde auf vier gesetzt.

Inaktive Datenbankbackups

Ein inaktives Datenbankbackup ist ein Backup, bei dessen Restore die Datenbank in einen früheren Zustand zurückversetzt wird.

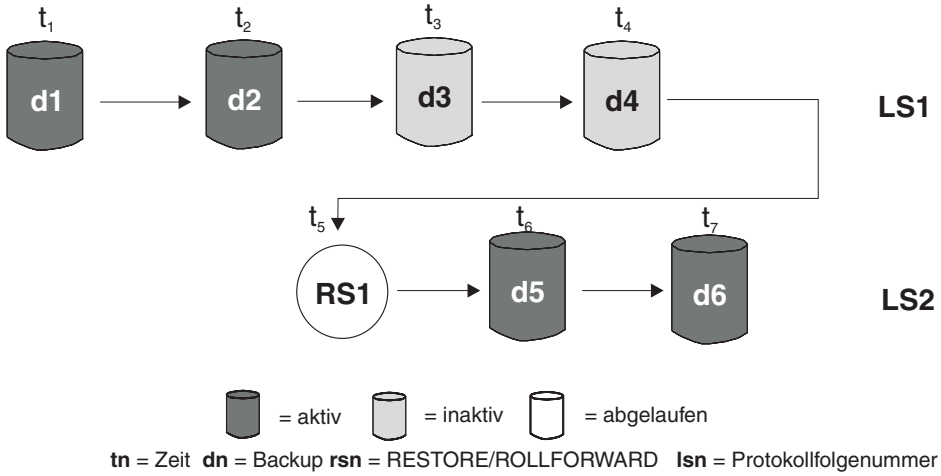


Abbildung 13. Inaktive Datenbankbackups

Abgelaufene Datenbankbackups

Ein abgelaufenes Datenbankbackup-Image wird nicht mehr benötigt, da neuere Backup-Images verfügbar sind.

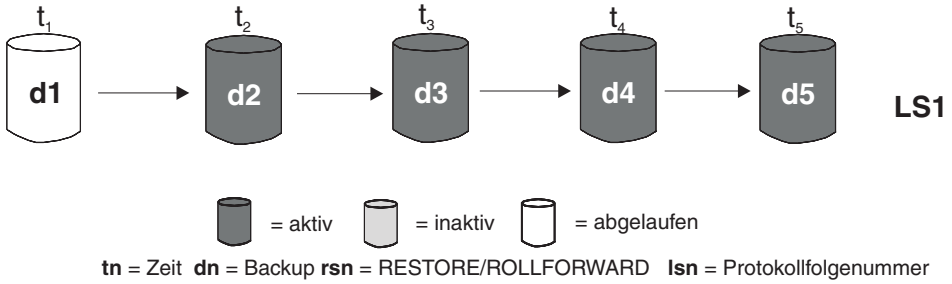


Abbildung 14. Abgelaufene Datenbankbackups

Einträge mit der Markierung do_not_delete

Sie können Einträge in der Datei des Recoveryprotokolls mit dem Befehl PRUNE HISTORY oder der API db2Prune entfernen (bereinigen). Der Datenbankmanager entfernt außerdem die Einträge in der Datei des Recoveryprotokolls als Teil der automatischen Bereinigung.

Es gibt lediglich drei Möglichkeiten, Einträge mit der Markierung do_not_delete zu entfernen:

- Rufen Sie den Befehl PRUNE HISTORY mit der Option WITH FORCE auf
- Rufen Sie die Prozedur ADMIN_CMD mit PRUNE HISTORY und der Option WITH FORCE auf
- Rufen Sie die API db2Prune mit der Option DB2_PRUNE_OPTION_FORCE auf

Die Einträge mit der Markierung do_not_delete werden ausschließlich dann aus der Datei des Recoveryprotokolls entfernt, wenn Sie eine dieser drei Aktionen ausführen.

Der Status der Einträge in der Datei des Recoveryprotokolls wird nicht vom Datenbankmanager auf do_not_delete gesetzt. Sie können den Status eines Eintrags in der Datei des Recoveryprotokolls mit dem Befehl UPDATE HISTORY auf do_not_delete setzen.

Weitere Beispiele für den Status verschiedener Einträge in der Datei des Recoveryprotokolls:

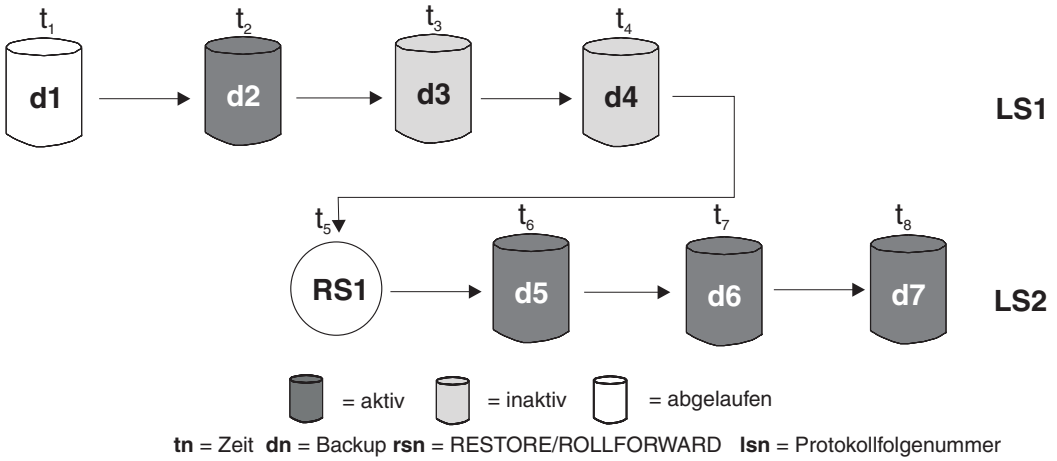


Abbildung 15. Gemischte aktive, inaktive und abgelaufene Datenbankbackups

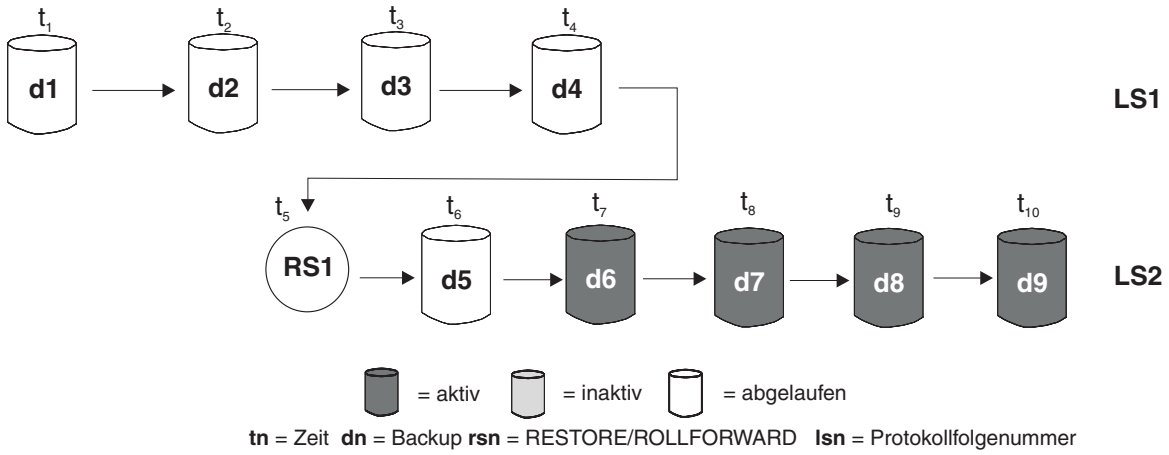


Abbildung 16. Abgelaufene Protokollfolge

Einträge in der Datei des Recoveryprotokolls mithilfe der Verwaltungssicht DB_HISTORY anzeigen

Mithilfe der Verwaltungssicht DB_HISTORY() können Sie auf den Inhalt der Datenbankprotokolldatei zugreifen. Diese Methode stellt eine Alternative zum CLP-Befehl LIST HISTORY bzw. den C-Protokoll-APIs dar.

Für die Verwendung dieser Funktion ist eine Datenbankverbindung erforderlich.

Lösch- und Aktualisierungsoperationen in der Datenbankprotokolldatei können nur über die Befehle PRUNE bzw. UPDATE HISTORY ausgeführt werden.

Diese Verwaltungssicht ist für Datenbanken, die mit DB2 Universal Database Version 8.2 oder einer früheren Version erstellt wurden, nicht verfügbar.

Gehen Sie wie folgt vor, um auf die Protokolldatei zuzugreifen:

1. Stellen Sie sicher, dass eine Verbindung zu einer Datenbank besteht.
2. Verwenden Sie die Verwaltungssicht DB_HISTORY() in einer SQL-Anweisung SELECT, um auf die Datenbankprotokolldatei für die Datenbank, zu der die Verbindung besteht, oder für die Datenbankpartition, die durch die Umgebungsvariable DB2NODE angegeben wird, zuzugreifen. Geben Sie beispielsweise zum Anzeigen des Inhalts der Protokolldatei Folgendes ein:

```
SELECT * FROM TABLE(DB_HISTORY()) AS LIST_HISTORY
```

Um die Syntax der Verwaltungssicht zu verdecken, können Sie eine Sicht wie folgt erstellen:

```
CREATE VIEW LIST_HISTORY AS
        SELECT * FROM TABLE(DB_HISTORY()) AS LIST_HISTORY
```

Nach der Erstellung dieser Sicht können Sie Abfragen dafür ausführen. Beispiel:

```
SELECT * FROM LIST_HISTORY
```

oder

```
SELECT dbpartitionnum FROM LIST_HISTORY
```

oder

```
SELECT dbpartitionnum, start_time, seqnum, tabname, sqlstate
       FROM LIST_HISTORY
```

In Tabelle 11 sind die Spalten und die Spaltendatentypen aufgeführt, die von der Sicht LIST_HISTORY zurückgegeben werden.

Tabelle 11. Inhalt der Protokolltabelle

Spaltenname	**Datentyp**
dbpartitionnum smallint	smallint
EID	bigint
start_time	varchar(14)
seqnum	smallint
end_time	varchar(14)
firstlog	varchar(254)
lastlog	varchar(254)
backup_id	varchar(24)
tabschema	varchar(128)
tabname	varchar(128)
comment	varchar(254)
cmd_text	clob(2M)
num_tbsps	integer
tbspnames	clob(5M)
operation	char(1)
operationtype	char(1)
objecttype	varchar(255)
location	char(1)
devicetype	char(1)
entry_status	varchar(8)
sqlcaid	varchar(8)
sqlcabc	integer
sqlcode	integer
sqlerrml	smallint
sqlerrmc	varchar(70)
sqlerrp	varchar(8)
sqlerrd1	integer
sqlerrd2	integer
sqlerrd3	integer
sqlerrd4	integer
sqlerrd5	integer
sqlwarn	varchar(11)
sqlstate	varchar(5)

Bereinigen der Datei des Recoveryprotokolls

Der Datenbankmanager erstellt in der Datei des Recoveryprotokolls Einträge für Ereignisse wie z. B. eine Backup- oder Restoreoperation, das Erstellen eines Tabellenbereichs usw. Wenn ein Eintrag in der Datei des Recoveryprotokolls nicht mehr relevant ist, weil die zugehörigen Recoveryobjekte nicht mehr für eine Recovery der Datenbank benötigt werden, empfiehlt es sich, diese Einträge aus der Datei des Recoveryprotokolls zu entfernen (sie zu *bereinigen*).

Verwenden Sie die folgenden Methoden, um die Einträge in der Datei des Recoveryprotokolls zu bereinigen:

- Rufen Sie den Befehl PRUNE HISTORY auf
- Rufen Sie die API db2Prune auf
- Rufen Sie die Prozedur ADMIN_CMD mit dem Parameter PRUNE_HISTORY auf

Wenn Sie eine dieser Methoden zum Bereinigen der Datei des Recoveryprotokolls verwenden, entfernt (bereinigt) der Datenbankmanager Einträge aus der Datei des Recoveryprotokolls, die älter als die von Ihnen angegebene Zeitmarke sind.

Wenn ein Eintrag in der Datei des Recoveryprotokolls den von Ihnen festgelegten Kriterien für das Bereinigen entspricht, dieser Eintrag jedoch noch für eine Recovery der Datenbank benötigt wird, entfernt der Datenbankmanager diesen Eintrag nur dann, wenn Sie den Parameter WITH FORCE oder die Markierung DB2PRUNE_OPTION_FORCE verwenden.

Wenn Sie den Parameter AND DELETE oder die Markierung DB2PRUNE_OPTION_DELETE verwenden, werden auch Protokolldateien, die den bereinigten Einträgen zugeordnet sind, gelöscht.

Wenn Sie den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON setzen und Sie den Parameter AND DELETE oder die Markierung DB2PRUNE_OPTION_DELETE verwenden, werden auch Protokolldateien, Backup-Images und Ladekopie-Images, die den bereinigten Einträgen zugeordnet sind, gelöscht.

Bereinigen der Datei des Recoveryprotokolls automatisieren

Sie können den Datenbankmanager so konfigurieren, dass der Status der Einträge in der Datei des Recoveryprotokolls automatisch bereinigt und aktualisiert wird.

Sie können den Status der Einträge in der Datei des Recoveryprotokolls manuell aktualisieren, indem Sie den Befehl UPDATE HISTORY, die API db2HistoryUpdate oder die Prozedur ADMIN_CMD zusammen mit dem Parameter "UPDATE_HISTORY" verwenden. Sie können außerdem den Befehl PRUNE HISTORY, die API db2Prune oder die Prozedur ADMIN_CMD mit dem Parameter "PRUNE_HISTORY" verwenden, um Einträge manuell in der Datei des Recoveryprotokolls zu entfernen (zu "bereinigen"). Es wird jedoch empfohlen, den Datenbankmanager für die Verwaltung der Datei des Recoveryprotokolls zu konfigurieren, statt die Datei des Recoveryprotokolls manuell zu aktualisieren und zu bereinigen.

Der Datenbankmanager aktualisiert und bereinigt die Einträge in der Datei des Recoveryprotokolls automatisch zu folgenden Zeiten:

- Nachdem eine vollständige (nicht inkrementelle) Datenbanksicherungsoperation oder eine vollständige (nicht inkrementelle) Tabellenbereichsoperation erfolgreich abgeschlossen wurde
- Nachdem eine Datenbankwiederherstellungsoperation, bei der eine aktualisierende Recovery nicht erforderlich ist, erfolgreich abgeschlossen wurde
- Nachdem eine aktualisierende Recoveryoperation für die Datenbank erfolgreich abgeschlossen wurde

Während des automatisierten Bereinigens führt der Datenbankmanager zwei Operationen aus:

1. Aktualisierung des Status der Einträge in der Datei des Recoveryprotokolls
2. Bereinigung abgelaufener Einträge in der Datei des Recoveryprotokolls

Der Datenbankmanager aktualisiert die Einträge der Datei des Recoveryprotokolls auf folgende Art und Weise:

- Alle aktiven Datenbank-Backup-Images, die nicht mehr benötigt werden, werden als abgelaufen markiert.
- Alle Datenbankbackups, die als inaktiv markiert sind und vor dem Zeitpunkt des abgelaufenen Datenbankbackups gespeichert wurden, werden ebenfalls als abgelaufen markiert. Alle zugeordneten inaktiven Backup-Images von Tabellenbereichen und Lade-Backup-Kopien werden ebenfalls als abgelaufen markiert.
- Wenn ein aktives Datenbank-Backup-Image wiederhergestellt wird, es sich aber nicht um das aktuelle in der Protokolldatei aufgezeichnete Datenbank-Backup handelt, werden alle nachfolgenden Datenbank-Backup-Images, die zur selben Protokollfolge gehören, als inaktiv markiert.
- Wenn ein inaktives Datenbank-Backup-Image wiederhergestellt wird, werden alle inaktiven Datenbankbackups, die zur aktuellen Protokollfolge gehören, wieder als aktiv markiert. Alle aktiven Datenbank-Backup-Images, die nicht länger zur aktuellen Protokollfolge gehören, werden als inaktiv markiert.
- Alle Datenbank- oder Tabellenbereichs-Backup-Images, die nicht der laufenden Protokollfolge (auch laufende Protokollkette genannt) entsprechen, werden als inaktiv markiert.

  Die laufende Protokollfolge wird durch das Datenbank-Backup-Image festgelegt, das wiederhergestellt wurde, und durch die verarbeiteten Protokolldateien.

Nach dem Restore eines Datenbank-Backup-Images werden alle zu einem späteren Zeitpunkt erstellten Datenbank-Backup-Images als inaktiv markiert, da mit dem wiederhergestellten Image eine neue Protokollkette beginnt. (Dies trifft zu, wenn das Backup-Image ohne aktualisierende Recovery wiederhergestellt wurde. Wenn eine aktualisierende Recovery durchgeführt wurde, werden alle nach der Unterbrechung der Protokollkette erstellten Datenbankbackups als inaktiv markiert. Es ist vorstellbar, das ein älteres Datenbank-Backup-Image wiederhergestellt werden muss, da das Dienstprogramm zur aktualisierenden Recovery die Protokollfolge verwendet hat, die ein aktuelles beschädigtes Backup-Image enthält.)

- Das Backup-Image eines Tabellenbereichs wird inaktiv, wenn nach seinem Restore der aktuelle Status der Datenbank nicht durch Anwenden der aktuellen Protokollfolge erreicht werden kann.
- Alle Einträge mit dem Status do_not_delete werden nicht bereinigt, und ihre zugehörigen Protokolldateien, Backup-Images und Ladekopie-Images werden ebenfalls nicht gelöscht.
- Wenn eine Datenbank migriert wird, werden in der Protokolldatei alle Backup-Einträge für die Datenbank im Onlinestatus und alle Backup-Einträge für den Tabellenbereich im Online- oder Offlinestatus als abgelaufen markiert, damit diese Einträge nicht von einer automatisierten REBUILD-Operation als Images für die erneute Erstellung ausgewählt werden können. Außerdem werden auch Ladekopie-Images und Protokollarchiveinträge als abgelaufen markiert, da diese Eintragstypen nicht für Recoveryzwecke verwendbar sind.

Die folgenden Datenbankkonfigurationsparameter steuern, welche Einträge vom Datenbankmanager bereinigt werden:

num_db_backups
Gibt die Anzahl der Datenbankbackups an, die für eine Datenbank beibehalten werden müssen

rec_his_retentn
Gibt die Anzahl an Tagen an, für die Langzeitinformationen für Backups aufbewahrt werden müssen

auto_del_rec_obj
Gibt an, ob der Datenbankmanager Protokolldateien, Backup-Images und Ladekopie-Images löschen sollte, die zu den Einträgen in der Datei des Recoveryprotokolls gehören, welche bereinigt werden

Richten Sie die folgenden Konfigurationsparameter ein, um den Datenbankmanager für die automatisierte Verwaltung der Datei des Recoveryprotokolls zu konfigurieren:

- num_db_backups
- rec_his_retentn
- auto_del_rec_obj

Wenn der Parameter auto_del_rec_obj auf ON gesetzt wird und mehr erfolgreiche Datenbankbackup-Einträge als im Konfigurationsparameter num_db_backups festgelegt vorhanden sind, bereinigt der Datenbankmanager automatisch alle Einträge in der Datei des Recoveryprotokolls, die älter als die im Parameter rec_his_retentn festgelegte Anzahl von Tagen sind.

Einträge in der Datei des Recoveryprotokolls vor der Bereinigung schützen

Sie können verhindern, dass Schlüsseleinträge in der Datei des Recoveryprotokolls bereinigt (gelöscht) und zugehörige Recoveryobjekte gelöscht werden, indem Sie den Status der Einträge in der Datei des Recoveryprotokolls auf do_not_delete (Nicht löschen) setzen.

Sie können Einträge in der Datei des Recoveryprotokolls mit dem Befehl PRUNE HISTORY, der Prozedur ADMIN_CMD mit PRUNE_HISTORY oder der API db2Prune entfernen (bereinigen). Wenn Sie den Parameter AND DELETE zusammen mit PRUNE HISTORY oder die Markierung DB2PRUNE_OPTION_DELETE mit db2Prune verwenden und der Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON gesetzt wurde, werden die zugehörigen Recoveryobjekte ebenfalls physisch gelöscht.

Der Datenbankmanager entfernt außerdem die Einträge in der Datei des Recoveryprotokolls als Teil der automatischen Bereinigung. Wenn der Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON gesetzt wurde, löscht der Datenbankmanager die Recoveryobjekte, die etwaigen bereinigten Einträgen zugeordnet sind.

Gehen Sie wie folgt vor, um Schlüsseleinträge in der Datei des Recoveryprotokolls sowie zugehörige Recoveryobjekte zu schützen:

Verwenden Sie den Befehl UPDATE HISTORY, die API db2HistoryUpdate oder die Prozedur ADMIN_CMD zusammen mit dem Parameter "UPDATE_HISTORY", um den Status von Schlüsseleinträgen in der Datei des Recoveryprotokolls auf do_not_delete zu setzen.
Es gibt drei Möglichkeiten, Einträge mit der Markierung do_not_delete zu bereinigen:

- Rufen Sie den Befehl PRUNE HISTORY mit der Option WITH FORCE auf
- Rufen Sie die Prozedur ADMIN_CMD mit PRUNE HISTORY und der Option WITH FORCE auf
- Rufen Sie die API db2Prune mit der Option DB2_PRUNE_OPTION_FORCE auf

Die Einträge mit der Markierung do_not_delete werden nur dann aus der Datei des Recoveryprotokolls entfernt, wenn Sie eine dieser Prozeduren ausführen.
Einschränkungen:

- Sie können nur den Status von Backup-Images, Ladekopie-Images und Protokolldateien auf do_not_delete setzen.
- Der Status eines Backup-Eintrags wird nicht an Ladekopie-Images, nicht inkrementelle Backups oder Protokolldateien weitergegeben, die zu der Backup-Operation gehören. Wenn Sie einen bestimmten Datenbankbackup-Eintrag und seine zugeordneten Protokolldateieinträge sichern möchten, müssen Sie jeweils den Status für den Datenbankbackup-Eintrag und den Eintrag für alle zugehörigen Protokolldateien festlegen.

Kapitel 8. Verwalten von Recoveryobjekten

Wenn Sie regelmäßige Backups Ihrer Datenbank durchführen, können sich sehr große Datenbank-Backup-Images ansammeln, sowie zahlreiche Datenbankprotokolle und Images von Ladekopien. Der Datenbankmanager von IBM Data Server kann die Verwaltung dieser Recoveryobjekte vereinfachen.

Das Speichern von Recoveryobjekten kann sehr viel Speicherplatz beanspruchen. Sobald neue Backup-Operationen ausgeführt werden, können Sie die älteren Recoveryobjekte löschen, da sie nicht mehr für ein Restore der Datenbank notwendig sind. Das Entfernen der älteren Recoveryobjekte kann jedoch zeitaufwendig sein. Außerdem kann es passieren, dass Sie beim Löschen älterer Recoveryobjekte versehentlich Recoveryobjekte löschen, die noch benötigt werden.

Es gibt zwei Möglichkeiten, mit dem Datenbankmanager Recoveryobjekte zu löschen, die nicht mehr für ein Restore der Datenbank erforderlich sind:

- Sie können den Befehl PRUNE HISTORY mit dem Parameter AND DELETE oder die API db2Prune mit der Markierung DB2PRUNE_OPTION_DELETE aufrufen.
- Sie können den Datenbankmanager so konfigurieren, dass nicht benötigte Recoveryobjekte automatisch gelöscht werden.

Protokollfolgenummern

Eine Protokollfolgenummer (LSN) gibt die Speicherposition in der Protokolldatei eines bestimmten Protokolldateisatzes an.

LSNs werden von zahlreichen Komponenten des DB2-Produkts zur Gewährleistung der Datenbankkonsistenz und -integrität verwendet. Sie spielen unter anderem bei Commit- und Rollback-Operationen, bei der Recovery nach Systemabsturz bzw. der aktualisierenden Recovery sowie bei der Synchronisation von Datenbankoperationen in Umgebungen mit partitionierten Datenbanken eine wichtige Rolle.

Die Rate, mit der die LSNs in den Protokolldateien zunehmen, hängt direkt mit der Datenbankaktivität zusammen. Das bedeutet, dass sich die LSNs beim Ausführen von Transaktionen und beim Schreiben von Einträgen in Protokolldateien kontinuierlich erhöhen. Bei einer größeren Aktivität in der Datenbank nehmen die LSNs schneller zu.

Obergrenzen für Protokollfolgenummern

In DB2 Version 9.5 und früheren Versionen haben Protokollfolgenummern (LSNs) ein 6-Byte-Format. Ab Fixpack 3 umfassen LSNs einen Bereich von 0x0000 0000 0000 zum Zeitpunkt der Datenbankerstellung bis 0xFFFF 0000 0000 (etwa 256 Terabyte). Bis zu Fixpack 3 lag die Obergrenze bei 0xFFFF FFFF FFFF. LSNs nehmen über die Lebensdauer der Datenbank zu, wenn Datensätze zu den Protokolldateien hinzugefügt werden.

Anmerkung: In Umgebungen mit partitionierten Datenbanken erfolgt die LSN-Zunahme in den jeweiligen Protokolldateien der einzelnen Datenbankpartitionen unabhängig voneinander. Daher müssen Sie bei einer Prüfung der LSNs in einer partitionierten Umgebung die LSNs jeder einzelnen Datenbankpartition untersuchen.

Es ist selten (und meistens äußerst unwahrscheinlich), dass die Datenbankaktivität so groß ist, dass sich die Protokolldateien dem LSN-Grenzwert von 256 Terabyte nähern. Wenn dies jedoch der Fall ist, schreibt der Datenbankmanager eine Nachricht (ADM1849C) in das Protokoll, um Sie darauf hinzuweisen, dass die LSN nahe am Höchstwert liegt. (Die Erkennung dieses Zustands und die Nachricht wurden in Version 9.5 Fixpack 3 hinzugefügt). Diese Nachricht wird zum Protokoll hinzugefügt, wenn eine LSN den Wert 0xF000 0000 0000 erreicht. Danach wird diese Nachricht bei jeder LSN-Zunahme um 0x0080 0000 0000 Byte erneut in die Protokolldatei geschrieben.

In Version 9.5 Fixpack 3 und höher schreibt der Datenbankmanager beim Erreichen des LSN-Werts 0xFFFF 0000 0000 eine andere Nachricht (ADM1850C) in das Protokoll. Darüber hinaus wird die Datenbank schreibgeschützt. Alle nachfolgenden Versuche, in die Datenbank zu schreiben, scheitern (SQL0946C). Die Aktivierung eines Schreibschutzes für die Datenbank ist eine Sicherheitseinrichtung. Damit sollen Inkonsistenz- oder Integritätsprobleme vermieden werden, die auftreten könnten, wenn die LSN den unterstützten Höchstwert überschreitet. Wenn die Nachricht ADM1849C oder ADM1850C in das Protokoll geschrieben wird, müssen Sie die im Abschnitt „Beheben eines Zustands eines erreichten LSN-Grenzwerts” auf Seite 241 beschriebenen Schritte ausführen.

Wichtig: Wenn es die Aktivität Ihrer Datenbanken wahrscheinlich macht, dass der LSN-Höchstwert erreicht wird, ist es wichtig, dass Sie Fixpack 3 (oder höher) installieren. Andernfalls wird die LSN-Zunahme nicht automatisch erkannt. Wenn die LSN in einem solchen Fall den Höchstwert (0xFFFF FFFF FFFF vor Fixpack 3) überschreitet, wird die Datenbank beschädigt. In diesem Fall können Sie eine Recovery nur durchführen, indem Sie die Datenbank aus einem Backup wiederherstellen und dabei eine aktualisierende Recovery zu einem Zeitpunkt weit vor dem Erreichen des LSN-Grenzwerts wählen. Wenn die beschließen, Fixpack 3 oder höher nicht zu installieren, ist die vorausschauende Überwachung der LSN-Zunahme von entscheidender Bedeutung. Weitere Informationen finden Sie im Abschnitt „Überwachung der Zunahme der Protokollfolgenummern” auf Seite 237.

Beispiel 1: LSN nähert sich dem Maximalwert

Sobald die LSN den ersten Schwellenwert 0xF000 0000 0000 erreicht, schreibt der Datenbankmanager die Nachricht ADM1849C in die Datei db2diag.log:

```
2010-04-07-10.02.41.189027-240 E139861A577 LEVEL: Error
PID : 9224292 TID : 3567 PROC : db2sysc 0
INSTANCE: db2inst1                NODE : 000
EDUID : 3567 EDUNAME: db2loggw (SAMPLE) 0
FUNCTION: DB2 UDB, data protection services, sqlpgWriteToDisk, probe:1033
MESSAGE : ADM1849C Die aktuelle Protokollfolgenummer (LSN - Log Sequence Number)
ist "F0800067B000", und nähert sich dem Maximalwert 0xFFFF FFFF FFFF.
Wenn die Datenbank den maximalen LSN-Wert erreicht, können Sie die
Datenbank nicht mehr verwenden.
```

Das folgende Beispiel zeigt, wie die Nachricht in einem Protokoll mit Benachrichtigungen für die Systemverwaltung angezeigt werden könnte, beispielsweise in db2inst1.nfy:

```
2010-04-07-10.02.41.189322 Instance:db2inst1 Node:000
PID:9224292(db2loggw (SAMPLE) 0) TID:3567 Appid:none
data protection services sqlpgWriteToDisk Probe:1033

ADM1849C Die aktuelle Protokollfolgenummer (LSN - Log Sequence Number)
ist "F0800067B000", und nähert sich dem Maximalwert 0xFFFF FFFF FFFF.
Wenn die Datenbank den maximalen LSN-Wert erreicht, können Sie die
Datenbank nicht mehr verwenden.
```

Beispiel 2: LSN überschreitet den zulässigen Höchstwert

Sobald die LSN den zulässigen Höchstwert 0xFFFF 0000 0000 erreicht, wird die Datenbank schreibgeschützt und der Datenbankmanager schreibt die Nachricht ADM1850C in die Datei db2diag.log:

```
2010-04-07-14.47.58.507137-240 E16916A663 LEVEL: Error
PID : 8384514 TID : 2829 PROC : db2sysc 0
INSTANCE: db2inst1 NODE : 000 DB : SAMPLE
APPHDL : 0-7 APPID: *LOCAL.db2inst1.100407184716
AUTHID : DB2INST1
EDUID : 2829 EDUNAME: db2agent (SAMPLE) 0
FUNCTION: DB2 UDB, data protection services, sqlpWriteLR, probe:1715
MESSAGE : ADM1850C Die Datenbank hat den LSN-Grenzwert (Log Sequence Number -
Protokollfolgenummer) "FFFF00000000" erreicht. Es können keine weiteren
Protokolle mehr geschrieben werden. Daher kann die Datenbank Transaktionen,
die das Schreiben von Protokolldatensätzen erfordern, nicht mehr verarbeiten.
```

Das folgende Beispiel zeigt, wie die Nachricht in einem Protokoll mit Benachrichtigungen für die Systemverwaltung angezeigt werden könnte, beispielsweise in db2inst1.nfy:

```
2010-04-07-14.47.58.516208 Instance:db2inst1 Node:000
PID:8384514(db2agent (SAMPLE) 0) TID:2829 Appid:*LOCAL.db2inst1.100407184716
data protection services sqlpWriteLR Probe:1715 Database:SAMPLE

ADM1850C Die Datenbank hat den LSN-Grenzwert (Log Sequence Number -
Protokollfolgenummer) "FFFF00000000" erreicht. Es können keine
weiteren Protokolle mehr geschrieben werden. Daher kann die Datenbank
Transaktionen, die das Schreiben von Protokolldatensätzen erfordern,
nicht mehr verarbeiten.
```

Upgrade auf DB2 Version 9.7 für nahezu unbegrenzte LSNs

Tipp: In DB2 Version 9.7 wird die Maximalgröße einer Protokollfolgenummer nicht als 6-Byte-, sondern als 8-Byte-Zahl dargestellt. Dank dieser höheren Obergrenze können die LSNs bis zu einem Wert von 0xFFFF FFFE FFFF FFEF anwachsen, was fast 16 Exbibyte entspricht (dabei ist ein Exbibyte = 2^{60} = 1024^6 Byte). In der Praxis bedeutet dies, dass durch den höheren Grenzwert ein möglicher Mangel an Protokollspeicherbereich nahezu ausgeschlossen ist. Wenn die LSNs jedoch bis zu dieser Größe zunehmen, wird die Nachricht ADM1849C zuerst in die Protokolle geschrieben, wenn der LSN-Wert 0xFFFF EFFF FFFF FFEF erreicht. Wenn der LSN-Wert 0xFFFF FFFE FFFF FFEF erreicht, wird die Datenbank schreibgeschützt (ADM1850C).

Überwachung der Zunahme der Protokollfolgenummern

Wenn Sie die Überwachung der Zunahme der Protokollfolgenummern (LSNs) umgehen wollen, müssen Sie ein Upgrade auf DB2 Version 9.7 durchführen. Diese Produktversion hat höhere LSN-Grenzwerte, was eine Überwachung der LSN-Zunahme praktisch überflüssig macht.

Wenn ein Upgrade auf DB2 Version 9.7 keine Option ist, müssen Sie die Zunahme der LSNs überwachen, um rechtzeitig die erforderlichen Aktionen ausführen zu können. Andernfalls wird Ihre Datenbank möglicherweise unerwartet schreibgeschützt, wenn die LSNs den zulässigen Höchstwert von 0xFFFF 0000 0000 überschreiten.

Für die Überwachung der LSN-Zunahme ist es erforderlich, regelmäßig die folgenden Schritte auszuführen:

1. Prüfen Sie die aktuelle Protokollfolgenummer. In Umgebungen mit partitionierten Datenbanken müssen Sie die LSN-Werte auf jeder einzelnen Partition überwachen.
2. Berechnen Sie mithilfe der aktuellen Protokollfolgenummer und einer Protokollfolgenummer, die von einem bekannten Zeitpunkt in der Vergangenheit stammt, die Rate, mit der neue LSNs zugeordnet werden.
3. Prognostizieren Sie mithilfe der LSN-Zunahmerate, wie lange es dauert, bis der zulässige LSN-Höchstwert erreicht ist und die Datenbank schreibgeschützt wird.

Ermitteln der aktuellen Protokollfolgenummer in einer Datenbank:

Sie können die aktuelle Protokollfolgenummer (LSN) einer Datenbank prüfen, um zu ermitteln, wie stark die LSNs noch zunehmen können. In Umgebungen mit partitionierten Datenbanken müssen Sie die LSNs jeder einzelnen Partition untersuchen, weil die LSNs auf den verschiedenen Datenbankpartitionen unabhängig voneinander zunehmen.

(Weitere Informationen zu den Grenzwerten für Protokollfolgenummern finden Sie im Abschnitt „Obergrenzen für Protokollfolgenummern" auf Seite 235.)

Die aktuelle Protokollfolgenummer einer gegebenen Datenbank lässt sich auf zweierlei Weise ermitteln.

- Befehl db2pd
- Gespeicherte Prozedur getcurrentlsn (Download von IBM.com erforderlich)

In den nachstehenden Anweisungen wird erläutert, wie die Protokollfolgenummer mit dem Befehl db2pd ermittelt wird. Weitere Informationen zur Verwendung dieser gespeicherten Prozedur finden Sie bei dem technischen Hinweis (TechNote) „LSN limit in DB2 V9.1 and V9.5, and ADM1849C or ADM1850C messages" (http://g01zciwas003.ahe.pok.ibm.com/support/dcf/preview.wss?host=g01zcidbs003.ahe.pok.ibm.com&db=support/swg/dmgtech.nsf&unid=59305F2F4EC72588852576C400650B02&taxOC=SSEPGG&MD=2010/04/20%2018:04:30&sid=).

In DB2 Version 9.5 Fixpack 4 oder niedriger müssen Sie die aktuelle Protokollfolgenummer auf der Basis des Bereichs der LSNs ableiten, die der aktuellen Protokolldatei zugeordnet sind. Sie können den exakten Wert der aktuellen Protokollfolgenummer nicht ermitteln. Vielmehr müssen Sie auf der Basis des möglichen Bereichs, in dem sich die Protokollfolgenummer befindet, eine vorsichtige Vermutung über ihren *möglichen Wert* anstellen. Genauere Informationen dazu finden Sie in den nachfolgenden Schritten. Im Gegensatz dazu wird die Protokollfolgenummer in Version 9.5 Fixpack 5 und höher explizit in den Kopfzeileninformationen der Protokolldatei angezeigt.

Verwenden Sie die folgenden Anweisungen, um die aktuelle Protokollfolgenummer zu ermitteln:

1. Führen Sie den Befehl db2pd mit der Option **-logs** aus. Formulieren Sie den Befehl bei einer Datenbank namens SALES beispielsweise wie folgt:

   ```
   db2pd -logs -db SALES
   ```

2. Untersuchen Sie zur Ermittlung der aktuellen Protokollfolgenummer die Ausgabe des vorangegangenen Befehls:
   - Führen Sie bei DB2 Version 9.5 Fixpack 4 oder niedriger die folgenden Schritte aus:
     a. Ermitteln Sie das aktuelle Protokoll.
     b. Suchen Sie in der nächsthöheren Protokolldatei nach der ersten Protokollfolgenummer. Gehen Sie zur Ermittlung der Menge des LSN-Freiraums davon aus, dass dies die aktuelle Protokollfolgenummer ist. In Wirklichkeit handelt es sich nicht um die aktuelle Protokollfolgenummer, sondern um eine vorsichtige Schätzung, da sie wissen, dass dies der größtmögliche LSN-Wert zum Zeitpunkt der Ausführung des Befehls db2pd war.

     Gehen Sie beispielsweise von der folgenden Ausgabe des Befehls db2pd aus:

```
Database Partition 0 -- Database TEST -- Active -- Up 0 days 00:00:02

Logs:
Current Log Number 0  1
Pages Written                0
Method 1 Archive Status      n/a
Method 1 Next Log to Archive n/a
Method 1 First Failure       n/a
Method 2 Archive Status      n/a
Method 2 Next Log to Archive n/a
Method 2 First Failure       n/a

Address            StartLSN       State      Size  Pages  Filename
0x07000000402EFF38 0x000001F88000 0x00000000 1024 1024  S0000000.LOG ← 2
0x0700000004005EF78 0x000002388000 0x00000000 1024 1024  S0000001.LOG
0x0700000004001FEB8 0x000002788000 0x00000000 1024 1024  S0000002.LOG
0x0700000004001FF78 0x000002B88000 0x00000000 1024 1024  S0000003.LOG
0x0700000004005FE78 0x000002F88000 0x00000000 1024 1024  S0000004.LOG
0x0700000004005FF38 0x000003388000 0x00000000 1024 1024  S0000005.LOG
0x07000000402F7AD8 0x000003788000 0x00000000 1024 1024  S0000006.LOG
0x07000000402F7B98 0x000003B88000 0x00000000 1024 1024  S0000007.LOG
0x07000000402F7C58 0x000003F88000 0x00000000 1024 1024  S0000008.LOG
0x07000000402F7D18 0x000004388000 0x00000000 1024 1024  S0000009.LOG
0x07000000402F7DD8 0x000004788000 0x00000000 1024 1024  S0000010.LOG
0x07000000402F7E98 0x000004B88000 0x00000000 1024 1024  S0000011.LOG
0x07000000402F7F58 0x000004F88000 0x00000000 1024 1024  S0000012.LOG
```

Abbildung 17. Ausgabe des Befehls db2pd -logs bei DB2 Version 9.5 Fixpack 4 oder niedriger

     Diesem Bericht können Sie entnehmen, dass die aktuelle Protokollnummer 0 **1** lautet. Dies entspricht der Protokolldatei `S00000000.LOG` **2** . Betrachten Sie nun die *nächste* Protokolldatei, `S00000001.LOG`. Die erste Protokollfolgenummer in dieser Protokolldatei ist 0x0000 0238 8000. Zur Schätzung des Freiraums für die LSN-Zunahme können Sie davon ausgehen, dass dies die aktuelle Protokollfolgenummer ist.
   - Suchen Sie bei DB2 Version 9.5 Fixpack 5 oder höher in der Kopfzeile der Ausgabe des Befehls db2pd -logs nach der aktuellen Protokollfolgenummer. Nachstehend finden Sie ein Beispiel für die Ausgabe des Befehls db2pd -logs bei Version 9.5 Fixpack 5:

```
Database Partition 0 - Database TEST - Active - Up 3
days 00:00:30

Logs:
Current Log Number            0
Pages Written                 612
Cur Commit Disk Log Reads     0
Cur Commit Total Log Reads    0
Method 1 Archive Status       n/a
Method 1 Next Log to Archive  n/a
Method 1 First Failure        n/a
Method 2 Archive Status       n/a
Method 2 Next Log to Archive  n/a
Method 2 First Failure        n/a
Log Chain ID                  0
Current LSN 0x000000000452CABB
```

Abbildung 18. Ausgabe des Befehls db2pd -logs bei DB2 Version 9.5 Fixpack 5 oder höher

Diesem Bericht können Sie entnehmen, dass die aktuelle Protokollfolgenummer für diese Datenbank 0x0000 0000 0452 CABB lautet.

Anmerkung: Zwar wird die Protokollfolgenummer als Wert im 8-Byte-Format angezeigt, doch ist die Protokollfolgenummer in Wirklichkeit nur eine 6-Byte-Nummer.

Berechnen der Zunahmeraten von Protokollfolgenummern:

Erfassen Sie mindestens zwei Datengruppen über einen bestimmten Zeitraum, um die LSN-Verwendungsrate zu messen. Wenn Sie zwischen den Erfassungen der einzelnen Datengruppen einen längeren Zeitraum verstreichen lassen, wird die LSN-Verwendungsrate genauer berechnet. Sie können LSN-Daten auch über einen großen Workloadbereich erfassen, sodass periodische Schwankungen der LSN-Verwendung einbezogen werden.

Verwenden Sie zur Berechnung der LSN-Verwendungsrate die folgende Formel:

$$\text{LSNUsageRate} = (\text{LSN}_2 - \text{LSN}_1) \ / \ (\text{Time}_2 - \text{Time}_1)$$

Dabei gilt Folgendes:

LSN_1 Ist der Wert der Protokollfolgenummer, die bei der Erfassung der ersten Stichprobendaten ermittelt wurde.

LSN_2 Ist der Wert der Protokollfolgenummer, die bei der Erfassung der zweiten Stichprobendaten ermittelt wurde.

Time_1 Ist der Zeitpunkt der Erfassung der ersten Stichprobendaten.

Time_2 Ist der Zeitpunkt der Erfassung der zweiten Stichprobendaten.

1. Ermitteln Sie mithilfe des Befehls db2pd -logs die Werte für LSN_1 und Time_1. Details hierzu finden Sie im Abschnitt „Ermitteln der aktuellen Protokollfolgenummer in einer Datenbank” auf Seite 238.
2. Wiederholen Sie Schritt 1 später, um die Werte für LSN_2 und Time_2 zu ermitteln.
3. Berechnen Sie die LSN-Verwendungsrate mit der oben angegebenen Formel. Gehen Sie beispielsweise davon aus, dass die folgenden Daten erfasst wurden:

Tabelle 12. An zwei Zeitpunkten erfasste LSN-Stichprobendaten

LSN_1	0x000002388000
$Time_1$	00:00:00
LSN_2	0x000002F88000
$Time_2$	12:00:00

Daraus ergibt sich folgende LSN-Verwendungsrate:

```
 (0x000002F88000 - 0x000002388000) / (12:00:00 - 00:00:00)
= (49.840.128 Byte - 37.257.216 Byte)/12 Stunden
= 1.048.576 Byte/Stunde
= 0x0000 0010 0000 Byte/Stunde
```

Mit der folgenden Formal können Sie berechnen, wie lange es dauert, bis der LSN-Höchstwert (0xFFFF 0000 0000) erreicht ist:

(0xFFFF 0000 0000 - current_LSN) / LSNUsageRate

Wenn man auf der Grundlage des obigen Beispiels davon ausgeht, dass LSN_2 die aktuelle Protokollfolgenummer ist, ergibt sich der folgende Zeitraum bis zum Erreichen des LSN-Maximalwerts:

```
 (0xFFFF 0000 0000 Byte - (0x000002F88000 Byte) / 0x0000 0010 0000 Byte/Stunde
= (281.470.681.743.360 - 49.840.128 Byte) / 1.048.576 Byte/Stunde
= 268.431.312,5 Stunden
= 11.184.638 Tage
```

Beheben eines Zustands eines erreichten LSN-Grenzwerts

Wenn Sie Version 9.5 Fixpack 3 installiert haben und eine der beiden Nachrichten erhalten, die anzeigen, dass sich die LSN dem maximalen Grenzwert nähert oder diesen erreicht hat (ADM1849C, ADM1850C), können Sie das Problem am einfachsten beheben, in dem Sie ein Upgrade auf DB2 Version 9.7 durchführen.

Version 9.7 besitzt einen erweiterten LSN-Grenzwert von 16 Exbibyte (16×1024^6 Byte), während ältere Versionen von DB2 einen Grenzwert von 256 Terabyte (256×1024^4 Byte) aufweisen. Durch die Erhöhung des Grenzwerts wird praktisch ausgeschlossen, dass der LSN-Grenzwert jemals erreicht wird.

Wichtig: Wenn Sie Version 9.5 Fixpack 3 nicht installiert haben, wird keine Fehlernachricht in die Protokolldatei geschrieben, wenn sich die LSNs dem Maximalwert nähern. Wenn Sie die LSN-Zunahme nicht überwacht und den LSN-Höchstwert überschritten haben, ist die Datenbank beschädigt. Stellen Sie die Datenbank von einem Backup wieder her, das Sie vor Erreichen des LSN-Grenzwerts erstellt haben. Je nachdem, wie nahe die LSN im Backup-Image am LSN-Höchstwert ist, bemerken Sie nach der Wiederherstellung möglicherweise, dass die LSNs bald wieder bis zum Maximalwert zunehmen. In einer solchen Situation ist die Überwachung der LSNs wichtig.

Als Alternative können Sie Kontakt mit IBM Software Support aufnehmen, um weitere für Sie verfügbare Optionen in Erfahrung zu bringen.

Wenn ein Upgrade auf Version 9.7 keine Option ist, können Sie eine neue Datenbank erstellen, in die Sie die Daten aus der ursprünglichen Datenbank laden. Dies bewirkt, dass die Folgenummern in den Protokollen wieder bei 0 anfangen. Führen Sie die folgenden Schritte aus, um eine neue Datenbank zu erstellen:

1. Entladen Sie alle Daten in der Datenbank. Sie können diese Task mit dem Dienstprogramm EXPORT oder mit einem Produkt wie Optim High Performance Unload ausführen.

2. Erstellen Sie eine neue Datenbank.
3. Laden Sie die Daten erneut in die neue Datenbank.

Wenn Sie kein Upgrade auf Version 9.7 durchführen, müssen Sie die Überwachung der LSN-Zunahme fortsetzen.

Wichtig: Die LSNs werden nur zurückgesetzt, wenn die Daten erneut in eine neue Datenbank geladen werden. Bei der Wiederherstellung der Daten in eine vorhandene Datenbank mit dem Befehl RESTORE geschieht dies nicht.

Löschen von Datenbankrecoveryobjekten mit dem Befehl PRUNE HISTORY oder der API db2Prune

Sie können den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ und den Befehl PRUNE HISTORY oder die API db2Prune zum Löschen von Recoveryobjekten verwenden.

Wenn Sie den Befehl PRUNE HISTORY oder die API db2Prune aufrufen, verhält sich der IBM Data Server-Datenbankmanager wie folgt:

- Löscht Einträge aus der Datei des Recoveryprotokolls, die nicht den Status DB2HISTORY_STATUS_DO_NOT_DEL aufweisen

Wenn Sie den Befehl PRUNE HISTORY mit dem Parameter AND DELETE oder die API db2Prune mit DB2PRUNE_OPTION_DELETE aufrufen, verhält sich der Datenbankmanager wie folgt:

- Löscht Einträge aus der Datei des Recoveryprotokolls, die älter als eine von Ihnen festgelegte Zeitmarke sind und nicht den Status DB2HISTORY_STATUS_DO_NOT_DEL aufweisen
- Löscht die physischen Protokolldateien, die zu den bereinigten (gelöschten) Einträgen gehören

Wenn Sie den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON setzen und Sie den Befehl PRUNE HISTORY mit dem Parameter AND DELETE oder die API db2Prune mit DB2PRUNE_OPTION_DELETE aufrufen, verhält sich der Datenbankmanager wie folgt:

- Löscht Einträge aus der Datei des Recoveryprotokolls, die nicht den Status DB2HISTORY_STATUS_DO_NOT_DEL aufweisen
- Löscht die physischen Protokolldateien, die zu den bereinigten (gelöschten) Einträgen gehören
- Löscht die Backup-Images, die den bereinigten (gelöschten) Einträgen zugeordnet sind
- Löscht die Images der Ladekopien, die den bereinigten (gelöschten) Einträgen zugeordnet sind

Gehen Sie wie folgt vor, um nicht benötigte Recoveryobjekte zu löschen:

1. Setzen Sie den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON.
2. Rufen Sie den Befehl PRUNE HISTORY mit dem Parameter AND DELETE oder die API db2Prune mit DB2PRUNE_OPTION_DELETE auf.

Automatisieren der Verwaltung von Datenbankrecoveryobjekten

Sie können den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ und die automatische Bereinigung der Datei des Recoveryprotokolls verwenden, um den IBM Data Server-Datenbankmanager so zu konfigurieren, dass er nicht benötigte Recoveryobjekte nach jeder vollständigen Datenbankbackup-Operation automatisch löscht.

Nach jeder erfolgreichen vollständigen (nicht inkrementellen) Datenbankbackup-Operation entfernt der Datenbankmanager die Datei des Recoveryprotokolls in Übereinstimmung mit dem Wert der Konfigurationsparameter num_db_backup und rec_his_retentn:

- Sind mehr Datenbankbackup-Einträge in der Datei des Recoveryprotokolls vorhanden, als der Wert des Konfigurationsparameters num_db_backup zulässt, entfernt der Datenbankmanager alle Einträge aus der Datei des Recoveryprotokolls, die älter als der Wert des Konfigurationsparameters rec_his_retentn sind und nicht den Status DB2HISTORY_STATUS_DO_NOT_DEL aufweisen.

Wenn Sie den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON setzen, führt der Datenbankmanager folgende Aktionen zusätzlich zum Löschen der Einträge in der Datei des Recoveryprotokolls aus:

- Löscht die physischen Protokolldateien, die zu den bereinigten (gelöschten) Einträgen gehören
- Löscht die Backup-Images, die den bereinigten (gelöschten) Einträgen zugeordnet sind
- Löscht die Images der Ladekopien, die den bereinigten (gelöschten) Einträgen zugeordnet sind

Wenn für die Berücksichtigung im aktuellen Recovery-Protokoll keine Datenbankgesamtbackup-Images verfügbar sind (z. B. weil sie niemals erstellt wurden), werden alle Images, die älter als der in REC_HIS_RETENTN angegebene Zeitraum sind, gelöscht.

Wenn der Datenbankmanager eine Datei nicht löschen kann, weil die Datei sich nicht mehr in der Speicherposition befindet, die in der Datei des Recoveryprotokolls aufgelistet ist, löscht der Datenbankmanager den Protokolleintrag.

Wenn der Datenbankmanager eine Datei nicht löschen kann, weil zwischen dem Datenbankmanager und dem Speichermanager oder der Speichereinheit ein Kommunikationsfehler aufgetreten ist, löscht der Datenbankmanager den Protokolleintrag nicht. Wenn der Fehler beseitigt worden ist, kann die betreffende Datei während der nächsten automatischen Bereinigung gelöscht werden.

Gehen Sie wie folgt vor, um den Datenbankmanager so zu konfigurieren, dass nicht benötigte Recoveryobjekte automatisch gelöscht werden:

1. Setzen Sie den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON.
2. Richten Sie die Konfigurationsparameter rec_his_retentn und num_db_backups so ein, dass die automatische Bereinigung der Datei des Recoveryprotokolls aktiviert ist.

Schutz von Recoveryobjekten vor dem Löschen

Die automatische Verwaltung von Recoveryobjekten spart Verwaltungszeit und Speicherbereich. Es kann jedoch sinnvoll sein, bestimmte Recoveryobjekte davor zu schützen, automatisch gelöscht zu werden. Sie können verhindern, dass Schlüsselrecoveryobjekte gelöscht werden, indem Sie den Status der zugehörigen Einträge in der Datei des Recoveryprotokolls auf do_not_delete (Nicht löschen) setzen.

Informationen zu dieser Task

Wenn Sie den Datenbankkonfigurationsparameter AUTO_DEL_REC_OBJ auf ON setzen, werden Recoveryobjekte gelöscht, wenn ihre zugehörigen Einträge in der Datei des Recoveryprotokolls bereinigt (gelöscht) werden. Die Einträge in der Datei des Recoveryprotokolls werden bereinigt, wenn eines der folgenden Ereignisse eintritt:

- Sie rufen den Befehl PRUNE HISTORY zusammen mit dem Parameter AND DELETE auf
- Sie rufen die API db2Prune mit der Markierung DB2PRUNE_OPTION_DELETE auf
- Der Datenbankmanager bereinigt automatisch die Datei des Recoveryprotokolls, was nach jedem erfolgreichen Gesamtbackup eines Tabellenbereichs oder einer Datenbank geschieht.

Unabhängig davon, ob Sie den Befehl PRUNE HISTORY oder die API db2Prune aufrufen, oder den Datenbankmanager so konfigurieren, dass automatisch die Einträge in der Datei des Recoveryprotokolls bereinigt werden, werden Einträge mit der Markierung do_not_delete nicht bereinigt, und die zugehörigen Recoveryobjekte werden nicht gelöscht.

Vorgehensweise

Verwenden Sie den Befehl UPDATE HISTORY, um den Status von zugehörigen Einträgen in der Datei des Recoveryprotokolls auf do_no_delete zu setzen. Einschränkungen:

- Sie können nur den Status von Backup-Images, Ladekopie-Images und Protokolldateien auf do_not_delete setzen.
- Der Status eines Backup-Eintrags wird nicht an Ladekopie-Images, nicht inkrementelle Backups oder Protokolldateien weitergegeben, die zu der Backup-Operation gehören. Wenn Sie einen bestimmten Datenbankbackup-Eintrag und seine zugeordneten Protokolldateieinträge sichern möchten, müssen Sie jeweils den Status für den Datenbankbackup-Eintrag und den Eintrag für alle zugehörigen Protokolldateien festlegen.

Verwalten von Momentaufnahmebackup-Objekten

Zur Verwaltung von Momentaufnahmebackup-Objekten müssen Sie den Befehl db2acsutil verwenden. Versetzen oder verschieben Sie keine Momentaufnahmebackup-Objekte mit Dateisystemdienstprogrammen.

Vorbereitung

Zur Ausführung von Backup- und Restoreoperationen für Momentaufnahmen benötigen Sie einen DB2 ACS-API-Treiber für Ihre Speichereinheit. In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS Series

Bevor Sie mit der Verwaltung von Momentaufnahmebackup-Objekten beginnen können, müssen Sie DB2 Advanced Copy Services (ACS) aktivieren. Siehe hierzu „Aktivieren von DB2 ACS" auf Seite 365.

Einschränkungen

Der Befehl db2acsutil wird derzeit nur unter AIX und Linux unterstützt.

Vorgehensweise

1. Verwenden Sie den Parameter **QUERY**, um eine Liste der verfügbaren Momentaufnahmebackup-Objekte aufzurufen.

   Verwenden Sie beispielsweise die folgende Syntax, um die verfügbaren Momentaufnahmebackup-Objekte für die Datenbankmanagerinstanz `dbminst1` aufzulisten:

   ```
   db2acsutil query instance dbminst1
   ```

2. Verwenden Sie den Parameter **STATUS**, um den Fortschritt einer bestimmten Momentaufnahmebackup-Operation zu überprüfen.

   Verwenden Sie beispielsweise die folgende Syntax, um den Fortschritt von Momentaufnahmebackup-Operationen anzuzeigen, die momentan möglicherweise auf der Datenbank `database1` ausgeführt werden:

   ```
   db2acsutil query status db database1
   ```

3. Verwenden Sie den Parameter **DELETE**, um ein bestimmtes Momentaufnahmebackup-Objekt zu löschen.

   Verwenden Sie beispielsweise die folgende Syntax, um alle Momentaufnahmebackup-Objekte für die Datenbank `database1` zu löschen, die älter als 10 Tage sind:

   ```
   db2acsutil delete older than 10 days ago db database1
   ```

Kapitel 9. Überwachen des Fortschritts von Backup, Restore- und Recoveryoperationen

Sie können den Befehl LIST UTILITIES verwenden, um Backup-, Restore- und aktualisierende Recoveryoperationen für eine Datenbank zu überwachen.

Gehen Sie wie folgt vor, um den Fortschritt von Backup-, Restore- und Recoveryoperationen zu überwachen:

Setzen Sie den Befehl LIST UTILITIES ab, und geben Sie die Option SHOW DETAIL an.

```
list utilities show detail
```

Es folgt eine Beispielausgabe der Leistungsüberwachung einer Offline-Backup-Operation für eine Datenbank:

```
LIST UTILITIES SHOW DETAIL

Kennung                       = 2
   Typ                        = BACKUP
   Datenbankname              = SAMPLE
   Beschreibung               = offline db
   Startzeit                  = 2003-10-30 12.55.31.786115
Drosselung:
   Priorität                  = Ungedrosselt
   Fortschrittsüberwachung:
   Geschätzte Fertigstellung (%)     = 41
               Gesamte Arbeit        = 20232453 Byte
               Abgeschlossene Arbeit = 230637 Byte
         Startzeit                   = 2003-10-30 12.55.31.786115
```

Für Backup-Operationen wird eine erste geschätzte Anzahl zu verarbeitender Byte angegeben. Während der Backup-Operation wird die geschätzte Anzahl zu verarbeitender Byte aktualisiert. Die angezeigten Byte entsprechen nicht der Größe des Images und sollten nicht zur Schätzung der Größe des Backup-Images verwendet werden. Das tatsächliche Image kann wesentlich kleiner sein, abhängig davon, ob es sich um ein inkrementelles Backup oder ein komprimiertes Backup handelt.
Für Restoreoperationen wird kein geschätzter Anfangswert angegeben. Stattdessen wird UNBEKANNT angegeben. Während die einzelnen Puffer aus dem Image gelesen werden, wird die tatsächliche Menge gelesener Byte aktualisiert. Für automatische inkrementelle Restores, bei denen möglicherweise mehrere Images wiederhergestellt werden, wird der Fortschritt anhand von Phasen überwacht. Jede Phase stellt ein Image dar, das aus der Kette von inkrementellen Backups wiederhergestellt werden soll. Zu Beginn wird nur eine Phase angegeben. Nachdem das erste Image wiederhergestellt wurde, wird die Gesamtanzahl der Phasen angezeigt. Beim Wiederherstellen der einzelnen Images werden die Anzahl der abgeschlossenen Phasen und die Anzahl der verarbeiteten Byte aktualisiert.
Für die Recovery nach einem Systemabsturz und die aktualisierende Recovery sind zwei Phasen der Fortschrittsüberwachung vorhanden: FORWARD und BACKWARD. In der Phase FORWARD werden Protokolldateien gelesen und die Protokollsätze auf die Datenbank angewendet. Für die Recovery nach einem Systemabsturz wird unter Verwendung der Protokollstartfolgenummer bis zum Ende der letzten Protokolldatei der Gesamtaufwand geschätzt. Für die aktualisierende Recovery wird zu Beginn dieser Phase für den geschätzten Gesamtaufwand UNBEKANNT angegeben. Die Menge der verarbeiteten Byte wird während der Ausführung des Prozesses aktualisiert.

Während der Phase BACKWARD werden alle nicht festgeschriebenen Änderungen, die während der Phase FORWARD angewendet wurden, rückgängig gemacht. Es wird eine Schätzung der Menge der zu verarbeitenden Protokolldaten in Byte angegeben. Die Menge der verarbeiteten Byte wird während der Ausführung des Prozesses aktualisiert.

Statuswerte für Tabellenbereiche

Der aktuelle Zustand eines Tabellenbereichs wird von seinem *Status* wiedergegeben. Zu den häufigsten bei der Recovery auftretenden Status gehören folgende:

- *Backup anstehend*. Ein Tabellenbereich wird in diesen Status versetzt, wenn eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt durchgeführt wurde, oder im Anschluss an eine LOAD-Operation, die mit der Option NO COPY ausgeführt wurde. Der Tabellenbereich muss gesichert werden, bevor er verwendet werden kann. (Wird kein Backup erstellt, kann der Tabellenbereich nicht aktualisiert werden, es sind jedoch Operationen mit Lesezugriff möglich.)
- *Restore anstehend*. Ein Tabellenbereich wird in diesen Status versetzt, wenn eine für den Tabellenbereich ausgeführte Operation zur aktualisierenden Recovery abgebrochen wird oder einen Fehler feststellt, der auf Nichtwiederherstellbarkeit hinweist. Ist Letzeres der Fall, muss der Tabellenbereich wiederhergestellt und anschließend erneut aktualisierend wiederhergestellt werden. Ein Tabellenbereich wird außerdem in diesen Status versetzt, wenn der Tabellenbereich bei einer RESTORE-Operation nicht wiederhergestellt werden kann.
- *Aktualisierende Recovery läuft*. Ein Tabellenbereich wird in diesen Status versetzt, während für den Tabellenbereich eine Operation zur aktualisierenden Recovery ausgeführt wird. Nach Beendigung dieser Operation zur aktualisierenden Recovery befindet sich der Tabellenbereich nicht mehr in diesem Status. Der Tabellenbereich kann diesen Status auch bei Abbruch der Operation zur aktualisierenden Recovery verlieren.
- *Aktualisierende Recovery anstehend*. Ein Tabellenbereich wird in diesen Status versetzt, nachdem sein Restore beendet wurde oder ein E/A-Fehler aufgetreten ist. Nach seinem Restore kann der Tabellenbereich bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt werden. Nach einem E/A-Fehler muss die aktualisierende Recovery des Tabellenbereichs bis zum Ende der Protokolle erfolgen.

Kapitel 10. Backup - Übersicht

In der einfachsten Form des DB2-Befehls BACKUP DATABASE müssen Sie lediglich den Aliasnamen der Datenbank angeben, die Sie sichern möchten. Beispiel:

```
db2 backup db sample
```

Wird der Befehl erfolgreich ausgeführt, wird ein neues Backup-Image in dem Verzeichnis erstellt, in dem der Befehl abgesetzt wurde. Das Image wird in diesem Verzeichnis erstellt, da der Befehl in diesem Beispiel kein bestimmtes Zielverzeichnis angibt. Backup-Images, die mit DB2 Version 9.5 oder einer neueren Version erstellt werden, werden mit dem Dateimodus 600 generiert. Dies bedeutet, dass unter UNIX nur der Instanzeigner über Lese- und Schreibzugriffsrechte verfügt und dass unter Windows nur Mitglieder der Gruppe DB2ADMNS (und Administratoren) über Zugriff auf die Backup-Images verfügen.

Anmerkung: Wenn sich der DB2-Client und der -Server nicht auf demselben System befinden, ermittelt DB2 das aktuelle Arbeitsverzeichnis auf der Clientmaschine und verwendet dieses Verzeichnis als Zielverzeichnis für das Backup auf dem Server. Aus diesem Grund wird empfohlen, dass Sie für das Backup-Image ein Zielverzeichnis angeben.

Backup-Images werden an der Zielposition erstellt, die Sie beim Aufrufen des Backup-Dienstprogramms angeben. Folgende Positionen sind möglich:

- Ein Verzeichnis (für Backups auf Platte oder Diskette)
- Eine Einheit (für Backups auf Band)
- Ein TSM-Server (TSM = Tivoli Storage Manager)
- Der Server eines anderen Lieferanten

Die Datei des Recoveryprotokolls wird jedes Mal automatisch mit den Ergebnisinformationen aktualisiert, wenn Sie eine Backup-Operation für eine Datenbank starten. Diese Datei wird im selben Verzeichnis wie die Datenbankkonfigurationsdatei erstellt.

Wenn Sie alte Back-Images löschen wollen, die nicht länger benötigt werden, können Sie die Dateien entfernen, wenn die Backups in Form von Dateien gespeichert sind. Wenn Sie nachfolgend den Befehl LIST HISTORY mit der Option BACKUP ausführen, werden auch Informationen zu den gelöschten Backup-Images zurückgegeben. Um diese Einträge aus der Datei des Recoveryprotokolls zu entfernen, müssen Sie den Befehl PRUNE ausführen.

Wenn Ihre Recoveryobjekte mit Tivoli Storage Manager (TSM) gespeichert wurden, können Sie die Recoveryobjekte mithilfe des Dienstprogramms 'db2adutl' abfragen, extrahieren, prüfen und löschen. Unter Linux und UNIX befindet sich dieses Dienstprogramm im Verzeichnis `sqllib/adsm`; bei Windows-Betriebssystemen finden Sie es im Verzeichnis `sqllib\bin`. Verwenden Sie für Momentaufnahmem das Dienstprogramm db2acsutil im Verzeichnis `sqllib/bin`.

Unter allen Betriebssystemen setzen sich die Dateinamen für auf Platte erstellte Backup-Images aus verschiedenen durch Punkte voneinander getrennten Elementen zusammen:

```
DB_alias.typ.instanzname.NODEnnnn.CATNnnnn.zeitmarke.folgenr
```

Beispiel:

```
STAFF.0.DB201.NODE0000.CATN0000.19950922120112.001
```

Anmerkung: DB2 Universal Database Version 8.2.2 und frühere Versionen verwendeten eine vierstufige Unterverzeichnisstruktur zum Speichern von Backup-Images auf Windows-Betriebssystemen:

```
DB_alias.typ\instanzname\NODEnnnn\CATNnnnn\jjjjmmtt\hhmmss.folgenr
```

Beispiel:

```
SAMPLE.0\DB2\NODE0000\CATN0000\20010320\122644.001
```

DB_alias
Der aus 1 bis 8 Zeichen bestehende Datenbankaliasname, der beim Aufrufen des Backup-Dienstprogramms angegeben wurde.

Typ Der Typ der Backup-Operation. Dabei gilt Folgendes: `0` steht für ein Backup der gesamten Datenbank, `3` steht für ein Backup auf Tabellenbereichsebene, und `4` steht für ein Backup-Image, das mit dem Befehl LOAD...COPY TO generiert wird.

instanzname
Der aus 1 bis 8 Zeichen bestehende Name der aktuellen Instanz, der der Umgebungsvariablen DB2INSTANCE entnommen wird.

NODEnummer
Die Datenbankpartitionsnummer. In Umgebungen mit nur einer Datenbankpartition lautet diese Nummer stets `NODE0000`. In Umgebungen mit partitionierten Datenbanken lautet die Nummer `NODE`*xxxx*, wobei *xxxx* die Nummer angibt, die der Datenbankpartition in der Datei `db2nodes.cfg` zugeordnet ist.

CATNnummer
Die Datenbankpartitionsnummer der Katalogpartition für die Datenbank. In Umgebungen mit nur einer Datenbankpartition lautet diese Nummer stets `CATN0000`. In Umgebungen mit partitionierten Datenbanken lautet die Nummer `CATN`*xxxx*, wobei *xxxx* die Nummer angibt, die der Datenbankpartition in der Datei `db2nodes.cfg` zugeordnet ist.

zeitmarke
Ein Wert aus 14 Zeichen, der das Datum und die Uhrzeit des Zeitpunkts angibt, zu dem das Backup durchgeführt wurde. Die Zeitmarke hat das Format *jjjjmmtthhnnss*. Dabei gilt Folgendes:

- *jjjj* steht für das Jahr (1995 bis 9999).
- *mm* steht für den Monat (01 bis 12).
- *tt* steht für den Tag des Monats (01 bis 31).
- *hh* steht für die Stunde (00 bis 23).
- *nn* steht für die Minuten (00 bis 59).
- *ss* steht für die Sekunden (00 bis 59).

folgenr
Eine dreistellige Zahl, die als Dateierweiterung verwendet wird.

Wenn ein Backup-Image auf Band geschrieben wird, gilt Folgendes:

- Es werden keine Dateinamen erstellt, sondern die obigen Informationen im Backup-Header zu Prüfzwecken gespeichert.
- Es muss eine Bandeinheit über die Standardschnittstelle des Betriebssystems verfügbar sein.

In einer großen Umgebung mit partitionierten Datenbanken ist es jedoch unter Umständen nicht praktisch, eine dedizierte Bandeinheit für jeden Datenbankpartitionsserver bereitzuhalten. Sie können die Bandeinheiten mit einem oder mehreren TSM-Servern verbinden, sodass der Zugriff auf diese Bandeinheiten jedem Datenbankpartitionsserver ermöglicht wird.

- In einer Umgebung mit partitionierten Datenbanken können Sie darüber hinaus Produkte verwenden, die Funktionen für virtuelle Bandeinheiten implementieren, wie zum Beispiel REELlibrarian 4.2 oder CLIO/S. Mithilfe dieser Produkte können Sie auf eine mit anderen Knoten (Datenbankpartitionsservern) verbundene Bandeinheit über eine Pseudobandeinheit zugreifen. Der Zugriff auf die ferne Bandeinheit wird transparent ermöglicht, während der Zugriff auf die Pseudobandeinheit über die Standardschnittstelle des Betriebssystems erfolgen kann.

Ein Backup für eine Datenbank, die sich nicht im normalen Status oder im Status 'Backup anstehend' befindet, ist nicht möglich. Für einen Tabellenbereich, der sich im normalen Status oder im Status 'Backup anstehend' befindet, kann ein Backup durchgeführt werden. Wenn sich der Tabellenbereich nicht im normalen Status oder im Status 'Backup anstehend' befindet, kann ein Backup zugelassen werden.

Gleichzeitig ablaufende Backup-Operationen für denselben Tabellenbereich sind nicht zulässig. Sobald eine Backup-Operation für einen Tabellenbereich eingeleitet wurde, schlagen alle nachfolgenden Versuche fehl (SQL2048).

Wenn eine Datenbank oder ein Tabellenbereich nur teilweise wiederhergestellt wurde, da während der Restoreoperation ein Systemabsturz auftrat, müssen Sie die Datenbank bzw. den Tabellenbereich erfolgreich wiederherstellen, bevor Sie ein Backup-Image davon erstellen können.

Das Backup schlägt fehl, wenn die Liste der zu sichernden Tabellenbereiche den Namen eines temporären Tabellenbereichs enthält.

Das Backup-Dienstprogramm ermöglicht die Steuerung des gemeinsamen Zugriffs für mehrere Prozesse, die Backupkopien verschiedener Datenbanken anlegen. Diese Steuerung des gemeinsamen Zugriffs gewährleistet, dass die Zieleinheiten für das Backup bis zur Beendigung aller Backup-Operationen geöffnet bleiben. Wenn während der Backup-Operation ein Fehler auftritt und ein geöffneter Container nicht geschlossen werden kann, erhalten andere Backup-Operationen mit demselben Ziellaufwerk eventuell Nachrichten über Zugriffsfehler. Zur Behebung etwaiger Zugriffsfehler müssen Sie die Backup-Operation, die den Fehler verursachte, beenden und die Verbindung zur Zieleinheit trennen. Wenn Sie das Backup-Dienstprogramm für gleichzeitig ablaufende Backup-Operationen auf Band verwenden, müssen Sie sicherstellen, dass diese Operationen nicht dasselbe Band ansteuern.

Anzeigen von Backup-Informationen

Mit dem Dienstprogramm db2ckbkp können Sie Informationen zu vorhandenen Backup-Images anzeigen. Dieses Dienstprogramm bietet folgende Möglichkeiten:

- Testen der Integrität des Backup-Images und Feststellen, ob ein Wiederherstellen möglich ist.
- Anzeigen der im Backup-Header gespeicherten Informationen.
- Anzeigen von Informationen zu den Objekten und des Protokolldateiheaders im Backup-Image.

Verwenden von Backup

Verwenden Sie den Befehl BACKUP DATABASE, um eine Kopie der Daten einer Datenbank zu erstellen und sie für den Fall eines Fehlers oder der Beschädigung der Originaldaten auf einem anderen Datenträger zu speichern. Sie können eine gesamte Datenbank, eine Datenbankpartition oder nur ausgewählte Tabellenbereiche mit Backup sichern.

Vorbereitung

Es braucht noch keine Verbindung zu der zu sichernden Datenbank zu bestehen: Das Backup-Datenbankdienstprogramm stellt automatisch eine Verbindung zu der angegebenen Datenbank her, die nach Abschluss der Backup-Operation beendet wird. Wenn bereits eine Verbindung zu der zu sichernden Datenbank besteht, wird diese Verbindung beim Absetzen des Befehls BACKUP DATABASE unterbrochen, und die Backup-Operation wird fortgesetzt.

Bei der Datenbank kann es sich um eine lokale oder ferne Datenbank handeln. Das Backup-Image bleibt auf dem Datenbankserver, sofern Sie kein Speicherverwaltungsprodukt wie Tivoli Storage Manager (TSM) oder DB2 Advanced Copy Services (ACS) verwenden.

Wenn Sie ein Offline-Backup durchführen und Sie die Datenbank mit der Anweisung ACTIVATE DATABASE aktiviert haben, müssen Sie die Datenbank inaktivieren, bevor Sie das Offline-Backup durchführen. Wenn die Datenbank über aktive Verbindungen verfügt, muss ein Benutzer mit der Berechtigung SYSADM eine Verbindung zu der Datenbank herstellen und die folgenden Befehle absetzen, um die Datenbank erfolgreich zu inaktivieren:

```
CONNECT TO aliasname_der_datenbank
QUIESCE DATABASE IMMEDIATE FORCE CONNECTIONS;
UNQUIESCE DATABASE;
TERMINATE;
DEACTIVATE DATABASE aliasname-der-datenbank
```

In einer Umgebung mit partitionierten Datenbanken haben Sie die folgenden Möglichkeiten: Mit dem Befehl BACKUP DATABASE können Sie Datenbankpartitionen einzeln sichern, mit dem Befehlsparameter **ON DBPARTITIONNUM** können Sie mehrere der Datenbankpartitionen auf einmal sichern und mit dem Parameter **ALL DBPARTITIONNUMS** können Sie alle Datenbankpartitionen gleichzeitig sichern. Mit dem Befehl LIST NODES können Sie die Datenbankpartitionen identifizieren, in denen Benutzertabellen enthalten sind, für die das Ausführen eines Backups sinnvoll ist.

Wenn Sie ein *Offline-Backup* in einer Umgebung mit partitionierten Datenbanken ausführen und kein SSV-Backup (SSV = Single System View, Einzelsystemsicht) verwenden, sollten Sie die Katalogpartition von allen anderen Datenbankpartitionen getrennt sichern. Zum Beispiel können Sie die Katalogpartition zuerst sichern und anschließend die anderen Datenbankpartitionen sichern. Dies ist notwendig, weil die Backup-Operation möglicherweise eine exklusive Datenbankverbindung mit der Katalogpartition erfordert, während deren die anderen Datenbankpartitionen keine Verbindung herstellen können. Wenn Sie ein *Online-Backup* ausführen, können alle Datenbankpartitionen (einschließlich der Katalogpartition) gleichzeitig und in beliebiger Reihenfolge gesichert werden.

Zum Backup von Datenbankpartitionen können Sie auch den Befehlseditor verwenden. Da diese Vorgehensweise jedoch die aktualisierende Recovery nicht unter-

stützt, empfiehlt es sich, die auf diesen Knoten vorhandene Datenbank regelmäßig zu sichern. Für den möglichen Fall einer Beschädigung der Datei 'db2nodes.cfg' ist es ratsam, zusammen mit allen Backup-Kopien, die Sie erstellen, auch eine Kopie dieser Datei zu speichern.

Auf einem System, in dem mit verteilten Anforderungen gearbeitet wird, werden die Backup-Operationen auf die Datenbank für verteilte Anforderungen sowie auf die Metadaten angewendet, die im Katalog dieser Datenbank gespeichert sind (Wrapper, Server, Kurznamen usw.). Datenquellenobjekte (Tabellen und Sichten) werden nicht gesichert, es sei denn, diese Objekte werden in der Datenbank für verteilte Anforderungen gespeichert.

Wenn eine Datenbank mit einem vorherigen Release des Datenbankmanagers erstellt, aber nicht auf das aktuelle Release migriert wurde, müssen Sie sie migrieren. Andernfalls ist es nicht möglich, eine Backup-Kopie der Datenbank anzulegen.

Einschränkungen

Für das Backup-Dienstprogramm gelten die folgenden Einschränkungen:

- Es ist nicht möglich, gleichzeitig ein Backup eines Tabellenbereichs und einen Restore eines Tabellenbereichs durchzuführen. Dies gilt auch dann, wenn Backup und Restore für unterschiedliche Tabellenbereiche erfolgen.
- Wenn Sie in der Lage sein wollen, in einer Umgebung mit partitionierten Datenbanken eine aktualisierende Recovery durchzuführen, müssen Sie die Datenbank auf den Knoten der Liste regelmäßig sichern und über mindestens ein Backup-Image der übrigen Knoten im System verfügen (auch von Knoten, die keine Benutzerdaten für diese Datenbank enthalten). In zwei Situationen ist ein Backup-Image einer Datenbankpartition auf einem Datenbankpartitionsserver erforderlich, der keine Benutzerdaten für die Datenbank enthält:
  - Sie haben dem Datenbanksystem einen Datenbankpartitionsserver hinzugefügt, nachdem das letzte Backup erstellt wurde, und müssen eine aktualisierende Recovery auf diesem Datenbankpartitionsserver durchführen.
  - Die punktuelle Recovery wird verwendet, wozu alle Datenbankpartitionen im System den Status 'aktualisierende Recovery anstehend' aufweisen müssen.
- Online-Backup-Operationen für DMS-Tabellenbereiche sind mit den folgenden Operationen nicht kompatibel:
  - Laden
  - Reorganisation (online und offline)
  - Löschen von Tabellenbereichen
  - Tabellenverkürzung
  - Indexerstellung
  - Verwenden des Parameters NOT LOGGED INITIALLY (wird mit den Anweisungen CREATE TABLE und ALTER TABLE verwendet)
- Wenn Sie versuchen, ein Offline-Backup einer Datenbank durchzuführen, die momentan aktiv ist, empfangen Sie einen Fehler. Sie können vor einem Offline-Backup sicherstellen, dass die Datenbank nicht aktiv ist, in dem Sie den Befehl DEACTIVATE DATABASE ausführen.

Informationen zu dieser Task

Das Backup-Dienstprogramm kann über den Befehlszeilenprozessor (CLP), den Assistenten **Datenbank sichern** in der Steuerzentrale, die Prozedur ADMIN_CMD

mit dem Parameter BACKUP DATABASE oder die Anwendungsprogrammierschnittstelle (API) db2Backup aufgerufen werden.

Das folgende Beispiel zeigt einen Befehl BACKUP DATABASE, der über den CLP abgesetzt wird:

```
db2 backup database sample to c:\DB2Backups
```

Vorgehensweise

Gehen Sie wie folgt vor, um den Assistenten **Datenbank sichern** zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis das Datenbank- oder Tabellenbereichsobjekt angezeigt wird, das Sie sichern möchten.
2. Klicken Sie mit der rechten Maustaste das Objekt an und wählen Sie **Sichern** im Kontextmenü aus. Der Assistent **Datenbank sichern** wird geöffnet.

Weitere Schritte

Ausführliche Informationen enthält die Kontexthilfefunktion der Steuerzentrale.

Wenn Sie ein Offline-Backup durchgeführt haben, müssen Sie nach der Ausführung die Datenbank reaktivieren:

```
ACTIVATE DATABASE sample
```

Durchführen eines Momentaufnahmebackups

Bei einer Momentaufnahmebackup-Operation wird zum Kopieren von Daten die leistungsstarke Kopiertechnologie einer Speichereinheit genutzt.

Vorbereitung

Zur Ausführung von Backup- und Restoreoperationen für Momentaufnahmen benötigen Sie einen DB2 ACS-API-Treiber für Ihre Speichereinheit. In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS Series

Vor der Durchführung eines Momentaufnahmebackups muss DB2 Advanced Copy Services (ACS) aktiviert werden. Siehe hierzu „Aktivieren von DB2 ACS" auf Seite 365.

Vorgehensweise

Sie können ein Momentaufnahmebackup unter Verwendung des Befehls BACKUP DATABASE mit dem Parameter **USE SNAPSHOT**, der API db2Backup mit dem Datenträgertyp SQLU_SNAPSHOT_MEDIA oder der Prozedur ADMIN_CMD mit den Parametern **BACKUP DATABASE** und **USE SNAPSHOT** durchführen. Befehl BACKUP DATABASE:

```
db2 backup db sample use snapshot
```

Prozedur ADMIN_CMD mit dem Parameter **BACKUP DATABASE**:

```
CALL SYSPROC.ADMIN_CMD
   ('backup db sample use snapshot')
```

API db2Backup:

```
int sampleBackupFunction( char dbAlias[],
                                                                  char user[],
                                                  char pswd[],
                          char workingPath[] )
{
  db2MediaListStruct mediaListStruct = { 0 };

  mediaListStruct.locations = &workingPath;
    mediaListStruct.numLocations = 1;
  mediaListStruct.locationType = SQLU_SNAPSHOT_MEDIA;


  db2BackupStruct backupStruct = { 0 };

  backupStruct.piDBAlias = dbAlias;
  backupStruct.piUsername = user;
  backupStruct.piPassword = pswd;
  backupStruct.piVendorOptions = NULL;
  backupStruct.piMediaList = &mediaListStruct;

  db2Backup(db2Version950, &backupStruct, &sqlca);

  return 0;
}
```

Ergebnisse

Nach der Durchführung eines Momentaufnahmebackups können Sie einen Restore für ein Momentaufnahmebackup durchführen.

Verwenden einer geteilten Spiegeldatenbank als Backup-Image

Verwenden Sie die folgende Prozedur, um eine geteilte Spiegeldatenbank einer Primärdatenbank zu erstellen, die als Backup-Image verwendet werden soll. Diese Prozedur kann anstelle von Datenbank-Backup-Operationen für die Primärdatenbank verwendet werden.

Gehen Sie wie folgt vor, um eine geteilte Spiegeldatenbank als „Backup-Image" zu verwenden:

1. Setzen Sie für die primäre Datenbank die E/A aus:

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

   Führen Sie keine weiteren Dienstprogramme oder Tools aus, während sich die Datenbank im ausgesetzten Status befindet. Das Erstellen einer Datenbankkopie sollte die einzige Aktivität sein.
2. Verwenden Sie geeignete Befehle auf Betriebssystemebene, um die Spiegeldatenbank(en) aus der primären Datenbank zu erstellen.

   Anmerkung: Stellen Sie sicher, dass Sie das gesamte Datenbankverzeichnis einschließlich des Datenträgerverzeichnisses kopieren. Sie müssen auch das Protokollverzeichnis und alle Containerverzeichnisse kopieren, die außerhalb des Datenbankverzeichnisses vorhanden sind. Eine Zusammenstellung dieser Informationen finden Sie in der Verwaltungssicht DBPATHS, in der alle Dateien und Verzeichnisse der Datenbank angezeigt werden, die geteilt werden müssen.

3. Nehmen Sie den E/A-Betrieb auf der primären Datenbank wieder auf:
   ```
   db2 set write resume for database
   ```
4. Auf dem primären System tritt ein Fehler auf, der einen Restore aus einem Backup erforderlich macht.
5. Stoppen Sie die Primärdatenbankinstanz:
   ```
   db2stop
   ```
6. Verwenden Sie Befehle auf Betriebssystemebene, um die geteilten Daten auf das primäre System zu kopieren. **Kopieren Sie nicht die geteilten Protokolldateien**, da die primären Protokolle für die aktualisierende Recovery benötigt werden.
7. Starten Sie die Primärdatenbankinstanz:
   ```
   db2start
   ```
8. Initialisieren Sie die Primärdatenbank:
   ```
   db2inidb aliasname-der-datenbank as mirror
   ```
9. Stellen Sie die primäre Datenbank bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt aktualisierend wieder her.

Backup auf Band

Wenn Sie Ihre Datenbank oder Ihren Tabellenbereich sichern wollen, müssen Sie die Block- und die Puffergröße korrekt definieren. Dies gilt besonders dann, wenn Sie mit variablen Blockgrößen arbeiten (z. B. unter AIX, wenn die Blockgröße auf null gesetzt wurde).

Beim Sichern gilt eine Einschränkung für die Anzahl der festen Blockgrößen, die verwendet werden können. Diese Einschränkung besteht, weil DB2-Datenbanksysteme den Backup-Image-Header als 4-KB-Block ausgeben. Die einzigen festen Blockgrößen, die von DB2-Datenbanksystemen unterstützt werden, sind 512, 1024, 2048 und 4096 Byte. Wenn Sie mit einer festen Blockgröße arbeiten, können Sie für das Backup eine beliebige Puffergröße angeben. Es ist jedoch möglich, dass Ihre Backup-Operation nicht erfolgreich abgeschlossen werden kann, wenn die feste Blockgröße nicht mit einem der von DB2-Datenbanksystemen unterstützten Werte übereinstimmt.

Wenn Ihre Datenbank umfangreich ist und Sie eine feste Blockgröße verwenden, benötigen Ihre Backup-Operationen möglicherweise mehr Zeit als erwartet. Zur Verbesserung der Leistung können Sie eine variable Blockgröße verwenden.

Anmerkung: Wenn Sie eine variable Blockgröße verwenden, stellen Sie sicher, dass Sie über erprobte Verfahren verfügen, die Ihnen eine erfolgreiche Recovery (einschließlich explizit angegebener Puffergrößen für die Befehle BACKUP und RESTORE) unter Verwendung von Backup-Images ermöglichen, die mit variabler Blockgröße erstellt wurden.

Bei der Verwendung variabler Blockgrößen müssen Sie eine Backup-Puffergröße angeben, die kleiner als der obere Grenzwert für die verwendete Bandeinheit oder gleich diesem Wert ist. Die Puffergröße muss dem oberen Grenzwert für die Blockgröße der verwendeten Einheit entsprechen, wenn Sie optimale Leistungswerte erzielen wollen.

Bevor eine Bandeinheit unter einem Windows-Betriebssystem verwendet werden kann, müssen Sie den folgenden Befehl absetzen:

```
db2 initialize tape on einheit using blkgröße
```

Dabei gilt Folgendes:

einheit Ein gültiger Bandeinheitenname. Der Standardwert bei Windows-Betriebssystemen ist `\\.\TAPE0`.

blkgröße
Der Blockungsfaktor für das Band. Er muss ein Faktor oder ein Vielfaches von 4096 sein. Der Standardwert ist die Standardblockgröße für die Einheit.

Wenn zum Restore ein mit variabler Blockgröße erstelltes Backup-Image verwendet wird, kann dies zu einer Fehlermeldung führen. In diesem Fall muss das Image eventuell mit einer passenden Blockgröße erneut geschrieben werden. Im Folgenden sehen Sie ein Beispiel für AIX:

```
tctl -b 0 -Bn -f /dev/rmt0 read > backup-datei.datei
   dd if=backup-datei.datei of=/dev/rmt0 obs=4096 conv=sync
```

Ein Speicherauszug des Backup-Images wird in einer Datei mit dem Namen `backup-datei.datei` erstellt. Der Befehl dd schreibt einen Speicherauszug des Images unter Verwendung einer Blockgröße von 4096 Byte zurück auf das Band.

Bei dieser Methode treten jedoch Probleme auf, wenn die Images zu umfangreich sind, um einen entsprechenden Speicherauszug in einer Datei zu speichern. Eine mögliche Lösung besteht in der Verwendung des Befehls dd. Mit diesem Befehl können Speicherauszüge von Images von einer Bandeinheit auf die andere gespeichert werden. Dieses Verfahren ist möglich, solange das Image nicht mehrere Bänder umfasst. Bei der Verwendung von zwei Bandeinheiten hat der Befehl dd folgendes Format:

```
dd if=/dev/rmt1 of=/dev/rmt0 obs=4096
```

Wenn der Einsatz von zwei Bandeinheiten nicht möglich ist, können Sie den Imagespeicherauszug mit dem Befehl dd auf einer Roheinheit speichern und ihn anschließend von dieser Roheinheit auf ein Band schreiben. Das Problem bei diesem Ansatz besteht darin, dass der Befehl dd die auf der Roheinheit gespeicherte Blockanzahl verfolgen *muss*. Diese Blockanzahl muss beim Zurückversetzen des Images auf ein Band angegeben werden. Wird der Befehl dd dazu verwendet, einen Imagespeicherauszug von einer Roheinheit auf ein Band zu schreiben, wird der gesamte Inhalt der Roheinheit auf das Band übertragen. Der Befehl dd kann nicht feststellen, wie viel Speicherplatz auf der Roheinheit zum Speichern des Images verwendet wird.

Wenn Sie das Backup-Dienstprogramm verwenden, müssen Sie den oberen Grenzwert für die Blockgröße der verwendeten Bandeinheit(en) kennen. Nachfolgend sind einige Beispiele aufgeführt:

Einheit	Anhang	Blockgrößengrenzwert	Grenzwert für DB2-Puffergröße (in 4-KB-Seiten)
8 mm	scsi	131.072	32
3420	s370	65.536	16
3480	s370	61.440	15
3490	s370	61.440	15
3490E	s370	65.536	16
7332 (4 mm)[1]	scsi	262.144	64

Einheit	Anhang	Blockgrößen-grenzwert	Grenzwert für DB2-Puffergröße (in 4-KB-Seiten)
3490e	scsi	262.144	64
3590[2]	scsi	2.097.152	512
3570 (Magstar MP)		262.144	64

Anmerkung:

1. Die Einheit 7332 implementiert keinen Blockgrößengrenzwert. Bei 256 KB handelt es sich lediglich um einen vorgeschlagenen Wert. Der geltende Blockgrößengrenzwert richtet sich nach dem übergeordneten Adapter.
2. Während auf der Einheit 3590 eine Blockgröße von 2 MB unterstützt wird, können Sie auch niedrigere Werte (z. B. 256 KB) ausprobieren, sofern die hierbei erzielten Leistungen Ihren Anforderungen gerecht werden.
3. Informationen zu den Grenzwerten Ihrer Einheit können Sie der Einheitendokumentation entnehmen bzw. beim Lieferanten der Einheit einholen.

Überprüfen der Kompatibilität der Bandeinheit

Bei UNIX-, Linux- und AIX-Betriebssystemen (ausschließlich) können Sie wie folgt bestimmen, ob Ihre Bandeinheit für eine Sicherung der DB2-Datenbanken unterstützt wird:

Führen Sie als Eigner der Datenbankmanagerinstanz den Betriebssystembefehl dd zum Lesen oder Beschreiben Ihrer Bandeinheit aus. Kann der Befehl dd erfolgreich ausgeführt werden, können Sie die DB2-Datenbanken auf der Bandeinheit sichern.

Backup in benannten Pipes

Auf UNIX-basierten Systemen ist die Unterstützung für das Sichern in (und Wiederherstellen aus) lokalen benannten Pipes verfügbar.

Das Aus- und Eingabeprogramm der benannten Pipe müssen sich auf derselben Maschine befinden. Die Pipe muss in einem lokalen Dateisystem vorhanden und lokalisierbar sein. Da die benannte Pipe als lokale Einheit behandelt wird, muss nicht speziell angegeben werden, dass es sich bei dem Ziel um eine benannte Pipe handelt.

Im Folgenden sehen Sie ein Beispiel für AIX:

1. Erstellen Sie eine benannte Pipe:
   ```
   mkfifo /u/dmcinnis/meinepipe
   ```
2. Wenn dieses Backup-Image vom Dienstprogramm RESTORE verwendet werden soll, muss die Restoreoperation *vor* der Backup-Operation aufgerufen werden, sodass alle Daten erfasst werden:
   ```
   db2 restore db sample from /u/dmcinnis/mypipe into mynewdb
   ```
3. Verwenden Sie diese Pipe als Ziel für die Backup-Operation einer Datenbank:
   ```
   db2 backup db sample to /u/dmcinnis/meinepipe
   ```

Backup von partitionierten Datenbanken

Ein Backup einer Datenbank in einer Umgebung mit partitionierten Datenbanken durchzuführen kann die folgenden Schwierigkeiten bereiten: Verfolgen, ob das Backup für alle Datenbankpartitionen erfolgreich war, Verwalten mehrerer Protokolldateien und Backup-Images und Sicherstellen, dass die Protokolldateien und Backup-Images für alle Datenbankpartitionen die erforderliche Mindestrecoveryzeit umfassen, die für ein Restore der Datenbank erforderlich ist. Ein SSV-Backup (SSV = single system view, Einzelsystemsicht) ist die einfachste Möglichkeit, für eine partitionierte Datenbank ein Backup durchzuführen.

Informationen zu dieser Task

Es gibt drei Möglichkeiten, ein Backup für eine Datenbank in einer Umgebung mit partitionierten Datenbanken durchzuführen:

- Backup jeweils einer Datenbankpartition zur Zeit mit dem Befehl BACKUP DATABASE, dem Befehl BACKUP DATABASE zusammen mit der Prozedur ADMIN_CMD oder der API-Funktion db2Backup
- Verwenden des Befehls db2_all zusammen mit dem Befehl BACKUP DATABASE, um für alle von Ihnen angegebenen Datenbankpartitionen ein Backup durchzuführen
- Ausführen eines SSV-Backups, um für einige oder alle Datenbankpartitionen gleichzeitig ein Backup durchführen zu können

Ein Backup jeweils nur für eine Datenbankpartition zur Zeit auszuführen ist zeitaufwendig und fehlerträchtig. Ein Backup für alle Partitionen mit dem Befehl db2_all ist einfacher, als für alle Datenbankpartitionen einzeln ein Backup durchzuführen, da Sie generell nur einen Befehl aufrufen müssen. Wenn Sie jedoch db2_all für ein Backup einer partitionierten Datenbank verwenden, müssen Sie db2_all in manchen Fällen trotzdem mehrfach aufrufen, da für die Datenbankpartition, die den Katalog enthält, nicht gleichzeitig ein Backup mit Datenbankpartitionen ohne Katalog durchgeführt werden kann. Unabhängig davon, ob Sie jeweils nur für eine Datenbankpartition ein Backup durchführen oder db2_all verwenden, ist das Verwalten von Backup-Images, die mit einer der beiden Methoden erstellt wurden, schwierig, da die jeweilige Zeitmarke für das Backup-Image der Datenbankpartition unterschiedlich ist und das Koordinieren der Mindestrecoveryzeit über die Backup-Images der Datenbankpartition hinweg ebenfalls schwierig ist.

Aus den oben genannten Gründen wird empfohlen, ein Backup für eine Datenbank in einer Umgebung mit partitionierten Datenbanken als SSV-Backup durchzuführen.

Gehen Sie wie folgt vor, um ein Backup für einige oder alle Datenbankpartitionen einer partitionierten Datenbank gleichzeitig in Form eines SSV-Backups durchzuführen:

Vorgehensweise

1. Optional: Die Datenbank kann im Onlinestatus bleiben oder offline geschaltet werden.

   Während Sie für eine partitionierte Datenbank ein Backup durchführen, kann die Datenbank online oder offline geschaltet sein. Ist die Datenbank online, übernimmt das Backup-Dienstprogramm gemeinsam genutzte Verbindungen zu den anderen Datenbankpartitionen, sodass die Benutzeranwendungen in der Lage sind, eine Verbindung zu einer Datenbankpartition herzustellen, während für diese ein Backup durchgeführt wird.

2. Rufen Sie in der Datenbankpartition, die den Datenbankkatalog enthält, das Backup mit den geeigneten Parametern für partitionierte Datenbanken auf.
   - Sie können den Befehl BACKUP DATABASE mit dem Parameter **ON DBPARTITIONNUMS** aufrufen.
   - Sie können den Befehl BACKUP DATABASE mit dem Parameter **ON DBPARTITIONNUMS** unter Verwendung der Prozedur ADMIN_CMD aufrufen.
   - Sie können die API db2Backup mit dem Parameter **iAllNodeFlag** aufrufen.
3. Optional: Sie können die für die Recovery erforderlichen Protokolldateien in die Backup-Images einbeziehen.

   Standardmäßig sind die Protokolldateien in Backup-Images enthalten, wenn Sie ein SSV-Backup durchführen (d. h. bei Angabe des Parameters **ON DBPARTITIONNUM**). Wenn Sie nicht möchten, dass die Protokolldateien in die Backup-Images einbezogen werden, verwenden Sie beim Ausführen des Backups den Befehlsparameter **EXCLUDE LOGS**. Bei anderen Backups als SSV-Backups sind die Protokolldateien standardmäßig nicht in den Backup-Images enthalten.

   Der Abschnitt „Einschließen von Protokolldateien in ein Backup-Image" auf Seite 173 enthält weitere Informationen.
4. Optional: Löschen Sie ältere Backup-Images. Die Methode, die Sie zum Löschen älterer Backup-Images verwenden, hängt davon ab, wie Sie die Backup-Images speichern. Wenn Sie z. B. die Backup-Images auf Platte speichern, können Sie die Dateien löschen; wenn Sie die Backup-Images mit Tivoli Storage Manager speichern, können Sie das Dienstprogramm db2adutl verwenden, um die Backup-Images zu löschen. Wenn Sie DB2 Advanced Copy Services (ACS) verwenden, können Sie mit db2acsutil Momentaufnahmebackup-Objekte löschen.

Backup partitionierter Tabellen mit IBM Tivoli Space Manager Hierarchical Storage Management

Das Hierarchical Storage Manager-Clientprogramm (HSM) von Tivoli Space Manager führt eine automatische Migration in Frage kommender Dateien auf sekundäre Speichereinheiten durch, um bestimmte Mindestgrößen an freiem Speicher in lokalen Dateisystemen sicherzustellen.

Bei einer Tabellenpartitionierung werden Tabellendaten auf mehrere Speicherobjekte verteilt, die als Datenpartitionen bezeichnet werden. HSM unterstützt ein Backup einzelner Datenpartitionen auf sekundären Speichereinheiten.

Bei SMS-Tabellenbereichen stellt sich jeder Datenpartitionsbereich als eine Datei im entsprechenden Verzeichnis dar. Daher ist eine Migration einzelner Datenbereiche (Datenpartitionen) auf sekundäre Speichereinheiten sehr einfach.

Bei DMS-Tabellenbereichen wird jeder Container durch eine Datei dargestellt. In diesem Fall empfiehlt es sich, weniger häufig genutzte Datenbereiche in einem eigenen Tabellenbereich zu speichern. Verwenden Sie bei der Eingabe der Anweisung CREATE TABLE mit der Klausel EVERY auch die Klausel NO CYCLE, um sicherzustellen, dass die Anzahl der in der IN-Klausel auf Tabellenebene aufgelisteten Tabellenbereiche mit der Anzahl der Datenpartitionen identisch ist, die erstellt werden. Das folgende Beispiel veranschaulicht diese Methode:

Beispiel 1

```
CREATE TABLE t1 (c INT) IN tbsp1, tbsp2, tbsp3 NO CYCLE
  PARTITION BY RANGE(c)
  (STARTING FROM 2 ENDING AT 6 EVERY 2);
```

Aktivieren des automatischen Backups

Eine Datenbank kann durch eine Vielzahl möglicher Hard- und Softwarefehler unbrauchbar werden. Die Bereithaltung eines möglichst aktuellen Gesamtbackups Ihrer Datenbank ist ein integraler Bestandteil der Planung und Implementierung einer Strategie zur Wiederherstellung eines Systems nach einem Katastrophenfall. Verwenden Sie das automatische Datenbankbackup im Rahmen Ihrer Wiederherstellungsstrategie, um DB2 zu veranlassen, Ihre Datenbank ordnungsgemäß und regelmäßig mit BACKUP zu sichern.

Sie können das automatische Backup über die Tools der grafischen Benutzerschnittstelle, die Befehlszeilenschnittstelle oder die gespeicherte Systemprozedur AUTOMAINT_SET_POLICY konfigurieren.

- Gehen Sie wie folgt vor, um das automatische Backup mit den Tools der grafischen Benutzerschnittstelle zu konfigurieren:
  1. Öffnen Sie den Assistenten **Automatische Verwaltung konfigurieren** entweder über die Steuerzentrale, indem Sie ein Datenbankobjekt mit der rechten Maustaste anklicken, oder über die Diagnosezentrale, indem Sie die Datenbankinstanz, die Sie zum automatischen Backup konfigurieren wollen, mit der rechten Maustaste anklicken. Wählen Sie **Automatische Verwaltung konfigurieren** im Kontextmenü aus.
  2. In diesem Assistenten können Sie das automatische Backup aktivieren und ein Verwaltungsfenster zur Ausführung des BACKUP-Dienstprogramms angeben.
- Setzen Sie die folgenden Konfigurationsparameter auf ON, um das automatische Backup über die Befehlszeilenschnittstelle zu konfigurieren:
  - AUTO_MAINT
  - AUTO_DB_BACKUP
- Gehen Sie wie folgt vor, um das automatische Backup über die gespeicherte Systemprozedur AUTOMAINT_SET_POLICY zu konfigurieren:
  1. Erstellen Sie XML-Konfigurationseingaben, in denen Einzelheiten wie Backup-Datenträger angegeben werden, ob das Backup online oder offline erfolgen sollte, sowie die Häufigkeit des Backups.

     Sie können den Inhalt der Beispieldatei DB2DefaultAutoBackupPolicy.xml in das Verzeichnis SQLLIB/samples/automaintcfg kopieren und die XML so modifizieren, dass sie Ihren Konfigurationsanforderungen entspricht.
  2. Optional: Erstellen Sie eine XML-Eingabedatei, die Ihre XML-Konfigurationseingaben enthält.
  3. Rufen Sie AUTOMAINT_SET_POLICY mit den folgenden Parametern auf:
     - Verwaltungstyp: AutoBackup
     - XML-Konfigurationseingaben: Entweder ein großes Binärobjekt, das Ihren XML-Konfigurationseingabetext enthält oder den Namen der Datei, die Ihre XML-Konfigurationseingaben enthält.

  Weitere Informationen zur Verwendung der gespeicherten Systemprozedur AUTOMAINT_SET_POLICY finden Sie im Abschnitt „Konfigurieren einer Richtlinie für automatische Verwaltung mit SYSPROC.AUTOMAINT_SET_POLICY oder SYSPROC.AUTOMAINT_SET_POLICYFILE" auf Seite 56.

Automatisches Datenbank-Backup

Eine Datenbank kann durch eine Vielzahl möglicher Hard- und Softwarefehler unbrauchbar werden. Das automatische Datenbank-Backup vereinfacht die Verwaltung von Datenbank-Backup-Tasks für den Datenbankadministrator (DBA), indem

sie fortlaufend sicherstellt, dass bei Bedarf ein neues Gesamtbackup der Datenbank durchgeführt wird. Sie stellt auf der Grundlage eines oder mehrerer der folgenden Kriterien fest, ob eine Backup-Operation ausgeführt werden muss:

- Sie haben nie ein Gesamtbackup der Datenbank durchgeführt.
- Die abgelaufene Zeit seit dem letzten Gesamtbackup ist länger als die angegebene Anzahl von Stunden.
- Der Speicherplatz für Transaktionsprotokolle seit dem letzten Backup ist größer als die angegebene Anzahl von 4-KB-Seiten (nur im Protokollarchivierungsmodus).

Schützen Sie Ihre Daten, indem Sie eine Strategie zur Recovery nach einem Katastrophenfall für Ihr System planen und implementieren. Wenn dies Ihren Anforderungen entgegenkommt, können Sie die Funktion für das automatische Datenbank-Backup in Ihre Backup- und Recoverystrategie integrieren.

Wenn die Datenbank für die aktualisierende Recovery (Protokollarchivierung) eingerichtet ist, kann das automatische Datenbank-Backup entweder für das Online- oder das Offline-Backup aktiviert werden. Ansonsten ist nur das Offline-Backup verfügbar. Das automatische Datenbank-Backup unterstützt Platten, Bänder, Tivoli Storage Manager (TSM) und DLL-Datenträgertypen von Fremdanbietern.

Über den Assistenten „Automatische Verwaltung konfigurieren" in der Steuerzentrale oder der Diagnosezentrale können Sie folgende Merkmale konfigurieren:

- Die erforderliche Zeit oder Anzahl von Protokollseiten zwischen den Backups
- Die Backup-Datenträger
- Online- oder Offline-Backup

Wenn ein Backup auf Platte ausgewählt wird, löscht die automatische Backup-Funktion die Backup-Images regelmäßig aus dem im Assistenten „Automatische Verwaltung konfigurieren" angegebenen Verzeichnis. Daher steht zu einem Zeitpunkt nur das letzte Backup-Image garantiert zur Verfügung. Es wird empfohlen, dieses Verzeichnis ausschließlich für die automatische Backup-Funktion zu reservieren und nicht zur Speicherung anderer Backup-Images zu verwenden.

Die Funktion für das automatische Datenbank-Backup kann mithilfe der Datenbankkonfigurationsparameter auto_db_backup und auto_maint aktiviert bzw. inaktiviert werden. In einer Umgebung mit partitionierten Datenbanken wird das automatische Datenbank-Backup in jeder Datenbankpartition ausgeführt, wenn die Datenbankkonfigurationsparameter für die jeweilige Datenbankpartition aktiviert sind.

Sie können das automatische Backup auch mit den gespeicherten Systemprozeduren AUTOMAINT_SET_POLICY oder AUTOMAINT_SET_POLICYFILE konfigurieren.

Optimieren der Leistung des Backup-Dienstprogramms

Wenn Sie eine Backup-Operation ausführen, wählt DB2 automatisch einen optimalen Wert für die Anzahl der Puffer, die Puffergröße und die Parallelitätseinstellungen aus. Die Werte basieren auf der Kapazität des für Dienstprogramme verfügbaren Zwischenspeichers, der Anzahl der verfügbaren Prozessoren und der Datenbankkonfiguration. Daher sollten Sie je nach der Größe der auf Ihrem System verfügbaren Speicherkapazität in Betracht ziehen, mehr Speicher zuzuordnen, indem Sie den Wert des Konfigurationsparameters UTIL_HEAP_SZ erhöhen.

Die Zielsetzung dabei ist, die zum Ausführen einer Backup-Operation erforderliche Zeit zu minimieren. Wenn Sie für die folgenden Parameter des Befehls BACKUP DATABASE nicht explizit Werte eingeben, wählt DB2 einen Wert für die Parameter aus:

- WITH anzahl-puffer BUFFERS
- PARALLELISM n
- BUFFER puffergröße

Wenn die Anzahl der Puffer und die Puffergröße nicht angegeben werden, sodass DB2 die Werte festlegt, sollte sich dies auf große Datenbanken nur minimal auswirken. Bei kleinen Datenbanken kann dies hingegen die Backup-Images prozentual beträchtlich vergrößern. Selbst wenn der letzte Datenpuffer, der auf Platte geschrieben wird, wenig Daten enthält, wird trotzdem der vollständige Puffer in das Image geschrieben. In einer kleinen Datenbank bedeutet dies, dass ein beträchtlicher Prozentsatz der Image-Größe möglicherweise leer gelassen wird.

Sie können auch eine der folgenden Aktionen ausführen, um die für eine Backup-Operation erforderliche Zeit zu reduzieren:

- Geben Sie ein Tabellenbereichsbackup an.

  Sie können einen Teil einer Datenbank sichern (und anschließend wiederherstellen), indem Sie die Option TABLESPACE des Befehls BACKUP DATABASE verwenden. Dies vereinfacht die Verwaltung von Tabellendaten, Indizes und Langfeld- bzw. LOB-Daten in separaten Tabellenbereichen.
- Erhöhen Sie den Wert des Parameters PARALLELISM des Befehls BACKUP DATABASE, bis er der Anzahl der zu sichernden Tabellenbereiche entspricht.

  Der Parameter PARALLELISM definiert die Anzahl der Prozesse bzw. Threads, die zum Lesen von Daten aus der Datenbank und zum Komprimieren von Daten während einer Operation für komprimiertes Backup gestartet werden. Jeder Prozess oder Thread wird einem bestimmten Tabellenbereich zugeordnet, d. h. es besteht kein erzielbarer Nutzen, wenn Sie für den Parameter PARALLELISM einen Wert angeben, der die Anzahl der Tabellenbereiche übersteigt, für die ein Backup durchgeführt wird. Nach dem Backup eines Tabellenbereichs wird ein anderer Bereich angefordert. Beachten Sie jedoch, dass für jeden Prozess bzw. Thread sowohl Hauptspeicher- als auch CPU-Systemaufwand anfallen.
- Erhöhen Sie die Backup-Puffergröße.

  Die ideale Puffergröße für ein Backup ist ein Vielfaches der Speicherbereichsgröße (EXTENTSIZE) des Tabellenbereichs plus eine Seite. Wenn Sie mehrere Tabellenbereiche mit verschiedenen Angaben für die Speicherbereichsgröße haben, geben Sie als Wert ein Vielfaches des Werts für die Speicherbereichsgrößen plus eine Seite an.
- Erhöhen Sie die Anzahl der Puffer.

  Verwenden Sie mindestens doppelt so viele Puffer wie Backup-Ziele (oder Sitzungen), um sicherzustellen, dass die Backup-Zieleinheiten nicht auf Daten warten müssen.
- Verwenden Sie mehrere Zieleinheiten.

Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von Backup

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf diese zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie übergeordnete Pflege- und Dienstprogrammoperationen des Datenbankmanagers zusammenzufassen. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf die Objekte zugreifen, für die sie zugriffsberechtigt sind, d. h. für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung SYSADM, SYSCTRL oder SYSMAINT, um das Backup-Dienstprogramm verwenden zu können.

Kompatibilität von Online-Backup und anderen Dienstprogrammen

Einige Dienstprogramme können zur gleichen Zeit wie ein Online-Backup ausgeführt werden, während dies bei anderen nicht möglich ist.

Die folgenden Dienstprogramme sind mit einem Online-Backup kompatibel:

- EXPORT
- ONLINE INSPECT

Die folgenden Dienstprogramme sind mit einem Online-Backup nur unter bestimmten Umständen kompatibel:

- ONLINE CREATE INDEX

  Im SMS-Modus sind eine Online-Indexerstellung und ein Online-Backup wegen der ALTER TABLE-Sperre nicht kompatibel. Die Online-Indexerstellung aktiviert diese Sperre im Exklusivmodus, während das Online-Backup sie im SHARE-Modus aktiviert.

  Im DMS-Modus können eine Online-Indexerstellung und ein Online-Backup in den meisten Fällen gleichzeitig ausgeführt werden. Es besteht jedoch die Möglichkeit, dass, wenn eine große Anzahl von Indizes vorhanden ist, die Online-Indexerstellung intern eine Online-Backup-Sperre aktiviert, die mit jedem gleichzeitig ausgeführten Online-Backup in Konflikt gerät.
- ONLINE INDEX REORG

  Ebenso wie eine Online-Indexerstellung ist eine Online-Indexreorganisation im SMS-Modus wegen der ALTER TABLE-Sperre nicht mit einem Online-Backup kompatibel. Die Online-Indexreorganisation aktiviert diese Sperre im Exklusivmodus, während das Online-Backup sie im SHARE-Modus aktiviert. Darüber hinaus führt eine Operation zur Online-Indexreorganisation vor der Umschaltphase ein Quiesce für die Tabelle aus und aktiviert eine Sperre des Modus Z, die ein Online-Backup verhindert. Allerdings sollte die ALTER TABLE-Sperre die gleichzeitige Ausführung eines Online-Backups verhindern, bevor die Tabellensperre des Modus Z angefordert wurde.

  Im DMS-Modus können eine Online-Indexreorganisation und ein Online-Backup gleichzeitig ausgeführt werden.

  Darüber hinaus führt eine Online-Indexreorganisation vor der Umschaltphase ein Quiesce für die Tabelle aus und aktiviert eine Sperre des Modus Z, die ein Online-Backup verhindert.

- REBALANCE

  Wenn ein Online-Backup und die Neuausgleichsfunktion gleichzeitig ausgeführt werden, hält das Online-Backup die Neuausgleichsfunktion an, ohne zu warten, bis sie abgeschlossen ist.
- IMPORT

  Das Dienstprogramm IMPORT ist mit einem Online-Backup kompatibel, sofern der Befehl IMPORT nicht mit der Option REPLACE ausgeführt wird. In diesem Fall aktiviert das Dienstprogramm IMPORT eine Sperre des Modus Z für die Tabelle und verhindert die gleichzeitige Ausführung eines Online-Backups.
- ALLOW READ ACCESS LOAD

  Ladeoperationen mit der Option ALLOW READ ACCESS sind nicht mit einem Online-Backup kompatibel, wenn der Befehl LOAD mit der Option COPY NO ausgeführt wird. In diesem Modus ändern beide Dienstprogramme den Tabellenbereichsstatus, sodass eines der Dienstprogramme einen Fehler meldet.

  Ladeoperationen mit der Option ALLOW READ ACCESS sind mit einem Online-Backup kompatibel, wenn der Befehl LOAD mit der Option COPY YES ausgeführt wird, obwohl es hinsichtlich der Kompatibilität noch einige Aspekte geben kann, die zu beachten sind. Im SMS-Modus können die Dienstprogramme gleichzeitig ausgeführt werden. Sie arbeiten jedoch mit inkompatiblen Tabellensperrmodi, sodass es eventuell zu Wartezeiten für Tabellensperren kommt. Im DMS-Modus arbeiten beide Dienstprogramme mit inkompatiblen Sperrmodi "Internal-B" (OLB), sodass es zu Wartezeiten für Sperren dieses Modus kommen kann. Wenn die Dienstprogramme gleichzeitig für denselben Tabellenbereich ausgeführt werden, wird das Dienstprogramm LOAD möglicherweise gezwungen, zu warten, bis das Backup-Dienstprogramm die Verarbeitung des Tabellenbereichs beendet hat, bevor das Dienstprogramm LOAD fortfahren kann.
- ONLINE TABLE REORG

  Die Bereinigungsphase der Online-Tabellenreorganisation kann nicht starten, solange ein Online-Backup ausgeführt wird. Sie können die Tabellenreorganisation, falls erforderlich, anhalten, sodass das Online-Backup abgeschlossen werden kann, bevor die Online-Tabellenreorganisation wieder aufgenommen wird.

  Sie können ein Online-Backup eines DMS-Tabellenbereichs starten, während eine Tabelle im gleichen Tabellenbereich online reorganisiert wird. Während der Abschneidephase kann es im Zusammenhang mit der Reorganisationsoperation jedoch zu Sperrenwartezeiten kommen.

  Sie können ein Online-Backup eines SMS-Tabellenbereichs nicht starten, während eine Tabelle im gleichen Tabellenbereich online reorganisiert wird. Beide Operationen erfordern eine exklusive Sperre.
- DDL-Anweisungen, die eine Sperre im Modus Z erfordern (wie ALTER TABLE, DROP TABLE und DROP INDEX)

  Ein Online-Backup eines DMS-Tabellenbereichs ist mit DDL-Anweisungen kompatibel, die eine Sperre im Modus Z erfordern.

  Ein Online-Backup eines SMS-Tabellenbereichs muss warten, bis die Sperre des Modus Z freigegeben wird.
- RUNSTATS (mit Option ALLOW WRITE und ALLOW READ)

  Der Befehl RUNSTATS ist mit dem Online-Backup kompatibel, es sei denn, der Systemkatalogtabellenbereich ist ein SMS-Tabellenbereich. Wenn sich der Systemkatalog in einem SMS-Tabellenbereich befindet, halten der Befehl RUNSTATS und das Online-Backup nicht kompatible Sperren für die Tabelle, die den Wartestatus für Sperren zur Folge haben.

Die folgenden Dienstprogramme sind mit einem Online-Backup nicht kompatibel:

- REORG TABLE
- RESTORE
- ROLLFORWARD
- ONLINE BACKUP
- ALLOW NO ACCESS LOAD
- SET WRITE

Backup - Beispiele

Beispiel 1

Im folgenden Beispiel wird die Datenbank SAMPLE in zwei gleichzeitig ablaufenden TSM-Clientsitzungen auf einem TSM-Server gesichert. Das Backup-Dienstprogramm berechnet die optimale Anzahl von Puffern. Die optimale Puffergröße in 4-KB-Seiten wird automatisch auf Basis der verfügbaren Speichermenge und der Anzahl verfügbarer Zieleinheiten berechnet. Die Parallelitätseinstellung wird ebenfalls automatisch berechnet und basiert auf der Anzahl der verfügbaren Prozessoren und der Anzahl der zu sichernden Tabellenbereiche.

```
db2 backup database sample use tsm open 2 sessions with 4 buffers

db2 backup database payroll tablespace (syscatspace, userspace1) to
/dev/rmt0, /dev/rmt1 with 8 buffers without prompting
```

Beispiel 2

Im Folgenden sehen Sie ein Beispiel für eine wöchentliche inkrementelle Backup-Strategie für eine wiederherstellbare Datenbank. Diese Strategie umfasst ein wöchentliches Gesamtbackup der Datenbank, ein tägliches nicht kumulatives Backup (Delta-Backup) sowie ein kumulatives Backup (inkrementell) in der Wochenmitte:

```
(So) db2 backup db kdr use tsm
(Mo) db2 backup db kdr online incremental delta use tsm
(Di) db2 backup db kdr online incremental delta use tsm
(Mi) db2 backup db kdr online incremental use tsm
(Do) db2 backup db kdr online incremental delta use tsm
(Fr) db2 backup db kdr online incremental delta use tsm
(Sa) db2 backup db kdr online incremental use tsm
```

Beispiel 3

Setzen Sie den folgenden Befehl ab, um in einer Windows-Umgebung eine Backup-Operation für eine Bandeinheit einzuleiten:

```
db2 backup database sample to \\.\tape0
```

Kapitel 11. Recovery - Übersicht

Das Recoverydienstprogramm führt auf Basis der in der Datei des Recoveryprotokolls vorhandenen Informationen die erforderlichen Operationen zum Wiederherstellen und aktualisierenden Wiederherstellen einer Datenbank bis zu einem bestimmten Zeitpunkt aus. Wenn Sie dieses Dienstprogramm verwenden, geben Sie an, ob die Datenbank bis zu einem bestimmten Zeitpunkt oder bis zum Ende der Protokolldateien wiederhergestellt werden soll. Das Dienstprogramm wählt anschließend das am besten geeignete Backup-Image aus und führt die Recoveryoperationen aus.

Das Recoverydienstprogramm bietet keine Unterstützung für die folgenden Optionen des Befehls RESTORE DATABASE:

- TABLESPACE tabellenbereichsname. Restoreoperationen für Tabellenbereiche werden nicht unterstützt.
- INCREMENTAL. Operationen zum inkrementellen Restore werden nicht unterstützt.
- OPEN anzahl-sitzungen SESSIONS. Die Anzahl E/A-Sitzungen, die mit TSM oder einem Programm eines anderen Lieferanten verwendet werden sollen, kann nicht angegeben werden.
- BUFFER puffergröße. Die Größe des für die Restoreoperation verwendeten Puffers kann nicht angegeben werden.
- DLREPORT dateiname. Es kann kein Dateiname angegeben werden für Berichtsdateien, deren Verknüpfung aufgehoben wurde.
- WITHOUT ROLLING FORWARD. Es kann nicht angegeben werden, dass die Datenbank nicht in den Status *Aktualisierende Recovery anstehend* versetzt werden soll, nachdem sie erfolgreich wiederhergestellt wurde.
- PARALLELISM n. Es kann kein Grad der Parallelität für die Restoreoperation angegeben werden.
- WITHOUT PROMPTING. Es kann nicht angegeben werden, dass die Restoreoperation nicht überwacht ausgeführt werden soll.

Darüber hinaus lässt das Recoverydienstprogramm nicht zu, dass Sie eine der REBUILD-Optionen angeben. Das Recoverydienstprogramm verwendet jedoch automatisch die geeignete REBUILD-Option, wenn es auf der Basis der Informationen in der Datei des Recoveryprotokolls keine Datenbank-Backup-Images lokalisieren kann.

Recovery von Daten

Verwenden Sie den Befehl RECOVER DATABASE, um eine Recovery einer Datenbank bis zu einem angegebenen Zeitpunkt anhand von Informationen in der Datei des Recoveryprotokolls durchzuführen.

Wenn Sie den Befehl RECOVER DATABASE nach einer unvollständigen Recoveryoperation ausführen, die in der Phase der aktualisierenden Recovery (Rollforward) beendet wurde, versucht das Recoverydienstprogramm, die vorige Recoveryoperation fortzusetzen, ohne die Restorephase zu wiederholen. Wenn Sie das Recoverydienstprogramm zwingen wollen, die Restorephase zu wiederholen, geben Sie den Befehl RECOVER DATABASE mit der Option RESTART ein. Dies veranlasst das Recoverydienstprogramm, jede frühere Recoveryoperation, die nicht

vollständig abgeschlossen wurde, zu ignorieren. Wenn Sie die Anwendungsprogrammierschnittstelle (API) verwenden, geben Sie die Aufruferaktion DB2RECOVER_RESTART für das Feld 'iRecoverAction' an, um das Recoverydienstprogramm zu veranlassen, die Restorephase zu wiederholen.

Wenn der Befehl RECOVER DATABASE während der Restorephase unterbrochen wird, kann er nicht fortgesetzt werden. Sie müssen den Befehl RECOVER DATABASE erneut absetzen.

Es sollte noch keine Verbindung zu der wiederherzustellenden Datenbank bestehen: Das Dienstprogramm für die Datenbankrecovery stellt automatisch eine Verbindung zu der angegebenen Datenbank her. Diese Verbindung wird nach Abschluss der Recoveryoperation beendet.

Bei der Datenbank kann es sich um eine lokale oder ferne Datenbank handeln.

Sie können das Recoverydienstprogramm über den Befehlszeilenprozessor (CLP) oder die Anwendungsprogrammierschnittstelle (API) db2Recover aufrufen.

Das folgende Beispiel zeigt, wie Sie den Befehl RECOVER DATABASE über den CLP verwenden:

```
db2 recover db sample
```

Anmerkung: In einer Umgebung mit partitionierten Datenbanken muss das Recoverydienstprogramm von der Katalogpartition der Datenbank aus aufgerufen werden.

Datenrecovery mit 'db2adutl'

Die folgenden Beispiele zeigen, wie eine knotenübergreifende Fehlerbehebung mit dem Befehl db2adutl sowie mithilfe der Datenbankkonfigurationsparameter *logarchopt1* und *vendoropt* in verschiedenen TSM-Umgebungen (TSM = Tivoli Storage Manager) ausgeführt wird.

In den folgenden Beispielen besitzt Computer 1 den Namen `bar` und wird unter dem Betriebssystem AIX betrieben. Der Benutzer dieser Maschine ist `roecken`. Die Datenbank auf der Maschine `bar` heißt `zample`. Computer 2 besitzt den Namen `dps`. Dieser Computer wird ebenfalls unter dem Betriebssystem AIX von dem Benutzer `regress9` ausgeführt.

Beispiel 1: Automatische Kennwortverwaltung durch den TSM-Server (die Option PASSWORDACCESS ist auf GENERATE gesetzt)

Dieses Beispiel für knotenübergreifende Recovery zeigt die Vorgehensweise zum Einrichten von zwei Computern für die gegenseitige Datenrecovery, wenn die Protokollarchive und Backups auf einem TSM-Server gespeichert sind und die Kennwörter mit der Option PASSWORDACCESS=GENERATE verwaltet werden.

Anmerkung: Nach dem Aktualisieren der Datenbankkonfiguration müssen Sie möglicherweise ein Offline-Backup der Datenbank erstellen.

1. Um die Datenbank für Protokollarchivierung für den Computer `bar` auf dem TSM-Server zu aktivieren, aktualisieren Sie den Datenbankkonfigurationsparameter *logarchmeth1* für die Datenbank `zample` mit dem folgenden Befehl:

   ```
   bar:/home/roecken> db2 update db cfg for zample using LOGARCHMETH1 tsm
   ```

   Die folgenden Informationen werden zurückgegeben:

```
DB20000I  Der Befehl UPDATE DATABASE CONFIGURATION wurde erfolgreich ausgeführt.
```

2. Trennen Sie mit dem folgenden Befehl alle Benutzer und Anwendungen von der Datenbank:

   ```
   db2 force applications all
   ```

3. Stellen Sie mit dem folgenden Befehl sicher, dass keine Anwendungen mit der Datenbank verbunden sind:

   ```
   db2 list applications
   ```

   Sie sollten eine Nachricht empfangen, die besagt, dass keine Daten zurückgegeben wurden.

   Anmerkung: In einer Umgebung mit partitionierten Datenbanken müssen Sie diesen Schritt in allen Datenbankpartitionen ausführen.

4. Erstellen Sie mit dem folgenden Befehl ein Backup der Datenbank auf dem TSM-Server:

   ```
   db2 backup db zample use tsm
   ```

   Informationen ähnlich den folgenden werden zurückgegeben:

   ```
   Das Backup war erfolgreich. Die Zeitmarke für dieses Backup-Image ist: 20090216151025
   ```

   Anmerkung: In einer Umgebung mit partitionierten Datenbanken müssen Sie diesen Schritt in allen Datenbankpartitionen ausführen. Die Reihenfolge, in der Sie diesen Schritt an den Datenbankpartitionen ausführen, variiert, je nachdem, ob Sie ein Online-Backup oder ein Offline-Backup durchführen. Weitere Informationen hierzu finden Sie in „Verwenden von Backup" auf Seite 252.

5. Stellen Sie mit dem folgenden Befehl die Verbindung zur Datenbank `zample` her:

   ```
   db2 connect to zample
   ```

6. Generieren Sie mit dem folgenden Befehl neue Transaktionsprotokolle für die Datenbank durch Erstellen einer Tabelle und Laden von Daten in den TSM-Server:

   ```
   bar:/home/roecken> db2 load from mr of del modified by noheader replace
into employee copy yes use tsm
   ```

   In diesem Beispiel hat die Tabelle den Namen `employee`, und die Daten werden aus einer Datei im ASCII-Format mit Begrenzern geladen, die den Namen `mr` hat. Die Option COPY YES wird angegeben, um eine Kopie der geladenen Daten zu erstellen, und durch die Option USE TSM wird angegeben, dass die Kopie der Daten auf dem TSM-Server zu speichern ist.

   Anmerkung: Sie können die Option COPY YES nur angeben, wenn die Datenbank für die aktualisierende Recovery konfiguriert ist. Das heißt, der Datenbankkonfigurationsparameter *logarchmeth1* muss auf den Wert USEREXIT, LOGRETAIN, DISK oder TSM gesetzt sein.

   Das Dienstprogramm gibt eine Reihe von Nachrichten zurück, um den Verarbeitungsfortschritt anzuzeigen:

   ```
   SQL3109N  Das Dienstprogramm beginnt mit dem Laden von Daten aus der Datei
   "/home/roecken/mr".

   SQL3500W  Die Phase "LOAD" wird gestartet (Zeit: "02/16/2009
   15:12:13.392633").

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

   SQL3520W  Synchronisationspunkt für Ladevorgang erfolgreich.

   SQL3110N  Die Verarbeitung des Dienstprogramms ist abgeschlossen. Es wurde(n) "1" Zeile(n)
   ```

```
aus der Eingabedatei gelesen.

SQL3519W  Synchronisationspunkt am Beginn des Ladevorgangs. Eingabesatzzähler = "1".

SQL3520W  Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3515W  Die Phase "LOAD" wurde beendet (Zeit: "02/16/2009 15:12:13.445718").

   Anzahl gelesener Zeilen          = 1
   Anzahl übersprungener Zeilen      = 0
   Anzahl geladener Zeilen          = 1
   Anzahl zurückgewiesener Zeilen    = 0
   Anzahl gelöschter Zeilen          = 0
   Anzahl festgeschriebener Zeilen  = 1
```

7. Vergewissern Sie sich nach dem Laden der Daten in die Tabelle, dass auf dem TSM-Server ein Backup-Image, ein Ladekopieimage und eine Protokolldatei vorhanden ist, indem Sie die folgende Abfrage für die Datenbank `zample` ausführen:

   ```
   bar:/home/roecken/sqllib/adsm>

   db2adutl query db zample
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
   Retrieving FULL DATABASE BACKUP information.
    1 Time: 20090216151025  Oldest log: S0000000.LOG  DB Partition Number: 0
           Sessions: 1

   Retrieving INCREMENTAL DATABASE BACKUP information.
       No INCREMENTAL DATABASE BACKUP images found for ZAMPLE

   Retrieving DELTA DATABASE BACKUP information.
       No DELTA DATABASE BACKUP images found for ZAMPLE

   Retrieving TABLESPACE BACKUP information.
       No TABLESPACE BACKUP images found for ZAMPLE

   Retrieving INCREMENTAL TABLESPACE BACKUP information.
       No INCREMENTAL TABLESPACE BACKUP images found for ZAMPLE

   Retrieving DELTA TABLESPACE BACKUP information.
       No DELTA TABLESPACE BACKUP images found for ZAMPLE

   Retrieving LOAD COPY information.
    1 Time: 20090216151213

   Retrieving LOG ARCHIVE information.
         Log file: S0000000.LOG, Chain Num: 0, DB Partition Number: 0,
   Taken at: 2009-02-16-15.10.38
   ```

8. Um die knotenübergreifende Recovery zu aktivieren, müssen Sie den Zugriff auf die zugeordneten Objekte des Computers `bar` für einen anderen Computer und ein anderes Benutzerkonto erteilen. Erteilen Sie in diesem Beispiel den Zugriff für den Computer `dps` und den Benutzer `regress9` mit dem folgenden Befehl:

   ```
   bar:/home/roecken/sqllib/adsm> db2adutl grant user regress9
            on nodename dps for db zample
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
   Successfully added permissions for regress9 to access ZAMPLE on node dps.
   ```

   Anmerkung: Sie können die Ergebnisse der GRANT-Operation für db2adutl überprüfen, indem Sie mit dem folgenden Befehl die aktuelle Zugriffsliste für den aktuellen Knoten abrufen:

   ```
   bar:/home/roecken/sqllib/adsm> db2adutl queryaccess
   ```

   Die folgenden Informationen werden zurückgegeben:

```
Node                 Username             Database Name   Type
-------------------------------------------------------------
DPS                  regress9             ZAMPLE          A
-------------------------------------------------------------
 Access Types:  B - backup images   L - logs   A - both
```

9. In diesem Beispiel ist der zweite Computer (dps) noch nicht für die knotenübergreifende Recovery der Datenbank zample konfiguriert. Vergewissern Sie sich mit dem folgenden Befehl, dass diesem Benutzer und diesem Computer keine Daten auf dem TSM-Server zugeordnet sind:

   ```
   dps:/home/regress9/sqllib/adsm> db2adutl query db zample
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
     --- Database directory is empty ---
   Warning: There are no file spaces created by DB2 on the ADSM server
   Warning: No DB2 backup images found in ADSM for any alias.
   ```

10. Rufen Sie mit dem folgenden Befehl vom TSM-Server eine Liste der Objekte für die Datenbank zample ab, die dem Benutzer roecken und dem Computer bar zugeordnet sind:

    ```
    dps:/home/regress9/sqllib/adsm> db2adutl query db zample nodename
                bar owner roecken
    ```

    Die folgenden Informationen werden zurückgegeben:

    ```
    --- Database directory is empty ---

  Query for database ZAMPLE

    Retrieving FULL DATABASE BACKUP information.
     1 Time: 20090216151025  Oldest log: S0000000.LOG  DB Partition Number: 0
            Sessions: 1

    Retrieving INCREMENTAL DATABASE BACKUP information.
        No INCREMENTAL DATABASE BACKUP images found for ZAMPLE

    Retrieving DELTA DATABASE BACKUP information.
        No DELTA DATABASE BACKUP images found for ZAMPLE

    Retrieving TABLESPACE BACKUP information.
        No TABLESPACE BACKUP images found for ZAMPLE

    Retrieving INCREMENTAL TABLESPACE BACKUP information.
        No INCREMENTAL TABLESPACE BACKUP images found for ZAMPLE

    Retrieving DELTA TABLESPACE BACKUP information.
        No DELTA TABLESPACE BACKUP images found for ZAMPLE

    Retrieving LOAD COPY information.
     1 Time: 20090216151213

    Retrieving LOG ARCHIVE information.
          Log file: S0000000.LOG, Chain Num: 0, DB Partition Number: 0,
    Taken at: 2009-02-16-15.10.38
    ```

    Diese Informationen stimmen mit den zuvor generierten TSM-Informationen überein und bestätigen damit, dass dieses Image auf dem Computer dps wiederhergestellt werden kann.

11. Stellen Sie mit dem folgenden Befehl die Datenbank zample vom TSM-Server auf dem Computer dps wieder her:

    ```
    dps:/home/regress9> db2 restore db zample use tsm options
       "'-fromnode=bar -fromowner=roecken'" without prompting
    ```

    Die folgenden Informationen werden zurückgegeben:

    ```
    DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.
    ```

    Anmerkung: Wenn die Datenbank zample auf dps bereits vorhanden wäre, würde der Parameter OPTIONS weggelassen, und der Datenbankkonfigurati-

onsparameter *vendoropt* würde verwendet. Dieser Konfigurationsparameter überschreibt den Parameter OPTIONS für eine BACKUP- oder RESTORE-Operation.

12. Führen Sie eine aktualisierende Wiederherstellung durch, um die Transaktionen anzuwenden, die beim Erstellen einer neuen Tabelle und beim Laden neuer Daten in der Protokolldatei für die Datenbank zample erfasst wurden. In diesem Beispiel schlägt die aktualisierende Wiederherstellung (ROLLFORWARD) in der Datenbank fehl, weil das Dienstprogramm für aktualisierende Wiederherstellung die Protokolldateien nicht finden kann, da keine Benutzer- und Computerinformationen angegeben sind:

    ```
    dps:/home/regress9> db2 rollforward db zample to end of logs and stop
    ```

    Der Befehl gibt den folgenden Fehler zurück:

    ```
    SQL4970N  Die aktualisierende Recovery der Datenbank "ZAMPLE"
    kann wegen fehlender Protokolldatei(en) auf Knoten "0" nicht den angegebenen
    Endpunkt (Protokollende oder angegebener Zeitpunkt) erreichen.
    ```

    Veranlassen Sie das Dienstprogramm für aktualisierende Wiederherstellung, die Protokolldateien für einen anderen Computer zu suchen, indem Sie den entsprechenden Wert für *logarchopt* angeben. Verwenden Sie in diesem Beispiel den folgenden Befehl, um den Datenbankkonfigurationsparameter *logarchopt1* festzulegen und nach Protokolldateien für den Benutzer roecken und den Computer bar zu suchen:

    ```
    dps:/home/regress9> db2 update db cfg for zample using logarchopt1
       "'-fromnode=bar -fromowner=roecken'"
    ```

13. Ermöglichen Sie mit dem folgenden Befehl dem Dienstprogramm für aktualisierende Wiederherstellung die Verwendung der Backup- und Ladekopieimages, indem Sie den Datenbankkonfigurationsparameter *vendoropt* festlegen:

    ```
    dps:/home/regress9> db2 update db cfg for zample using VENDOROPT
       "'-fromnode=bar -fromowner=roecken'"
    ```

14. Mit dem folgenden Befehl können Sie die knotenübergreifende Datenrecovery beenden, indem die in der Protokolldatei der Datenbank zample erfassten Transaktionen angewendet werden:

    ```
    dps:/home/regress9> db2 rollforward db zample to end of logs and stop
    ```

    Die folgenden Informationen werden zurückgegeben:

    ```
                                  Status der aktualisierenden Recovery

        Aliasname der Eingabedatenbank              = zample
        Anzahl der Knoten, die Status zurückgaben   = 1

        Knotennummer                                = 0
        Aktualisierende Recovery: Status            = nicht anstehend
        Nächste zu lesende Protokolldatei           =
        Verarbeitete Protokolldateien               = S0000000.LOG - S0000000.LOG
    Letzte festgeschriebene Transaktion             = 2009-02-16-20.10.38.000000 UTC

      DB20000I  Der Befehl ROLLFORWARD wurde erfolgreich ausgeführt.
    ```

    Die Datenbank zample auf dem Computer dps unter dem Benutzer regress9 wurde auf denselben Stand wie die Datenbank auf dem Computer bar unter dem Benutzer roecken wiederhergestellt.

Beispiel 2: Kennwortverwaltung durch den Benutzer (die Option PASSWORDACCESS ist auf PROMPT gesetzt)

Dieses Beispiel für knotenübergreifende Recovery zeigt die Vorgehensweise zum Einrichten von zwei Computern für die gegenseitige Datenrecovery, wenn die Protokollarchive und Backups auf einem TSM-Server gespeichert sind und die Kennwörter von den Benutzern verwaltet werden. In diesen Umgebungen sind zusätzli-

che Informationen erforderlich. Dies sind insbesondere der TSM-Knotenname und das Kennwort für den Computer, auf dem die Objekte erstellt wurden.

1. Fügen Sie in der Datei `dsm.sys` die folgende Zeile hinzu, weil der Quellencomputer den Namen `bar` trägt:

   ```
   NODENAME bar
   ```

   Anmerkung: Auf Windows-Betriebssystemen heißt diese Datei *dsm.opt*. Nach dem Aktualisieren dieser Datei müssen Sie für Ihr System einen Warmstart durchführen, damit die Änderungen wirksam werden.

2. Rufen Sie mit dem folgenden Befehl vom TSM-Server eine Liste der Objekte ab, die dem Benutzer `roecken` und dem Computer `bar` zugeordnet sind:

   ```
   dps:/home/regress9/sqllib/adsm> db2adutl query db zample nodename bar
      owner roecken password *******
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
   Query for database ZAMPLE

      Retrieving FULL DATABASE BACKUP information.
       1 Time: 20090216151025  Oldest log: S0000000.LOG  DB Partition Number: 0
              Sessions: 1

      Retrieving INCREMENTAL DATABASE BACKUP information.
          No INCREMENTAL DATABASE BACKUP images found for ZAMPLE

      Retrieving DELTA DATABASE BACKUP information.
          No DELTA DATABASE BACKUP images found for ZAMPLE

      Retrieving TABLESPACE BACKUP information.
          No TABLESPACE BACKUP images found for ZAMPLE

      Retrieving INCREMENTAL TABLESPACE BACKUP information.
          No INCREMENTAL TABLESPACE BACKUP images found for ZAMPLE

      Retrieving DELTA TABLESPACE BACKUP information.
          No DELTA TABLESPACE BACKUP images found for ZAMPLE

      Retrieving LOAD COPY information.
       1 Time: 20090216151213

      Retrieving LOG ARCHIVE information.
            Log file: S0000000.LOG, Chain Num: 0, DB Partition Number: 0,
      Taken at: 2009-02-16-15.10.38
   ```

3. Führen Sie die folgenden Schritte aus, wenn die Datenbank `zample` auf dem Computer `dps` noch nicht vorhanden ist:

   a. Erstellen Sie mit dem folgenden Befehl eine leere Datenbank `zample`:

      ```
      dps:/home/regress9> db2 create db zample
      ```

   b. Aktualisieren Sie den Datenbankkonfigurationsparameter *tsm_nodename* mit dem folgenden Befehl:

      ```
      dps:/home/regress9> db2 update db cfg for zample using tsm_nodename bar
      ```

   c. Aktualisieren Sie den Datenbankkonfigurationsparameter *tsm_password* mit dem folgenden Befehl:

      ```
      dps:/home/regress9> db2 update db cfg for zample using
         tsm_password ********
      ```

4. Versuchen Sie mit dem folgenden Befehl, die Datenbank `zample` wiederherzustellen:

   ```
   dps:/home/regress9> db2 restore db zample use tsm options
      "'-fromnode=bar -fromowner=roecken'" without prompting
   ```

   Die Restoreoperation wird erfolgreich ausgeführt, jedoch wird eine Warnung ausgeben:

   ```
      SQL2540W  RESTORE wurde erfolgreich ausgeführt, es wurde jedoch eine
   Warnung "2523" beim Restore der Datenbank im Modus
   'No Interrupt' festgestellt.
   ```

5. Führen Sie mit dem folgenden Befehl eine aktualisierende Recovery durch:

   ```
   dps:/home/regress9> db2 rollforward db zample to end of logs and stop
   ```

   Weil in diesem Beispiel die Datenbankkonfigurationsdatei durch die Restore-operation ersetzt wurde, kann das Dienstprogramm zur aktualisierenden Recovery nicht die richtigen Protokolldateien finden und die folgende Fehlernachricht wird zurückgegeben:

   ```
   SQL1268N  Die aktualisierende Recovery wurde nach dem
   Fehler "-2112880618" beendet, während die Protokolldatei "S0000000.LOG"
   für die Datenbank "ZAMPLE" auf dem Knoten "0" abgerufen wurde.
   ```

   Setzen Sie die folgenden TSM-Datenbankkonfigurationsparameter auf die richtigen Werte zurück:

   a. Legen Sie mit dem folgenden Befehl den Konfigurationsparameter *tsm_nodename* fest:

      ```
      dps:/home/regress9> db2 update db cfg for zample using tsm_nodename bar
      ```

   b. Legen Sie mit dem folgenden Befehl den Datenbankkonfigurationsparameter *tsm_password* fest:

      ```
      dps:/home/regress9> db2 update db cfg for zample using tsm_password *******
      ```

   c. Legen Sie den Datenbankkonfigurationsparameter *logarchopt1* so fest, dass vom Dienstprogramm für aktualisierende Recovery die richtigen Protokolldateien gefunden werden:

      ```
      dps:/home/regress9> db2 update db cfg for zample using logarchopt1
         "'-fromnode=bar -fromowner=roecken'"
      ```

   d. Legen Sie mit dem folgenden Befehl den Datenbankkonfigurationsparameter *vendoropt* so fest, dass die Recoverydatei auch während der aktualisierenden Recovery verwendet werden kann:

      ```
      dps:/home/regress9> db2 update db cfg for zample using VENDOROPT
         "'-fromnode=bar -fromowner=roecken'"
      ```

6. Mit dem folgenden Befehl können Sie die knotenübergreifende Recovery durch das Ausführen der aktualisierenden Recovery beenden:

   ```
   dps:/home/regress9> db2 rollforward db zample to end of logs and stop
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
                                  Status der aktualisierenden Recovery

       Aliasname der Eingabedatenbank             = zample
       Anzahl der Knoten, die Status zurückgaben  = 1

       Knotennummer                               = 0
       Aktualisierende Recovery: Status           = nicht anstehend
       Nächste zu lesende Protokolldatei          =
       Verarbeitete Protokolldateien              = S0000000.LOG - S0000000.LOG
   Letzte festgeschriebene Transaktion            = 2009-02-16-20.10.38.000000 UTC

     DB20000I  Der Befehl ROLLFORWARD wurde erfolgreich ausgeführt.
   ```

Die Datenbank `zample` auf dem Computer `dps` unter dem Benutzer `regress9` wurde auf denselben Stand wie die Datenbank auf dem Computer `bar` unter dem Benutzer `roecken` wiederhergestellt.

Beispiel 3: Konfigurieren des TSM-Servers für die Verwendung von Client-Proxy-Knoten

Dieses Beispiel für die knotenübergreifende Recovery zeigt die Vorgehensweise zum Einrichten von zwei Computern als Proxy-Knoten für die gegenseitige Datenrecovery, wenn die Protokollarchive und Backups auf einem TSM-Server gespeichert sind und die Kennwörter mit der Option PASSWORDACCESS=GENERATE verwaltet werden.

Wichtig: Der Parameter **OPTIONS** des Befehls db2adutl ist in Version 9.5 Fixpack 6 und späteren Fixpacks verfügbar. Zum Ausführen der unten angegebenen Schritte muss Version 9.5 Fixpack 6 (oder ein späteres Fixpack) installiert sein.

Anmerkung: Nach dem Aktualisieren der Datenbankkonfiguration müssen Sie möglicherweise ein Offline-Backup der Datenbank erstellen.

In diesem Beispiel werden die Computer `bar` und `dps` unter dem Proxy-Namen `clusternode` registriert. Die Computer sind bereits als Proxy-Knoten konfiguriert.

1. Registrieren Sie mit den folgenden Befehlen die Computer `bar` und `dps` als Proxy-Knoten auf dem TSM-Server:

   ```
   REGISTER NODE clusternode mypassword
   GRANT PROXYNODE TARGET=clusternode AGENT=bar,dps
   ```

2. Um die Datenbank für die Protokollarchivierung für den TSM-Server zu aktivieren, aktualisieren Sie den Datenbankkonfigurationsparameter *logarchmeth1* für die Datenbank `zample` mit dem folgenden Befehl:

   ```
   bar:/home/roecken> db2 update db cfg for zample using
      LOGARCHMETH1 tsm logarchopt1 "'-asnodename=clusternode'"
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
   DB20000I  Der Befehl UPDATE DATABASE CONFIGURATION wurde erfolgreich ausgeführt.
   ```

3. Trennen Sie mit dem folgenden Befehl alle Benutzer und Anwendungen von der Datenbank:

   ```
   db2 force applications all
   ```

4. Stellen Sie mit dem folgenden Befehl sicher, dass keine Anwendungen mit der Datenbank verbunden sind:

   ```
   db2 list applications
   ```

   Sie sollten eine Nachricht empfangen, die besagt, dass keine Daten zurückgegeben wurden.

   Anmerkung: In einer Umgebung mit partitionierten Datenbanken müssen Sie diesen Schritt in allen Datenbankpartitionen ausführen.

5. Erstellen Sie mit dem folgenden Befehl ein Backup der Datenbank auf dem TSM-Server:

   ```
   db2 backup db zample use tsm options "'-asnodename=clusternode'"
   ```

   Informationen ähnlich den folgenden werden zurückgegeben:

   ```
   Das Backup war erfolgreich. Die Zeitmarke für dieses Backup-Image ist: 20090216151025
   ```

   Anstatt die Option '-asnodename' im Befehl 'backup' anzugeben, können Sie den Datenbankkonfigurationsparameter *vendoropt* aktualisieren.

   Anmerkung: In einer Umgebung mit partitionierten Datenbanken müssen Sie diesen Schritt in allen Datenbankpartitionen ausführen. Die Reihenfolge, in der Sie diesen Schritt an den Datenbankpartitionen ausführen, variiert, je nachdem, ob Sie ein Online-Backup oder ein Offline-Backup durchführen. Weitere Informationen hierzu finden Sie in „Verwenden von Backup” auf Seite 252.

6. Stellen Sie mit dem folgenden Befehl die Verbindung zur Datenbank `zample` her:

   ```
   db2 connect to zample
   ```

7. Generieren Sie mit dem folgenden Befehl neue Transaktionsprotokolle für die Datenbank durch Erstellen einer Tabelle und Laden von Daten in den TSM-Server:

```
bar:/home/roecken> db2 load from mr of del modified by noheader replace
into employee copy yes use tsmwhere
```

In diesem Beispiel hat die Tabelle den Namen employee, und die Daten werden aus einer Datei im ASCII-Format mit Begrenzern geladen, die den Namen mr hat. Die Option COPY YES wird angegeben, um eine Kopie der geladenen Daten zu erstellen, und durch die Option USE TSM wird angegeben, dass die Kopie der Daten auf dem TSM-Server zu speichern ist.

Anmerkung: Sie können die Option COPY YES nur angeben, wenn die Datenbank für die aktualisierende Recovery konfiguriert ist. Das heißt, der Datenbankkonfigurationsparameter *logarchmeth1* muss auf den Wert USEREXIT, LOGRETAIN, DISK oder TSM gesetzt sein.

Das Dienstprogramm gibt eine Reihe von Nachrichten zurück, um den Verarbeitungsfortschritt anzuzeigen:

```
SQL3109N  Das Dienstprogramm beginnt mit dem Laden von Daten aus der Datei
"/home/roecken/mr".

SQL3500W  Die Phase "LOAD" wird gestartet (Zeit: "02/16/2009
15:12:13.392633").

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

SQL3520W  Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3110N  Die Verarbeitung des Dienstprogramms ist abgeschlossen. Es wurde(n) "1" Zeile(n)
aus der Eingabedatei gelesen.

SQL3519W  Synchronisationspunkt am Beginn des Ladevorgangs. Eingabesatzzähler = "1".

SQL3520W  Synchronisationspunkt für Ladevorgang erfolgreich.

SQL3515W  Die Phase "LOAD" wurde beendet (Zeit: "02/16/2009  15:12:13.445718").

   Anzahl gelesener Zeilen          = 1
   Anzahl übersprungener Zeilen     = 0
   Anzahl geladener Zeilen          = 1
   Anzahl zurückgewiesener Zeilen   = 0
   Anzahl gelöschter Zeilen         = 0
   Anzahl festgeschriebener Zeilen  = 1
```

8. Vergewissern Sie sich nach dem Laden der Daten in die Tabelle, dass auf dem TSM-Server ein Backup-Image, ein Ladekopieimage und eine Protokolldatei vorhanden ist, indem Sie die folgende Abfrage für die Datenbank zample ausführen:

   ```
   bar:/home/roecken/sqllib/adsm> db2adutl query db zample
options "-asnodename=clusternode"
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
   Retrieving FULL DATABASE BACKUP information.
    1 Time: 20090216151025  Oldest log: S0000000.LOG  DB Partition Number: 0
           Sessions: 1

   Retrieving INCREMENTAL DATABASE BACKUP information.
       No INCREMENTAL DATABASE BACKUP images found for ZAMPLE

   Retrieving DELTA DATABASE BACKUP information.
       No DELTA DATABASE BACKUP images found for ZAMPLE

   Retrieving TABLESPACE BACKUP information.
       No TABLESPACE BACKUP images found for ZAMPLE

   Retrieving INCREMENTAL TABLESPACE BACKUP information.
       No INCREMENTAL TABLESPACE BACKUP images found for ZAMPLE

   Retrieving DELTA TABLESPACE BACKUP information.
       No DELTA TABLESPACE BACKUP images found for ZAMPLE

   Retrieving LOAD COPY information.
   ```

```
 1 Time: 20090216151213

Retrieving LOG ARCHIVE information.
      Log file: S0000000.LOG, Chain Num: 0, DB Partition Number: 0,
Taken at: 2009-02-16-15.10.38
```

9. In diesem Beispiel ist der zweite Computer (dps) noch nicht für die knotenübergreifende Recovery der Datenbank zample konfiguriert. Vergewissern Sie sich mit dem folgenden Befehl, dass diesem Benutzer und diesem Computer keine Daten zugeordnet sind:

   ```
   dps:/home/regress9/sqllib/adsm> db2adutl query db zample
   ```

   Die folgenden Informationen werden zurückgegeben:

   ```
     --- Database directory is empty ---
   Warning: There are no file spaces created by DB2 on the ADSM server
   Warning: No DB2 backup images found in ADSM for any alias.
   ```

10. Rufen Sie mit dem folgenden Befehl vom TSM-Server eine Liste der Objekte für die Datenbank zample ab, die dem Proxy-Knoten clusternode zugeordnet sind:

    ```
    dps:/home/regress9/sqllib/adsm> db2adutl query db zample
       options="-asnodename=clusternode"
    ```

    Die folgenden Informationen werden zurückgegeben:

    ```
    --- Database directory is empty ---

  Query for database ZAMPLE

    Retrieving FULL DATABASE BACKUP information.
     1 Time: 20090216151025  Oldest log: S0000000.LOG  DB Partition Number: 0
            Sessions: 1

    Retrieving INCREMENTAL DATABASE BACKUP information.
        No INCREMENTAL DATABASE BACKUP images found for ZAMPLE

    Retrieving DELTA DATABASE BACKUP information.
        No DELTA DATABASE BACKUP images found for ZAMPLE

    Retrieving TABLESPACE BACKUP information.
        No TABLESPACE BACKUP images found for ZAMPLE

    Retrieving INCREMENTAL TABLESPACE BACKUP information.
        No INCREMENTAL TABLESPACE BACKUP images found for ZAMPLE

    Retrieving DELTA TABLESPACE BACKUP information.
        No DELTA TABLESPACE BACKUP images found for ZAMPLE

    Retrieving LOAD COPY information.
     1 Time: 20090216151213

    Retrieving LOG ARCHIVE information.
          Log file: S0000000.LOG, Chain Num: 0, DB Partition Number: 0,
    Taken at: 2009-02-16-15.10.38
    ```

    Diese Informationen stimmen mit den zuvor generierten TSM-Informationen überein und bestätigen damit, dass dieses Image auf dem Computer dps wiederhergestellt werden kann.

11. Stellen Sie mit dem folgenden Befehl die Datenbank zample vom TSM-Server auf dem Computer dps wieder her:

    ```
    dps:/home/regress9> db2 restore db zample use tsm options
       "'-asnodename=clusternode'" without prompting
    ```

    Die folgenden Informationen werden zurückgegeben:

    ```
    DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.
    ```

    Anmerkung: Wenn die Datenbank zample auf dps bereits vorhanden wäre, würde der Parameter OPTIONS weggelassen, und der Datenbankkonfigurati-

onsparameter *vendoropt* würde verwendet. Dieser Konfigurationsparameter überschreibt den Parameter OPTIONS für eine BACKUP- oder RESTORE-Operation.

12. Führen Sie eine aktualisierende Wiederherstellung durch, um die Transaktionen anzuwenden, die beim Erstellen einer neuen Tabelle und beim Laden neuer Daten in der Protokolldatei für die Datenbank `zample` erfasst wurden. In diesem Beispiel schlägt die aktualisierende Wiederherstellung (ROLLFORWARD) in der Datenbank fehl, weil das Dienstprogramm für aktualisierende Wiederherstellung die Protokolldateien nicht finden kann, da keine Benutzer- und Computerinformationen angegeben sind:

    ```
    dps:/home/regress9> db2 rollforward db zample to end of logs and stop
    ```

    Der Befehl gibt den folgenden Fehler zurück:

    ```
    SQL4970N  Die aktualisierende Recovery der Datenbank "ZAMPLE"
    kann wegen fehlender Protokolldatei(en) auf Knoten "0" nicht den angegebenen
    Endpunkt (Protokollende oder angegebener Zeitpunkt) erreichen.
    ```

    Veranlassen Sie das Dienstprogramm für aktualisierende Recovery, die Protokolldateien auf einem anderen Computer zu suchen, indem Sie den entsprechenden Wert für *logarchopt* angeben. Verwenden Sie in diesem Beispiel den folgenden Befehl, um den Datenbankkonfigurationsparameter *logarchopt1* festzulegen und nach Protokolldateien für den Benutzer `roecken` und den Computer `bar` zu suchen:

    ```
    dps:/home/regress9> db2 update db cfg for zample using logarchopt1
       "'-asnodename=clusternode'"
    ```

13. Ermöglichen Sie mit dem folgenden Befehl dem Dienstprogramm für aktualisierende Wiederherstellung die Verwendung der Backup- und Ladekopieimages, indem Sie den Datenbankkonfigurationsparameter *vendoropt* festlegen:

    ```
    dps:/home/regress9> db2 update db cfg for zample using VENDOROPT
       "'-asnodename=clusternode'"
    ```

14. Mit dem folgenden Befehl können Sie die knotenübergreifende Datenrecovery beenden, indem die in der Protokolldatei der Datenbank `zample` erfassten Transaktionen angewendet werden:

    ```
    dps:/home/regress9> db2 rollforward db zample to end of logs and stop
    ```

    Die folgenden Informationen werden zurückgegeben:

    ```
                                  Status der aktualisierenden Recovery

        Aliasname der Eingabedatenbank              = zample
        Anzahl der Knoten, die Status zurückgaben   = 1

        Knotennummer                                = 0
        Aktualisierende Recovery: Status            = nicht anstehend
        Nächste zu lesende Protokolldatei           =
        Verarbeitete Protokolldateien               = S0000000.LOG - S0000000.LOG
    Letzte festgeschriebene Transaktion             = 2009-02-16-20.10.38.000000 UTC

      DB20000I  Der Befehl ROLLFORWARD wurde erfolgreich ausgeführt.
    ```

    Die Datenbank `zample` auf dem Computer `dps` unter dem Benutzer `regress9` wurde auf denselben Stand wie die Datenbank auf dem Computer `bar` unter dem Benutzer `roecken` wiederhergestellt.

Recovery einer gelöschten Tabelle

Es ist möglich, dass Sie versehentlich eine Tabelle mit Daten löschen, die noch benötigen. In solchen Fällen empfiehlt es sich, nach dem Löschen einer Tabelle die kritischen Tabellen wiederherstellbar zu machen. Die Tabellendaten können durch eine Datenbankrecovery mit anschließender aktualisierender Recovery bis zu einem bestimmten Zeitpunkt, der vor dem Löschzeitpunkt der Tabelle liegen muss, wiederhergestellt werden. Dies kann zeitaufwendig sein, wenn die Datenbank sehr

groß ist. Zudem sind die Daten während der Recovery nicht verfügbar. Die Funktion zur Recovery gelöschter Tabellen ermöglicht es, gelöschte Tabellen über einen Restore und anschließende aktualisierende Recovery auf Tabellenbereichsebene wiederherzustellen. Dies nimmt weniger Zeit in Anspruch, als eine Recovery der gesamten Datenbank, und die Datenbank steht den Benutzern während der Recovery der Tabelle weiterhin zur Verfügung.

Damit eine gelöschte Tabelle wiederherstellbar ist, muss für den Tabellenbereich, in dem sich die Tabelle befindet, der Parameter DROPPED TABLE RECOVERY aktiviert sein. Dies kann während der Erstellung des Tabellenbereichs vorgenommen werden oder durch Aufrufen der Anweisung ALTER TABLESPACE. Die Option DROPPED TABLE RECOVERY ist tabellenbereichsspezifisch und kann nur für einen regulären Tabellenbereich angegeben werden. Wenn Sie feststellen möchten, ob ein Tabellenbereich wiederherstellbar ist, können Sie die Spalte DROP_RECOVERY in der Katalogtabelle SYSCAT.TABLESPACES abfragen.

Die Option DROPPED TABLE RECOVERY ist bei der Erstellung eines Tabellenbereichs automatisch aktiviert. Wenn Sie keinen Tabellenbereich für die Recovery einer gelöschten Tabelle aktivieren möchten, können Sie entweder die Option DROPPED TABLE RECOVERY explizit auf OFF setzen, wenn Sie den Befehl CREATE TABLESPACE absetzen, oder Sie können den Befehl ALTER TABLESPACE verwenden, um die Recovery für gelöschte Tabellen für einen vorhandenen Tabellenbereich zu inaktivieren. Die Funktion DROPPED TABLE RECOVERY kann einen Leistungseinfluss auf die aktualisierende Recovery haben, wenn zu viele gelöschte Tabellen wiederherzustellen sind oder die Verlaufsdatei groß ist.

Wenn eine Anweisung DROP TABLE für eine Tabelle ausgeführt wird, deren Tabellenbereich für eine Recovery gelöschter Tabellen aktiviert ist, wird ein zusätzlicher Eintrag in den Protokolldateien vorgenommen, der die gelöschte Tabelle angibt. Außerdem wird in der Datei des Recoveryprotokolls ein Eintrag mit Informationen erstellt, die für die erneute Erstellung der Tabelle verwendet werden können.

Für partitionierte Tabellen ist die Option DROPPED TABLE RECOVERY immer aktiviert, selbst wenn DROPPED TABLE RECOVERY für nicht partitionierte Tabellen in einem oder mehreren Tabellenbereichen inaktiviert ist. Für eine partitionierte Tabelle wird nur ein Protokollsatz über die gelöschte Tabelle geschrieben. Dieser Protokollsatz reicht aus, um alle Datenpartitionen der Tabelle wiederherzustellen.

Für die Datentypen, die von einer gelöschten Tabelle wiederhergestellt werden können, gibt es einige Einschränkungen. Folgende Daten können nicht wiederhergestellt werden:

- Die Option DROPPED TABLE RECOVERY kann nicht für temporäre Tabellen verwendet werden.
- Die den Zeilentypen zugeordneten Metadaten. (Die Daten selbst werden wiederhergestellt, nicht jedoch die Metadaten.) Die Daten in der Hierarchietabelle der typisierten Tabelle werden wiederhergestellt. Diese Daten sind möglicherweise größeren Umfangs, als sie in der gelöschten typisierten Tabelle den Anschein machten.
- XML-Daten. Wenn Sie versuchen, eine gelöschte Tabelle mit XML-Daten wiederherzustellen, sind die entsprechenden Spaltendaten leer.

Wenn sich die Tabelle im Status *Reorganisation anstehend* befand, als sie gelöscht wurde, stimmt die DDL-Anweisung CREATE TABLE in der Protokolldatei nicht genau mit der Importdatei überein. Die Importdatei liegt in dem Format der Tabelle vor, das sie hatte, bevor die erste Anweisung ALTER mit empfohlenem REORG

ausgeführt wurde. Die Anweisung CREATE TABLE in der Protokolldatei stimmt jedoch im Status der Tabelle einschließlich der Ergebnisse aller Anweisungen ALTER TABLE überein.

Mit LOAD oder IMPORT zu verwendende Änderungswerte für Dateitypen
Geben Sie für eine Recovery der Tabelle durch Laden oder Importieren die folgenden Änderungswerte für Dateitypen an:

- Der Dateitypänderungswert USEGRAPHICCODEPAGE muss im Befehl IMPORT oder LOAD verwendet werden, wenn die wiederherzustellenden Daten den Datentyp GRAPHIC oder VARGRAPHIC aufweisen. Der Grund hierfür liegt darin, dass mehrere Codepages enthalten sein könnten.
- Der Dateitypänderungswert DELPRIORITYCHAR muss im Befehl IMPORT oder LOAD verwendet werden. Er ermöglicht es den Befehlen LOAD und IMPORT, Zeilen syntaktisch zu analysieren, die Zeilenumbruchzeichen innerhalb von Zeichen- oder Grafikspaltendaten enthalten.

Es kann nur jeweils eine gelöschte Tabelle wiederhergestellt werden. Gehen Sie wie folgt vor, um eine gelöschte Tabelle wiederherzustellen:

1. Geben Sie die gelöschte Tabelle an, indem Sie den Befehl LIST HISTORY DROPPED TABLE aufrufen. Die Kennung der gelöschten Tabelle wird in der Spalte für die Backup-ID ausgegeben.
2. Führen Sie den Restore auf Datenbank- oder Tabellenbereichsebene mit einem Backup-Image aus, das vor dem Löschen der Tabelle erstellt wurde.
3. Erstellen Sie ein Exportverzeichnis, in das die Dateien geschrieben werden können, die die Tabellendaten enthalten. Auf das Verzeichnis muss entweder von allen Datenbankpartitionen aus zugegriffen werden können, oder es muss in allen Datenbankpartitionen vorhanden sein. Jede Datenbankpartition erstellt automatisch Unterverzeichnisse in diesem Exportverzeichnis. Diese Unterverzeichnisse sind nach dem Muster NODEnnnn benannt, wobei nnnn für die Datenbankpartition oder Knotennummer steht. Die Datendateien, die die gelöschten Tabellendaten enthalten, so wie sie zuvor in jeder Datenbankpartition vorhanden waren, werden in ein untergeordnetes Unterverzeichnis des Namens data exportiert. Beispiel:

   ```
   \export_directory\NODE0000\data.
   ```

4. Führen Sie eine aktualisierende Recovery bis zu einem Zeitpunkt nach dem Löschen der Tabelle aus. Verwenden Sie dabei für den Befehl ROLLFORWARD DATABASE die Option RECOVER DROPPED TABLE. Sie können alternativ eine aktualisierende Recovery bis zum Ende der Protokolldateien durchführen, sodass Aktualisierungen an anderen Tabellen im Tabellenbereich oder der Datenbank nicht verloren gehen.
5. Erstellen Sie die Tabelle mithilfe der Anweisung CREATE TABLE anhand der Recoveryprotokolldatei erneut.
6. Importieren Sie die Tabellendaten, die während der aktualisierenden Recovery exportiert wurden, in die Tabelle. Wenn sich die Tabelle im Status *Reorganisation anstehend* befand, als sie gelöscht wurde, muss der Inhalt der DDL-Anweisung CREATE TABLE eventuell geändert werden, sodass er dem Inhalt der Datendatei entspricht.

Recovery nach Systemabsturz

Transaktionen (bzw. UOWs), die für eine Datenbank ausgeführt werden, können auf unerwartete Weise unterbrochen werden. Wenn eine Störung auftritt, bevor alle Änderungen, die Bestandteil der UOW (Unit of Work - Arbeitseinheit) sind, beendet und festgeschrieben wurden, verbleibt die Datenbank in einem inkonsistenten und unbrauchbaren Status. Die *Recovery nach einem Systemabsturz* versetzt die Datenbank wieder in einen konsistenten und verwendbaren Status. Dies geschieht durch Rollback der unvollständigen Transaktionen und Beenden der festgeschriebenen Aktionen, die sich zum Zeitpunkt des Systemabsturzes noch im Hauptspeicher befanden (Abb. 19). Wenn eine Datenbank sich in einem konsistenten und verwendbaren Status befindet, hat sie einen Punkt erreicht, der als „Konsistenzzustand" bezeichnet wird.

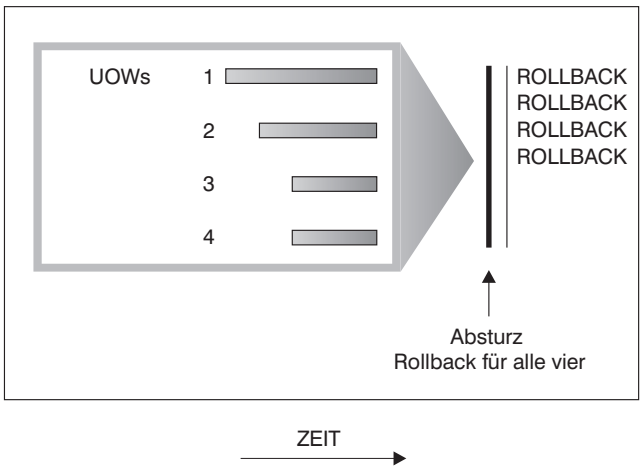


Abbildung 19. Rollback von UOWs (Recovery nach einem Systemabsturz)

Ein *Transaktionsfehler* ergibt sich aus einem schwer wiegenden Fehler bzw. einer Bedingung, die zur abnormalen Beendigung der Datenbank oder des Datenbankmanagers führt. Durch nur teilweise beendete oder zum Zeitpunkt des Fehlers noch nicht auf Platte geschriebene UOWs wird die Datenbank inkonsistent. Daher muss die Datenbank nach einem Transaktionsfehler wiederhergestellt werden. Folgende Bedingungen können zu einem Transaktionsfehler führen:

- Ein Stromausfall auf der Maschine, durch den der Datenbankmanager und die Datenbankpartitionen abnormal beendet werden
- Ein Hardwarefehler, z. B. Datenverlust im Hauptspeicher, oder ein Platten-, CPU- oder Netzwerkfehler
- Ein schwer wiegender Betriebssystemfehler, durch den DB2 abnormal beendet wird
- Eine Anwendung, die abnormal beendet wird

Wenn unvollständige UOWs vom Datenbankmanager automatisch rückgängig gemacht werden sollen, aktivieren Sie den Datenbankkonfigurationsparameter für automatischen Wiederanlauf (*autorestart*), indem Sie ihn auf `ON` setzen. (Dies ist der Standardwert.) Wenn Sie den automatischen Neustart inaktivieren möchten, setzen Sie den Datenbankkonfigurationsparameter *autorestart* auf `OFF`. Ist der automatische Neustart inaktiviert, müssen Sie im Fall eines Datenbankfehlers den Befehl RESTART DATABASE absetzen. Wenn die Datenbank-E/A vor dem Auftreten des Absturzes ausgesetzt wurde, müssen Sie mit dem Befehl RESTART DATABASE die Option WRITE RESUME angeben, damit die Recovery nach dem Systemabsturz

fortgesetzt wird. Der Neustart der Datenbank wird im Protokoll mit den Benachrichtigungen für die Systemverwaltung aufgezeichnet.

Wenn für eine Datenbank, für die die aktualisierende Recovery aktiviert ist (d. h., der Konfigurationsparameter *logarchmeth1* ist auf einen anderen Wert als OFF gesetzt), eine Recovery nach einem Systemabsturz durchgeführt wird und während dieser Recovery ein Fehler auftritt, der auf einen einzelnen Tabellenbereich zurückzuführen ist, wird dieser Tabellenbereich in den Offlinestatus versetzt. Auf ihn kann erst wieder zugegriffen werden, nachdem er repariert wurde.Die Recovery nach dem Systemabsturz wird fortgesetzt. Nach der Beendigung der Recovery nach Systemabsturz kann auf die anderen Tabellenbereiche in der Datenbank zugegriffen werden, und es können Verbindungen zur Datenbank hergestellt werden. Wenn jedoch der in den Offlinestatus versetzte Tabellenbereich die Systemkataloge enthält, muss er repariert werden, bevor Verbindungen hergestellt werden können.

Recovery beschädigter Tabellenbereiche

Ein beschädigter Tabellenbereich enthält einen oder mehrere Container, auf den/die nicht zugegriffen werden kann. Dies wird häufig durch Datenträgerprobleme verursacht, die entweder permanent (z. B. eine defekte Platte) oder temporär sind (z. B. eine Platte, die offline ist, oder ein abgehängtes Dateisystem).

Wenn der beschädigte Tabellenbereich der Bereich für die Systemkatalogtabellen ist, kann die Datenbank nicht erneut gestartet werden. Wenn die Containerprobleme nicht so behoben werden können, dass die ursprünglichen Daten erhalten bleiben, sind nur die folgenden Optionen verfügbar:

- Wiederherstellen der Datenbank
- Wiederherstellen des Katalogtabellenbereichs

Anmerkung:

1. Der Restore des Tabellenbereichs ist nur für wiederherstellbare Datenbanken zulässig, da die Datenbank aktualisierend wiedergeherstellt werden muss.
2. Wenn Sie den Katalogtabellenbereich wiederherstellen, müssen Sie eine aktualisierende Recovery bis zum Ende der Protokolle durchführen.

Wenn es sich bei dem beschädigten Tabellenbereich *nicht* um den Tabellenbereich des Systemkatalogs handelt, versucht DB2, so viel von der Datenbank wie möglich verfügbar zu machen.

Wenn der beschädigte Tabellenbereich der einzige Tabellenbereich für temporäre Tabellen ist, müssen Sie einen neuen Tabellenbereich für temporäre Tabellen erstellen, sobald eine Verbindung zur Datenbank hergestellt werden kann. Nach der Erstellung kann der neue Tabellenbereich für temporäre Tabellen verwendet werden, und der normale Datenbankbetrieb, der einen Tabellenbereich für temporäre Tabellen erfordert, kann wieder aufgenommen werden. Den Offlinetabellenbereich für temporäre Tabellen können Sie löschen, wenn Sie es wünschen. Für die Reorganisation von Tabellen, die mit einem Tabellenbereich für temporäre Systemtabellen arbeitet, sind die folgenden speziellen Aspekte zu berücksichtigen:

- Wenn der Konfigurationsparameter *indexrec* der Datenbank oder des Datenbankmanagers auf RESTART gesetzt ist, müssen alle ungültigen Indizes während der Datenbankaktivierung erneut erstellt werden. Dies schließt Indizes aus einer Reorganisation ein, die während der Erstellungsphase abgestürzt ist.

- Wenn in einem beschädigten Tabellenbereich für temporäre Tabellen unvollständige Reorganisationsanforderungen vorhanden sind, müssen Sie möglicherweise den Konfigurationsparameter *indexrec* auf ACCESS setzen, um Fehler beim Neustart zu verhindern.

Recovery von Tabellenbereichen in wiederherstellbaren Datenbanken

Wenn eine Recovery nach Systemabsturz erforderlich ist, befindet sich der beschädigte Tabellenbereich offline und lässt keinen Zugriff mehr zu. Er wird in den Status 'Aktualisierende Recovery anstehend' versetzt. Wenn keine weiteren Probleme auftreten, kann ein Neustart die Datenbank auch mit dem beschädigten Tabellenbereich wieder online verfügbar machen. Wieder online ist der beschädigte Tabellenbereich nicht verwendbar, jedoch kann der Rest der Datenbank genutzt werden. Zur Reparatur des beschädigten Tabellenbereichs und zur Wiederherstellung seiner Verwendbarkeit führen Sie die folgende Prozedur aus.

Verwenden Sie eine der folgenden Prozeduren, um den beschädigten Tabellenbereich verwendbar zu machen:

- Methode 1
  1. Beheben Sie die Fehler an den beschädigten Containern ohne Verlust der ursprünglichen Daten.
  2. Führen Sie eine aktualisierende Recovery bis zum Ende der Protokolle für den Tabellenbereich aus.

     Anmerkung: Die Operation zur aktualisierenden Recovery versucht dabei zunächst, den Tabellenbereich wieder vom Offlinestatus in den normalen Status zurückzuversetzen.
- Methode 2
  1. Beheben Sie die Fehler an den beschädigten Containern mit oder ohne Verlust der ursprünglichen Daten.
  2. Führen Sie eine Restoreoperation für den Tabellenbereich aus.
  3. Führen Sie eine aktualisierende Recovery bis zum Ende der Protokolle oder bis zu einem bestimmten Zeitpunkt für den Tabellenbereich aus.

Recovery von Tabellenbereichen in nicht wiederherstellbaren Datenbanken

Wenn eine Recovery nach Systemabsturz erforderlich ist, jedoch Tabellenbereiche beschädigt sind, könn Sie die Datenbank nur dann erfolgreich neu starten, wenn die beschädigten Tabellenbereiche aus der Datenbank gelöscht werden. In einer nicht wiederherstellbaren Datenbank werden die Protokolle, die zur Recovery der beschädigten Tabellenbereiche erforderlich sind, nicht behalten. Daher besteht die einzig korrekte Aktion für solche Tabellenbereiche darin, sie zu löschen.

Gehen Sie wie folgt vor, um eine Datenbank mit beschädigten Tabellenbereichen erneut zu starten:

1. Rufen Sie eine Operation ohne Qualifikationsmerkmal zum erneuten Starten einer Datenbank auf. Wenn keine beschädigten Tabellenbereiche vorhanden sind, ist diese Operation erfolgreich. Wenn sie fehlschlägt (SQL0290N), finden Sie im Protokoll mit den Benachrichtigungen für die Systemverwaltung eine vollständige Liste der Tabellenbereiche, die momentan beschädigt sind.
2. Wenn Sie bereit sind, alle diese Tabellenbereiche zu löschen, leiten Sie eine weitere Neustartoperation für die Datenbank ein, wobei Sie alle beschädigten Ta-

bellenbereiche unter der Option DROP PENDING TABLESPACES auflisten. Wenn ein beschädigter Tabellenbereich in der Liste DROP PENDING TABLESPACES aufgeführt ist, wird der Tabellenbereich in den Status 'Löschen anstehend' versetzt, und Sie müssen den Tabellenbereich nach Abschluss der Recoveryoperation löschen.

Die Neustartoperation wird ohne Recovery der angegebenen Tabellenbereiche fortgesetzt. Wenn ein beschädigter Tabellenbereich *nicht* in der Liste DROP PENDING TABLESPACES aufgeführt ist, schlägt die Operation RESTART DATABASE mit der SQL-Nachricht SQL0290N fehl.

Anmerkung: Wenn der Name eines Tabellenbereichs in der Liste DROP PENDING TABLESPACES aufgeführt wird, bedeutet dies nicht, dass der Tabellenbereich in den Status 'Löschen anstehend' versetzt wird. Er wird nur dann in diesen Status versetzt, wenn der Tabellenbereich während des Neustarts beschädigt ist.

3. Ist die Neustartoperation erfolgreich, rufen Sie den Befehl LIST TABLESPACES auf, um zu ermitteln, welche Tabellenbereiche sich im Status 'Löschen anstehend' befinden.
4. Setzen Sie DROP TABLESPACE-Anweisungen ab, um die einzelnen Tabellenbereiche zu löschen, die sich im Status 'Löschen anstehend' befinden. Danach können Sie den Speicherbereich zurückfordern, den die beschädigten Tabellenbereiche verwendet haben, oder Sie können die Tabellenbereiche erneut erstellen.
5. Wenn Sie diese Tabellenbereiche nicht löschen (d. h. die darin enthaltenen Daten nicht verlieren) wollen, haben Sie folgende Möglichkeiten:
   - Beheben Sie die Fehler an den beschädigten Containern (ohne Verlust der ursprünglichen Daten).
   - Setzen Sie den Befehl RESTART DATABASE erneut ab.
   - Führen Sie eine Operation RESTORE DATABASE aus.

Begrenzen der Auswirkungen von Datenträgerausfällen

Zur Verringerung der Wahrscheinlichkeit eines Datenträgerfehlers und zur Vereinfachung der Recovery der Daten nach einem solchen Fehler sollten Sie folgende Maßnahmen ergreifen:

- Spiegeln oder duplizieren Sie die Platten, die die Daten und Protokolle für wichtige Datenbanken enthalten.
- Verwenden Sie eine RAID-Konfiguration (RAID - Redundant Array of Independent Disks), wie beispielsweise RAID-Stufe 5.
- Sehen Sie in einer Umgebung mit partitionierten Datenbanken für die Behandlung der Daten und der Protokolle in der Katalogpartition ein genau definiertes Verfahren vor. Da diese Datenbankpartition für die Verwaltung der Datenbank von entscheidender Bedeutung ist, sollten Sie Folgendes beachten:
  - Stellen Sie sicher, dass er sich auf einer zuverlässigen Platte befindet.
  - Duplizieren Sie ihn.
  - Erstellen Sie häufig Backups.
  - Speichern Sie keine Benutzerdaten auf ihm.

Schützen vor Plattenfehlern

Wenn Sie sich Gedanken über die Möglichkeit einer Beschädigung von Daten oder aktiven Protokollen aufgrund eines Plattenfehlers machen, sollten Sie über den Ein-

satz von Systemen nachdenken, die eine gewisse Toleranz gegenüber Plattenfehlern gewährleisten. Im Allgemeinen bietet sich hier die Verwendung einer *Platteneinheit* (Disk Array) an.

Platteneinheiten werden manchmal einfach als RAID (Redundant Array of Independent Disks) bezeichnet. Platteneinheiten können darüber hinaus durch Software auf Betriebssystem- oder Anwendungsebene implementiert werden. Das Unterscheidungsmerkmal zwischen Hardware- und Softwareplatteneinheiten ist die Art der CPU-Verarbeitung von E/A-Anforderungen. Bei Hardwareplatteneinheiten werden die E/A-Aktivitäten von Plattencontrollern verwaltet, während dies bei Softwareplatteneinheiten vom Betriebssystem bzw. von einer Anwendung übernommen wird.

Hardwareplatteneinheiten

Bei einer Hardwareplatteneinheit werden mehrere Platten von einem Plattencontroller mit eigener CPU verwendet und verwaltet. Da die gesamte Logik, die zur Verwaltung der zu dieser Einheit gehörenden Platten erforderlich ist, im Plattencontroller enthalten ist, ist diese Implementierung vom Betriebssystem unabhängig.

Es gibt verschiedene Typen der RAID-Architektur, die sich in Funktion und Leistung unterscheiden. Die derzeit gängigsten Typen sind jedoch RAID-Stufe 1 und Stufe 5.

RAID-Stufe 1 ist auch als Spiegelung (Mirroring) oder Duplizierung (Duplexing) von Platten bekannt. Bei der *Plattenspiegelung* werden Daten (eine vollständige Datei) von einer Platte auf eine zweite Platte kopiert, wobei nur ein einziger Plattencontroller verwendet wird. Die Vorgänge bei der *Plattenduplizierung* ähneln denen der Plattenspiegelung, jedoch sind hierbei die Platten an einen zweiten Plattencontroller angeschlossen (wie zwei SCSI-Adapter). Diese Verfahren bieten einen guten Datenschutz: es kann eine der beiden Platten ausfallen, und die Daten bleiben über die jeweils andere Platte verfügbar. Bei der Plattenduplizierung ist auch bei Ausfall eines Plattencontrollers der vollständige Schutz der Daten gewährleistet. Die Leistung ist gut, es wird jedoch die doppelte Anzahl an Platten benötigt.

RAID-Stufe 5 implementiert das plattenübergreifende Lesen und Schreiben von Daten (Striping) und die plattenübergreifende Parität nach Sektoren. Die Paritätsinformationen werden mit Daten verzahnt und nicht auf einem dedizierten Laufwerk gespeichert. Der Datenschutz ist gewährleistet: Wenn eine Platte ausfällt, stehen die Daten über die Informationen von den anderen Platten zusammen mit den verteilten Paritätsinformationen immer noch zur Verfügung. Die Leseleistung ist gut, die Schreibleistung jedoch nicht. Für eine RAID-5-Konfiguration sind mindestens drei identische Platten erforderlich. Die Menge des zusätzlich erforderlichen Plattenspeicherplatzes für den Systemaufwand variiert mit der Anzahl der Platten in der Platteneinheit. Im Fall einer RAID-5-Konfiguration mit fünf Platten beträgt der Speichermehraufwand 20 %.

Die RAID-Stufe 1+0 (10) umfasst das Spiegeln und Striping von Daten auf mindestens zwei Platten. Beim Spiegeln werden die Daten auf mindestens zwei Platten gleichzeitig geschrieben. Hierdurch erhalten Sie dieselbe Fehlertoleranz wie bei RAID-Stufe 1. Beim Striping werden die Daten in Blöcke unterteilt, und jeder Block wird in ein separates Plattenlaufwerk geschrieben. Hiermit wird eine hohe E/A-Leistung erzielt, indem das E/A-Aufkommen auf viele Kanäle und Laufwerke verteilt wird; die RAID-Stufe 1+0 reduziert jedoch den effektiven Plattenspeicherplatz auf die Hälfte, da alle Daten gespiegelt werden. Für die Implementierung der RAID-Stufe 10 sind mindestens 4 Laufwerke erforderlich.

Die RAID-Stufe 0+1 wird als gespiegeltes Array implementiert, dessen Segmente RAID-0-Arrays sind, und weist dieselbe Fehlertoleranz auf wie die RAID-Stufe 5. Dies liefert eine hohe E/A-Geschwindigkeit, indem das E/A-Aufkommen auf viele Kanäle und Laufwerke verteilt wird. Die RAID-Stufe 0+1 darf jedoch nicht mit der RAID-Stufe 1+0 verwechselt werden. Der Ausfall eines einzelnen Laufwerks führt dazu, dass das gesamte Array praktisch ein Array der RAID-Stufe 0 wird.

Bei Verwendung einer RAID-Platteneinheit (jedoch nicht RAID-Stufe 0) können Sie trotz einer ausgefallenen Platte auf die Daten der Platteneinheit zugreifen. Wenn Hot Plug- oder Hot Swap-fähige Platten in der Platteneinheit verwendet werden, kann die ausgefallene Platte während des Betriebs der Platteneinheit gegen eine Ersatzplatte ausgetauscht werden. Wenn bei einer RAID-5-Konfiguration zwei Platten gleichzeitig ausfallen, gehen alle Daten verloren (jedoch ist die Wahrscheinlichkeit eines gleichzeitigen Ausfalls zweier Platten sehr gering).

Für Ihre Protokolle können Sie eine Hardwareplatteneinheit der RAID-Stufe 1 oder eine Softwareplatteneinheit in Betracht ziehen, da diese Möglichkeiten eine Wiederherstellbarkeit bis zu dem Punkt des Ausfalls bieten und eine gute Schreibleistung zeigen, was für Protokolle wichtig ist. Verwenden Sie den Konfigurationsparameter *mirrorlogpath*, um auf einem Dateisystem der RAID-Stufe 1 einen Spiegelprotokollpfad anzugeben. In Fällen, in denen Zuverlässigkeit von essenzieller Bedeutung ist (d. h., dass keine Zeit für eine Recovery der Daten nach einem Plattenfehler verloren gehen darf) und die Schreibleistung nicht so wichtig ist, kann eine RAID-5-Konfiguration für die Hardwareplatteneinheit in Betracht kommen. Wenn hingegen die Schreibleistung eine erhebliche Rolle spielt und Sie diese trotz des Aufwands für zusätzlichen Plattenspeicherplatz sicherstellen wollen, ziehen Sie eine RAID-1-Hardwareplatteneinheit sowohl für Ihre Daten als auch Ihre Protokolle in Betracht.

Weitere Informationen zu den verfügbaren RAID-Stufen finden Sie unter folgender Adresse (nur englisch): http://www.acnc.com/04_01_00.html

Softwareplatteneinheiten

Eine Softwareplatteneinheit leistet im wesentlichen dasselbe wie eine Hardwareplatteneinheit, jedoch wird der Plattenverkehr entweder vom Betriebssystem oder von einem auf dem Server aktiven Anwendungsprogramm verwaltet. Wie alle anderen Programme auch steht die Softwareplatteneinheit bei der Nutzung der CPU- und Systemressourcen in einer Konkurrenzsituation. Dies ist keine gute Lösung für ein System mit knappen CPU-Ressourcen, und es ist zu bedenken, dass die Gesamtleistung der Platteneinheit von der Auslastung und Kapazität der CPU des Servers abhängig ist.

Eine typische Softwareplatteneinheit bietet Funktionen zur Spiegelung von Platten. Obwohl redundante Platten erforderlich sind, ist eine Softwareplatteneinheit relativ preiswert zu implementieren, da kostenintensive Plattencontroller nicht benötigt werden.

Vorsicht:
Wenn sich das Bootlaufwerk des Betriebssystems in der Platteneinheit befindet, kann Ihr System nicht starten, wenn dieses Laufwerk ausfällt. Wenn das Laufwerk ausfällt, bevor die Platteneinheit aktiv ist, kann die Platteneinheit keinen Zugriff auf das Laufwerk ermöglichen. Ein Bootlaufwerk sollte von der Platteneinheit getrennt betrieben werden.

Begrenzen der Auswirkungen von Transaktionsfehlern

Zur Verringerung der Auswirkungen von Transaktionsfehlern versuchen Sie, Folgendes sicherzustellen:

- Eine ununterbrochene Stromversorgung für jeden DB2-Server
- Ausreichenden Plattenspeicherplatz für Datenbankprotokolle auf allen Datenbankpartitionen
- Zuverlässige Kommunikationsverbindungen zwischen den Datenbankpartitionsservern in einer Umgebung mit partitionierten Datenbanken
- Synchronisation der Systemuhren in einer Umgebung mit partitionierten Datenbanken

Recovery nach Transaktionsfehlern in einer Umgebung mit partitionierten Datenbanken

Tritt ein Transaktionsfehler in einer Umgebung mit partitionierten Datenbanken auf, ist in der Regel eine Datenbankrecovery sowohl auf dem ausgefallenen Datenbankpartitionsserver als auch auf allen anderen, an der Transaktion beteiligten Datenbankpartitionsservern erforderlich:

- Eine Recovery nach Systemabsturz wird auf dem ausgefallenen Datenbankpartitionsserver durchgeführt, nachdem die vorausgegangene Bedingung korrigiert wurde.
- Die *Recovery der Datenbankpartitionen nach einem Fehler* erfolgt auf den anderen (weiterhin aktiven) Datenbankpartitionsservern unmittelbar nach Feststellung des Fehlers.

In einer Umgebung mit partitionierten Datenbanken ist der Datenbankpartitionsserver, auf dem die Anwendung übergeben wird, die Koordinatorpartition, und der erste Agent, der für die Anwendung aktiv wird, ist der Koordinatoragent. Der Koordinatoragent ist für die Verteilung der Arbeit auf andere Datenbankpartitionsserver verantwortlich und protokolliert, welche Datenbankpartitionsserver an der Transaktion beteiligt sind. Wenn die Anwendung eine COMMIT-Anweisung für eine Transaktion ausführt, schreibt der Koordinatoragent die Transaktion mithilfe des Protokolls zum zweiphasigen Commit fest. Während der ersten Phase sendet die Koordinatorpartition eine PREPARE-Anforderung an alle anderen an der Transaktion beteiligten Datenbankpartitionsserver. Die Server antworten daraufhin wie folgt:

READ-ONLY
: Auf diesem Server gab es keine Datenänderung.

YES
: Auf diesem Server gab es eine Datenänderung.

NO
: Aufgrund eines Fehlers wurde der Server nicht für das Commit vorbereitet.

Wenn einer der Server mit `NO` antwortet, wird die Transaktion rückgängig gemacht. Andernfalls beginnt die Koordinatorpartition mit der zweiten Phase.

Während der zweiten Phase schreibt die Koordinatorpartition einen Commitprotokollsatz und sendet anschließend eine Commitanforderung an alle Server, die mit `YES` geantwortet haben. Wenn alle anderen Datenbankpartitionsserver das Commit durchgeführt haben, senden sie eine Bestätigung des Commits an die Koordinatorpartition. Die Transaktion ist abgeschlossen, wenn der Koordinatoragent von allen beteiligten Servern die Commitbestätigungen empfangen hat. Wenn dies der Fall ist, schreibt der Koordinatoragent einen FORGET-Protokollsatz.

Recovery auf einem aktiven Datenbankpartitionsserver nach Transaktionsfehler

Wenn ein Datenbankpartitionsserver feststellt, dass ein anderer Server inaktiv ist, werden alle Arbeiten, an denen der ausgefallene Datenbankpartitionsserver beteiligt ist, gestoppt:

- Wenn der noch aktive Datenbankpartitionsserver die Koordinatorpartition für eine Anwendung ist und die Anwendung auf dem ausgefallenen Datenbankpartitionsserver ausgeführt wird (und nicht zum Commit bereit ist), wird der Koordinatoragent unterbrochen, um Recoverymaßnahmen durchzuführen. Wenn der Koordinatoragent in der zweiten Phase der Commitverarbeitung ist, empfängt die Anwendung die Nachricht SQL0279N und verliert die Datenbankverbindung. Ansonsten sendet der Koordinatoragent eine Rollback-Anforderung an alle an der Transaktion beteiligten Server, und die Anwendung empfängt die Nachricht SQL1229N.
- Wenn der ausgefallene Datenbankpartitionsserver die Koordinatorpartition für die Anwendung war, werden Agenten, die noch für die Anwendung auf den aktiven Servern arbeiten, unterbrochen, um Recoverymaßnahmen durchzuführen. Die Transaktion wird lokal auf jeder Datenbankpartition rückgängig gemacht, auf der sich die Transaktion nicht im Status vorbereiteter Transaktionen befindet. Auf den Datenbankpartitionen, auf denen sich die Transaktion im Status vorbereiteter Transaktionen befindet, wird die Transaktion zu einer unbestätigten Transaktion. Die Koordinatordatenbankpartition ist nicht darüber informiert, dass die Transaktion auf einigen Datenbankpartitionen unbestätigt ist, weil sie nicht verfügbar ist.
- Wenn die Anwendung die Verbindung zu dem ausgefallenen Datenbankpartitionsserver (bevor er ausfiel) herstellte, aber weder der lokale Datenbankpartitionsserver noch der ausgefallene Datenbankpartitionsserver die Koordinatorpartition ist, werden Agenten, die für diese Anwendung aktiv sind, unterbrochen. Die Koordinatorpartition sendet eine Nachricht über eine Rollback-Operation oder eine Trennoperation (DISCONNECT) an die anderen Datenbankpartitionsserver. Die Transaktion ist nur auf den Datenbankpartitionsservern unbestätigt, die weiterhin aktiv sind, wenn die Koordinatorpartition die Nachricht SQL0279 zurückgibt.

Jeder Prozess (wie z. B. ein Agent oder ein Deadlock-Detektor), der versucht, eine Anforderung an den ausgefallenen Server zu senden, wird informiert, dass er die Anforderung nicht senden kann.

Recovery nach Transaktionsfehler auf dem ausgefallenen Datenbankpartitionsserver

Wenn der Transaktionsfehler zu einer abnormalen Beendigung des Datenbankmanagers führt, können Sie den Befehl db2start mit der Option RESTART angeben, um den Datenbankmanager nach dem Neustart der Datenbankpartition erneut zu starten. Falls der Neustart der Datenbankpartition nicht möglich ist, können Sie den Befehl db2start auch in einer anderen Datenbankpartition zum Starten des Datenbankmanagers verwenden.

Die abnormale Beendigung des Datenbankmanagers kann zur Inkonsistenz einiger Datenbankpartitionen auf dem Server führen. Um diese Datenbankpartitionen wieder in einen konsistenten Status zu versetzen, kann auf einem Datenbankpartitionsserver wie folgt eine Recovery nach einem Systemabsturz ausgelöst werden:

- Explizit durch den Befehl RESTART DATABASE

- Implizit durch eine CONNECT-Anforderung, wenn der Datenbankkonfigurationsparameter *autorestart* auf `ON` gesetzt ist

Bei der Recovery nach einem Systemabsturz werden die Protokollsätze in den aktiven Protokolldateien erneut angewendet, um sicherzustellen, dass die Ergebnisse aller vollständigen Transaktionen in der Datenbank vorhanden sind. Nachdem alle Änderungen erneut angewendet wurden, werden alle nicht festgeschriebenen Transaktionen *außer* unbestätigten Transaktionen lokal rückgängig gemacht. In einer Umgebung mit partitionierten Datenbanken gibt es zwei Typen unbestätigter Transaktionen:

- Auf einem Datenbankpartitionsserver, der nicht die Koordinatorpartition ist, gilt eine Transaktion als unbestätigt, wenn sie zwar vorbereitet (PREPARE), aber noch nicht festgeschrieben (COMMIT) wurde.
- In der Koordinatorpartition ist eine Transaktion unbestätigt, wenn sie festgeschrieben (COMMIT), aber noch nicht als abgeschlossen protokolliert wurde (d. h., der Protokollsatz FORGET wurde noch nicht geschrieben). Diese Situation tritt ein, wenn der Koordinatoragent noch nicht alle Commitbestätigungen von allen Servern empfangen hat, die für die Anwendung aktiv waren.

Bei der Recovery nach einem Systemabsturz wird versucht, alle unbestätigten Transaktionen durch eine der folgenden Aktionen aufzulösen. Die durchgeführte Aktion ist davon abhängig, ob der Datenbankpartitionsserver die Koordinatorpartition für eine Anwendung war:

- Wenn der Server, der erneut gestartet wird, nicht die Koordinatorpartition für die Anwendung ist, sendet er eine Abfragenachricht an den Koordinatoragenten, um das Ergebnis der Transaktion festzustellen.
- Wenn der erneut gestartete Server die Koordinatorpartition für die Anwendung *ist*, sendet er eine Nachricht an alle anderen Agenten (untergeordneten Agenten), von denen der Koordinatoragent immer noch Commitbestätigungen erwartet.

Es ist möglich, dass durch eine Recovery nach einem Systemabsturz nicht alle unbestätigten Transaktionen aufgelöst werden können. Einige der Datenbankpartitionsserver sind z. B. möglicherweise nicht verfügbar. Wenn die Koordinatorpartition die Recovery nach einem Systemabsturz vor den anderen, an der Transaktion beteiligen Datenbankpartitonen abschließt, kann die Recovery nach einem Systemabsturz die unbestätigten Transaktionen nicht auflösen. Dies ist die erwartete Funktionsweise, da die Recovery nach einem Systemabsturz von jeder Datenbankpartition unabhängig durchgeführt wird. In diesem Fall wird die SQL-Warnung SQL1061W zurückgegeben. Unbestätigte Transaktionen belegen Ressourcen, z. B. Sperren und Speicherbereich für aktive Protokolle. Daher ist es möglich, dass ein Punkt erreicht wird, an dem keine Änderungen an der Datenbank mehr durchgeführt werden können, weil der Speicherbereich für die aktiven Protokolldateien durch unbestätigte Transaktionen belegt ist. Aus diesem Grund sollten Sie nach einer Recovery nach einem Systemabsturz feststellen, ob unbestätigte Transaktionen verblieben sind, und alle Datenbankpartitionsserver, die zur Auflösung der unbestätigten Transaktionen erforderlich sind, so schnell wie möglich wieder verfügbar machen. In einigen Szenarios, in denen **LOCKTIMEOUT** -1 ist oder wenn der Parameter SET CURRENT LOCK TIMEOUT WAIT verwendet wird, können Sperren, die von unbestätigten Transaktionen verursacht werden, dazu führen, dass die Knoten-Recovery auf dem Koordinatorknoten nicht beendet werden kann. Diese Situation wird am besten dadurch aufgelöst, dass für sämtliche Datenbankpartitionsserver eine Recovery durchgeführt wird.

Anmerkung: In einer Umgebung mit partitionierten Datenbankservern wird der Befehl RESTART auf Knotenbasis ausgeführt. Um sicherzustellen, dass die Datenbank auf allen Knoten erneut gestartet wird, verwenden Sie den folgenden empfohlenen Befehl:

```
db2_all "db2 restart database <datenbankname>"
```

Wenn mindestens ein Server, der zur Auflösung einer unbestätigten Transaktion benötigt wird, nicht rechtzeitig wieder verfügbar gemacht werden kann und der Zugriff auf Datenbankpartitionen auf anderen Servern erforderlich ist, können Sie die unbestätigte Transaktion durch eine heuristische Entscheidung manuell auflösen. Mithilfe des Befehls LIST INDOUBT TRANSACTIONS können Sie die unbestätigte Transaktion auf dem Server abfragen, festschreiben oder rückgängig machen.

Anmerkung: Der Befehl LIST INDOUBT TRANSACTIONS wird auch in einer verteilten Transaktionsumgebung verwendet. Zur Unterscheidung zwischen den beiden Typen unbestätigter Transaktionen enthält das Feld für die Quelle (*Originator*) in der Ausgabe des Befehls LIST INDOUBT TRANSACTIONS eine der folgenden Angaben:

- DB2 Enterprise Server Edition. Dies zeigt an, dass die Transaktion aus einer Umgebung mit partitionierten Datenbanken stammt.
- XA. Dies zeigt an, dass die Transaktion aus einer verteilten Umgebung stammt.

Identifizieren des ausgefallenen Datenbankpartitionsservers

Wenn ein Datenbankpartitionsserver ausfällt, empfängt die Anwendung normalerweise einen der folgenden SQLCODE-Werte. Die Methode zum Identifizieren des jeweils ausgefallenen Datenbankmanagers hängt vom empfangenen SQLCODE-Wert ab:

SQL0279N
: Dieser SQLCODE-Wert wird empfangen, wenn ein Datenbankpartitionsserver, der an einer Transaktion beteiligt ist, während der Commitverarbeitung beendet wird.

SQL1224N
: Dieser SQLCODE-Wert wird empfangen, wenn der ausgefallene Datenbankpartitionsserver die Koordinatorpartition für die Transaktion ist.

SQL1229N
: Dieser SQLCODE-Wert wird empfangen, wenn der ausgefallene Datenbankpartitionsserver nicht die Koordinatorpartition für die Transaktion ist.

Welcher Datenbankpartitionsserver ausgefallen ist, kann in zwei Schritten festgestellt werden. Der SQL-Kommunikationsbereich, der zum SQLCODE-Wert SQL1229N gehört, enthält in der sechsten Feldposition des Felds *sqlerrd* die Knotennummer des Servers, der den Fehler erkannte. (Die Knotennummer, die für den Server geschrieben wird, entspricht der Knotennummer in der Datei `db2nodes.cfg`.) Auf dem Datenbankpartitionsserver, der den Fehler feststellt, wird eine Nachricht mit der Knotennummer des ausgefallenen Servers in das Protokoll mit den Benachrichtigungen für die Systemverwaltung geschrieben.

Anmerkung: Wenn mehrere logische Knoten auf einem Prozessor verwendet werden, kann der Ausfall eines logischen Knotens den Ausfall anderer logischer Knoten auf demselben Prozessor verursachen.

Recovery nach dem Ausfall eines Datenbankpartitionsservers

Gehen Sie wie folgt vor, um eine Recovery nach dem Ausfall eines Datenbankpartitionsservers auszuführen.

1. Beheben Sie den Fehler, der den Ausfall verursachte.
2. Starten Sie den Datenbankmanager erneut, indem Sie auf einem beliebigen Datenbankpartitionsserver den Befehl db2start absetzen.
3. Starten Sie die Datenbank erneut, indem Sie auf dem bzw. den ausgefallenen Datenbankpartitionsserver(n) den Befehl RESTART DATABASE absetzen.

Wiederherstellen von unbestätigten Transaktionen auf Mainframe- oder Mid-Range-Servern

Recovery von unbestätigten Transaktionen auf dem Host, wenn bei DB2 Connect der DB2-Synchronisationspunktmanager konfiguriert ist

Wenn Ihre Anwendung auf einen Host- oder einen System i-Datenbankserver während einer Transaktion zugegriffen hat, gibt es einige Unterschiede, wie für unbestätigte Transaktionen eine Recovery durchgeführt wird. Für den Zugriff auf den Host- oder System i-Datenbankserver wird DB2 Connect verwendet. Die Schritte zur Recovery unterscheiden sich, wenn bei DB2 Connect der DB2-Synchronisationspunktmanager konfiguriert ist.

Die Recovery von unbestätigten Transaktionen auf dem Host- oder System i-Server wird normalerweise automatisch vom Transaktionsmanager (TM) und dem DB2-Synchronisationspunktmanager (SPM) ausgeführt. Eine unbestätigte Transaktion auf einem Host- oder System i-Server belegt keine Ressourcen an der lokalen DB2-Position, sondern auf dem Host- oder System i-Server, so lange die Transaktion an dieser Position unbestätigt ist. Wenn der Administrator des Host- oder System i-Servers feststellt, dass eine heuristische Entscheidung getroffen werden muss, kann sich der Administrator an den lokalen DB2-Datenbankadministrator wenden (z. B . telefonisch), um festzustellen, ob für die Transaktion auf dem Host- oder System i-Server eine Commitoperation oder ein Rollback durchgeführt werden soll. Wenn dies geschieht, kann der Befehl LIST DRDA INDOUBT TRANSACTIONS verwendet werden, um den Status der Transaktion auf der lokalen DB2 Connect-Instanz zu ermitteln. In den meisten Fällen können Sie in einer SNA-Kommunikationsumgebung folgendermaßen vorgehen:

1. Stellen Sie wie nachstehend gezeigt eine Verbindung zum SPM her:

```
db2 => connect to db2spm

   Datenbankverbindungsinformationen

 Datenbankserver                 = SPM0500
 SQL-Berechtigungs-ID            = CRUS
 Aliasname der lokalen Datenbank = DB2SPM
```

2. Führen Sie den Befehl LIST DRDA INDOUBT TRANSACTIONS aus, um die dem SPM bekannten unbestätigten Transaktionen anzuzeigen. Das folgende Beispiel zeigt eine dem SPM bekannte unbestätigte Transaktion. Der Datenbankname db_name ist der lokale Aliasname für den Host- oder System i-Server. Der Partner-LU-Name partner_lu ist der vollständig qualifizierte LU-Name des Host- oder System i-Servers. Dies erlaubt die bestmögliche Identifikation des Host- oder System i-Servers und sollte vom Anrufer des Host- oder System i-Servers bereitgestellt werden. Die Kennung der logischen Arbeitseinheit luwid stellt eine eindeutige Kennung für eine Transaktion bereit und ist auf allen Hosts und System i-Servern verfügbar. Wenn die fragliche Transaktion ange-

zeigt wird, kann anhand des Feldes uow_status das Ergebnis der Transaktion festgestellt werden, wenn der Wert C (COMMIT) oder R (ROLLBACK) ist. Wenn Sie den Befehl LIST DRDA INDOUBT TRANSACTIONS mit dem Parameter WITH PROMPTING absetzen, können Sie die Transaktion interaktiv festschreiben, rückgängig machen oder ignorieren.

```
db2 => list drda indoubt transactions
 Unbestätigte DRDA-Transaktionen:
1.db_name: DBAS3    db_alias: DBAS3    role: AR
  uow_status: C  partner_status: I  partner_lu: USIBMSY.SY12DQA
corr_tok: USIBMST.STB3327L
 luwid: USIBMST.STB3327.305DFDA5DC00.0001
 xid: 53514C2000000017 00000000544D4442 0000000000305DFD A63055E962000000
           00035F
```

3. Wenn eine unbestätigte Transaktion für partner_lu und für luwid nicht angezeigt wird bzw. wenn der Befehl LIST DRDA INDOUBT TRANSACTIONS folgende Ausgabe ergibt:

   ```
   db2 => list drda indoubt transactions
   SQL1251W  Keine Daten für manuelle Abfrage zurückgegeben.
   ```

   dann wurde die Transaktion rückgängig gemacht.

Es gibt jedoch noch eine weitere Situation, die zwar unwahrscheinlich ist, aber dennoch auftreten kann. Wenn eine unbestätigte Transaktion mit einer ordnungsgemäßen luwid für partner_lu angezeigt wird, aber der uow_status den Wert I aufweist, kann der SPM nicht entscheiden, ob die Transaktion festzuschreiben oder mit ROLLBACK rückgängig zu machen ist. In dieser Situation sollten Sie den Parameter WITH PROMPTING verwenden, um für die Transaktion entweder eine Commitoperation oder ein Rollback auf der DB2 Connect-Workstation durchzuführen. Lassen Sie DB2 Connect anschließend mit dem Host oder System i-Server auf der Basis der heuristischen Entscheidung resynchronisieren.

Recovery von unbestätigten Transaktionen auf dem Host, wenn DB2 Connect nicht den DB2-Synchronisationspunktmanager verwendet

Wenn Ihre Anwendung auf einen Host- oder einen System i-Datenbankserver während einer Transaktion zugegriffen hat, gibt es einige Unterschiede, wie für unbestätigte Transaktionen eine Recovery durchgeführt wird. Für den Zugriff auf Host- oder System i-Datenbankserver wird DB2 Connect verwendet. Die Recoveryschritte sind unterschiedlich, wenn bei DB2 Connect der DB2-Synchronisationspunktmanager konfiguriert ist.

Verwenden Sie die Informationen in diesem Abschnitt, wenn die TCP/IP-Konnektivität verwendet wird, um DB2 für z/OS in einer Aktualisierung auf mehreren Systemen entweder über DB2 Connect Personal Edition oder DB2 Connect Enterprise Edition zu aktualisieren und der DB2-Synchronisationspunktmanager nicht verwendet wird. Die Recovery unbestätigter Transaktionen ist in diesem Fall anders als bei Verwendung des DB2-Synchronisationspunktmanagers.Wenn eine unbestätigte Transaktion in dieser Umgebung auftritt, wird auf dem Client, auf dem Datenbankserver und/oder in der Transaktionsmanagerdatenbank (TMD), je nachdem, wo der Fehler festgestellt wurde, ein Alert-Eintrag generiert. Der Alert-Eintrag wird in die Datei `db2alert.log` geschrieben.

Die Resynchronisation aller unbestätigten Transaktionen erfolgt automatisch, sobald der Transaktionsmanager und alle beteiligten Datenbanken sowie ihre Verbindungen wieder verfügbar sind. Es ist besser, eine automatische Resynchronisation

zuzulassen, als heuristisch eine Entscheidung beim Datenbankserver herbeizuführen. Wenn dies jedoch erforderlich ist, gehen Sie wie im Folgenden beschrieben vor.

Anmerkung: Da der DB2-Synchronisationspunktmanager nicht verwendet wird, können Sie den Befehl LIST DRDA INDOUBT TRANSACTIONS nicht verwenden.

1. Setzen Sie auf dem z/OS-Host den Befehl DISPLAY THREAD TYPE(INDOUBT) ab.

   Stellen Sie anhand dieser Liste die Transaktion fest, die Sie heuristisch beenden möchten. Weitere Informationen zum Befehl DISPLAY finden Sie im Handbuch *DB2 for z/OS Command Reference*. Die angezeigte LUWID kann derselben luwid in der Transaktionsmanagerdatenbank zugeordnet werden.
2. Setzen Sie den Befehl RECOVER THREAD(<LUWID>) ACTION(ABORT|COMMIT) in Abhängigkeit von der Aktion ab, die Sie ausführen möchten.

   Weitere Informationen zum Befehl RECOVER THREAD finden Sie im Handbuch *DB2 for z/OS Command Reference*.

Recovery nach einem Katastrophenfall

Unter dem Begriff *Recovery nach einem Katastrophenfall* werden die Aktivitäten zusammengefasst, die zum Wiederherstellen der Datenbank nach einem Brand, einem Erdbeben, nach Vandalismus oder anderen zerstörerischen Ereignissen erforderlich sind. Ein Plan zur Recovery nach einem Katastrophenfall kann Folgendes vorsehen:

- Einen Zweitstandort, der im Notfall zur Verfügung steht
- Eine andere Maschine, auf der die Datenbank wiederhergestellt werden kann
- Die Aufbewahrung von Datenbankbackups, Tabellenbereichsbackups oder beiden sowie archivierten Protokollen an einem anderen Standort

Wenn Ihr Plan zur Recovery nach einem Katastrophenfall vorsieht, die gesamte Datenbank auf einer anderen Maschine wiederherzustellen, wird empfohlen, zumindest über ein Datenbankgesamtbackup und alle archivierten Protokolldateien für die Datenbank zu verfügen. Es ist zwar auch möglich, eine Datenbank erneut zu erstellen (Rebuild), wenn ein Gesamtbackup aller Tabellenbereiche in der Datenbank vorhanden ist, jedoch kann diese Methode eine große Anzahl von Backup-Images erfordern und mehr Zeit in Anspruch nehmen als die Recovery mithilfe eines Datenbankgesamtbackups.

Sie können auch eine Bereitschaftsdatenbank auf dem aktuellen Stand halten, indem Sie die Protokolle, die archiviert werden, auf sie anwenden. Oder Sie können die Datenbank- oder Tabellenbereichsbackups und die Protokollarchive am Bereitschaftsstandort lagern und einen Restore bzw. eine aktualisierende Recovery nur durchführen, wenn ein Katastrophenfall eingetreten ist. (Im letzteren Fall sind möglichst junge Backup-Images vorzuziehen.) Im Katastrophenfall ist es gewöhnlich jedoch nicht möglich, alle Transaktionen bis zum Zeitpunkt des Eintritts der Katastrophe wiederherzustellen.

Der Nutzen eines Tabellenbereichsbackups zur Recovery nach einem Katastrophenfall hängt vom Ausmaß der Beschädigung ab. In der Regel ist eine Recovery nach einem Katastrophenfall weniger kompliziert und zeitaufwendig, wenn Sie die gesamte Datenbank wiederherstellen. Daher sollte am Bereitschaftsstandort ein Datenbankgesamtbackup bereitgehalten werden. Im Fall einer beschädigten Platte kann die Recovery mithilfe eines Tabellenbereichsbackups jedes Tabellenbereichs

auf dieser Platte durchgeführt werden. Wenn Sie aufgrund eines Plattenfehlers (oder aus einem anderen Grund) keinen Zugriff auf einen Container mehr haben, können Sie den Container an einer anderen Position wiederherstellen.

Eine weitere Möglichkeit, Ihre Daten vor einem partiellen oder vollständigen Standortausfall zu schützen, ist die Implementierung der DB2-Funktion HADR (High Availability Disaster Recovery). Nach der Installation und Konfiguration von HADR bietet diese Funktion Schutz vor Datenverlust durch Replizieren von Datenänderungen aus einer Quellendatenbank, der so genannten Primärdatenbank, in eine Zieldatenbank, der so genannten Bereitschaftsdatenbank.

Sie können Ihre Daten gegen einen partiellen oder vollständigen Standortausfall auch durch Replikation schützen. Die Replikation ermöglicht ein regelmäßiges Kopieren von Daten in mehrere ferne Datenbanken. Die DB2-Datenbank stellt eine Reihe von Replikationstools zur Verfügung, bei denen Sie angeben können, welche Daten zu kopieren sind, in welche Datenbanktabellen die Daten zu kopieren sind und wie häufig Aktualisierungen zu kopieren sind.

Darüber hinaus kann auch eine Speicherspiegelung, zum Beispiel mithilfe von Peer-to-Peer Remote Copy (PPRC) zum Schutz Ihrer Daten eingesetzt werden. PPRC ermöglicht ein synchrones Kopieren eines Datenträgers bzw. einer Platte zum Schutz gegen Katastrophen.

DB2 stellt Ihnen zur Planung der Recovery nach einem Fehler mehrere Optionen zur Auswahl. Abhängig von Ihren Geschäftsanforderungen können Sie entweder Tabellenbereichsbackups oder Datenbankgesamtbackups als Schutz gegen Datenverlust verwenden oder überlegen, ob sich Ihre Umgebung für eine Lösung wie HADR eignet. Für welche Lösung Sie sich auch entscheiden, Sie sollten Ihre Recoveryprozeduren in jedem Fall in einer Testumgebung testen, bevor Sie sie in Ihrer Produktionsumgebung implementieren.

Versionsrecovery

Eine *Versionsrecovery* ist die Wiederherstellung einer früheren Version der Datenbank mithilfe eines Images der Datenbank, das im Rahmen einer Backup-Operation erstellt wurde. Sie können diese Recoverymethode mit nicht wiederherstellbaren Datenbanken verwenden (d. h. Datenbanken, für die Sie über keine archivierten Protokolldateien verfügen). Sie können diese Methode auch bei wiederherstellbaren Datenbanken einsetzen, indem Sie mit dem Befehl RESTORE DATABASE die Option WITHOUT ROLLING FORWARD verwenden. Durch eine RESTORE-Operation der Datenbank wird die gesamte Datenbank mithilfe eines zu einem früheren Zeitpunkt erstellten Backup-Images wiederhergestellt. Ein Datenbank-Backup ermöglicht es Ihnen, eine Datenbank in dem Status wiederherzustellen, in dem sie sich zum Zeitpunkt des Backups befand. Es gehen jedoch alle UOWs verloren, die seit dem Zeitpunkt des Backups bis zum Eintreten des Fehlers ausgeführt wurden (siehe Abb. 20 auf Seite 295).

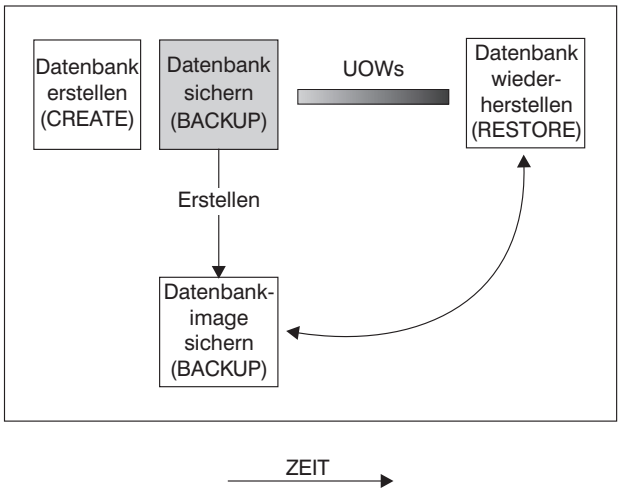


Abbildung 20. Versionsrecovery. Zeigt an, dass alle UOWs verloren gehen, die vom Zeitpunkt des Backups ausgehend bis zum Eintreten des Fehlers ausgeführt wurden.

Bei Einsatz der Methode der Versionsrecovery müssen Sie regelmäßige Gesamtbackups der Datenbank planen und durchführen.

In einer Umgebung mit partitionierten Datenbanken ist eine Datenbank auf viele Datenbankpartitionsserver (oder Knoten) verteilt. Sie müssen alle Datenbankpartitionen wiederherstellen, und die Backup-Images, die Sie zum Restore der Datenbank verwenden, müssen alle zum gleichen Zeitpunkt erstellt worden sein. (Jede Datenbankpartition wird separat gesichert und wiederhergestellt.) Ein Backup, bei dem alle Backupkopien der Datenbankpartitionen zur gleichen Zeit erstellt werden, wird als *Versionsbackup* bezeichnet.

Aktualisierende Recovery

Sie können die Methode der *aktualisierenden Recovery* nur dann einsetzen, wenn Sie ein Backup der Datenbank erstellt und die Protokolldateien archiviert haben (dazu muss einer der Datenbankkonfigurationsparameter *logarchmeth1* oder *logarchmeth2* auf einen anderen Wert als `OFF` gesetzt sein). Der Restore der Datenbank unter Angabe der Option WITHOUT ROLLING FORWARD (d. h. ohne aktualisierende Recovery) entspricht der Methode der Versionsrecovery. Die Datenbank wird in einem Status wiederhergestellt, der dem Zeitpunkt entspricht, zu dem das Image des Offline-Backups erstellt wurde. Wenn Sie für die Datenbank ein Restore durchführen und *nicht* die Option WITHOUT ROLLING FORWARD für die Restore-Datenbankoperation, befindet sich die Datenbank am Ende der Restoreoperation im Status "aktualisierende Recovery anstehend". Dadurch kann die aktualisierende Recovery durchgeführt werden.

Anmerkung: Die Option WITHOUT ROLLING FORWARD kann in folgenden Fällen nicht verwendet werden:
- Sie führen die RESTORE-Operation aus einem Online-Backup-Image aus.
- Sie setzen eine RESTORE-Operation auf Tabellenbereichsebene ab.

Es können zwei Typen der aktualisierenden Recovery in Erwägung gezogen werden:

- *Aktualisierende Recovery der Datenbank*. Bei diesem Typ der aktualisierenden Recovery werden im Anschluss an den Restore der Datenbank Transaktionen angewendet, die in den Datenbankprotokollen aufgezeichnet sind (siehe Abb. 21 auf Seite 296).

In den Datenbankprotokollen werden alle Änderungen aufgezeichnet, die an der Datenbank vorgenommen werden. Diese Methode vervollständigt die Recovery der Datenbank bis zu ihrem Status an einem bestimmten Zeitpunkt oder bis zu ihrem Status unmittelbar vor dem Fehler, das heißt bis zum Ende der aktiven Protokolldateien.

In einer Umgebung mit partitionierten Datenbanken ist eine Datenbank über viele Datenbankpartitionen verteilt, und der Befehl ROLLFORWARD DATABASE muss in der Datenbankpartition abgesetzt werden, in der sich die Katalogtabellen für die Datenbank befinden (Katalogpartition). Wenn Sie eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt durchführen, müssen alle Datenbankpartitionen aktualisierend wiederhergestellt werden, um sicherzustellen, dass alle Datenbankpartitionen den gleichen Stand haben. Wenn Sie eine einzelne Datenbankpartition wiederherstellen müssen, können Sie eine aktualisierende Recovery bis zum Ende der Protokolle durchführen, um sie auf den gleichen Stand wie die anderen Datenbankpartitionen in der Datenbank zu bringen. Bei der aktualisierenden Recovery einer einzelnen Datenbankpartition kann nur die Recovery bis zum Ende der Protokolle verwendet werden. Die Recovery bis zu einem bestimmten Zeitpunkt wird immer auf *alle* Datenbankpartitionen angewendet.

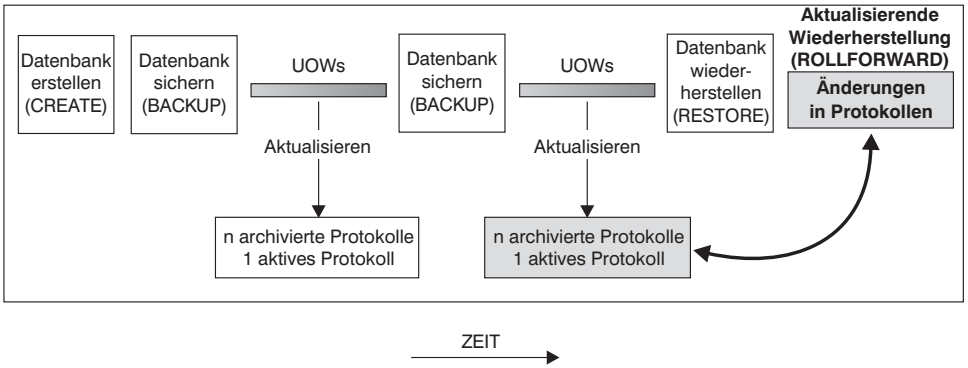


Abbildung 21. Aktualisierende Recovery einer Datenbank. Im Falle einer lange laufenden Transaktion kann es mehr als ein aktives Protokoll geben.

- *Aktualisierende Recovery eines Tabellenbereichs*. Wenn für eine Datenbank die aktualisierende Recovery aktiviert ist, können auch Tabellenbereiche gesichert, wiederhergestellt und aktualisierend wiederhergestellt werden (siehe Abb. 22 auf Seite 297). Zum Wiederherstellen und aktualisierenden Wiederherstellen eines Tabellenbereichs benötigen Sie ein Backup-Image entweder der gesamten Datenbank (d. h. aller Tabellenbereiche) oder mindestens eines einzelnen Tabellenbereichs. Außerdem benötigen Sie die Protokollsätze, die die wiederherzustellenden Tabellenbereiche betreffen. Sie können einen Tabellenbereich durch die Protokolle bis zu einem von zwei Punkten aktualisierend wiederherstellen:
  - Dem Ende der Protokolldateien
  - Einem bestimmten Zeitpunkt (als Recovery *bis zu einem bestimmten Zeitpunkt* bezeichnet)

Eine aktualisierende Recovery von Tabellenbereichen kann in den beiden folgenden Situationen durchgeführt werden:

- Ein Tabellenbereich befindet sich nach seinem Restore immer im Status *Aktualisierende Recovery anstehend* und muss aktualisierend wiederhergestellt werden. Rufen Sie den Befehl ROLLFORWARD DATABASE auf, um die Protokolle auf die Tabellenbereiche entweder bis zu einem Zeitpunkt oder bis zum Ende der Protokolldateien anzuwenden.

- Wenn sich mindestens ein Tabellenbereich nach einer Recovery infolge eines Systemabsturzes im Status *Aktualisierende Recovery anstehend* befindet, beheben Sie zuerst das Problem mit dem Tabellenbereich. In einigen Fällen kann ein Fehler am Tabellenbereich ohne Restore der Datenbank behoben werden. Beispielsweise kann ein Spannungsverlust den Tabellenbereich in den Status *Aktualisierende Recovery anstehend* versetzen. Ein Restore der Datenbank ist in diesem Fall nicht erforderlich. Nachdem das Problem mit dem Tabellenbereich behoben ist, können Sie den Befehl ROLLFORWARD DATABASE verwenden, um die Protokolle bis zum Ende der Protokolldateien auf die Tabellenbereiche anzuwenden. Wenn der Fehler vor der Recovery nach einem Systemabsturz behoben wird, reicht diese Recovery möglicherweise aus, um die Datenbank in einen konsistenten, verwendbaren Status zu versetzen.

  Anmerkung: Wenn der fehlerhafte Tabellenbereich die Systemkatalogtabellen enthält, können Sie die Datenbank nicht starten. Sie müssen den Tabellenbereich SYSCATSPACE wiederherstellen und anschließend eine aktualisierende Recovery bis zum Ende der Protokolldateien durchführen.

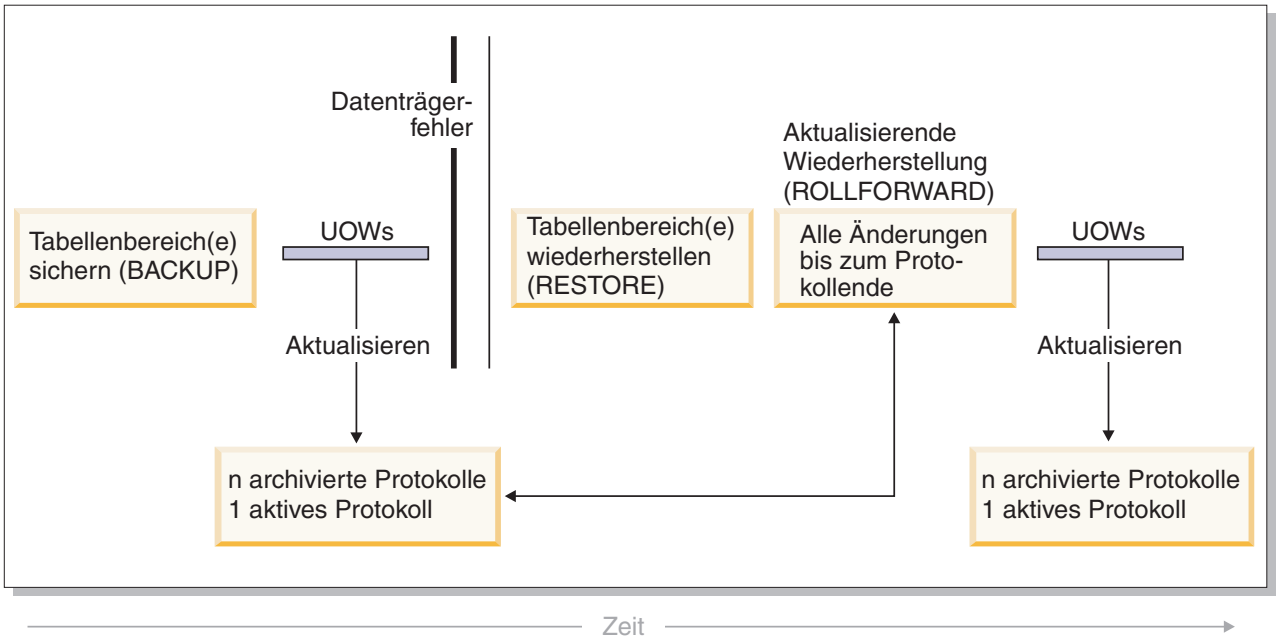


Abbildung 22. Aktualisierende Recovery von Tabellenbereichen. Im Falle einer lange laufenden Transaktion kann es mehr als ein aktives Protokoll geben.

Wenn Sie in einer Umgebung mit partitionierten Datenbanken einen Tabellenbereich bis *zu einem bestimmten Zeitpunkt* aktualisierend wiederherstellen, müssen Sie die Liste der Datenbankpartitionen nicht angeben, auf die sich der Tabellenbereich verteilt. DB2 übergibt die Anforderung zur aktualisierenden Recovery an alle Datenbankpartitionen. Dies bedeutet, dass der Tabellenbereich in allen Datenbankpartitionen, auf denen der Tabellenbereich sich befindet, wiederhergestellt werden muss.

Wenn Sie in einer Umgebung mit partitionierten Datenbanken einen Tabellenbereich bis *zum Ende der Protokolle* aktualisierend wiederherstellen, müssen Sie die Liste der Datenbankpartitionen angeben, wenn Sie den Tabellenbereich *nicht* in allen Datenbankpartitionen aktualisierend wiederherstellen wollen. Wenn Sie alle Tabellenbereiche (in allen Datenbankpartitionen), die sich im Status *Aktualisierende Recovery anstehend* befinden, bis zum Ende der Protokolle aktualisierend wiederher-

stellen wollen, müssen Sie die Liste der Datenbankpartitionen nicht angeben. Standardmäßig wird die Anforderung zur aktualisierenden Recovery der Datenbank an alle Datenbankpartitionen gesendet.

Wenn Sie einen Tabellenbereich aktualisierend wiederherstellen, der einen Teil einer partitionierten Tabelle enthält, und Sie ihn bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen, müssen Sie auch alle anderen Tabellenbereiche, in denen sich die Tabelle befindet, bis zu demselben Zeitpunkt aktualisierend wiederherstellen. Sie können einen einzelnen Tabellenbereich, der einen Teil einer partitionierten Tabelle enthält, jedoch bis zum Ende der Protokolle aktualisierend wiederherstellen.

Anmerkung: Wenn eine partitionierte Tabelle (mit ATTACH) zugeordnete, (mit DETACH) getrennte oder (mit DROP) gelöschte Datenpartitionen hat, muss die aktualisierende Recovery bis zu einem bestimmten Zeitpunkt auch alle Tabellenbereiche für diese Datenpartitionen berücksichtigen. Um zu bestimmen, ob eine partitionierte Tabelle zugeordnete, getrennte oder gelöschte Datenpartitionen hat, fragen Sie die Katalogtabelle SYSDATAPARTITIONS ab.

Inkrementelles Backup und inkrementelle Recovery

Da die Größe von Datenbanken, speziell die von Warehouses, in zunehmenden Maße Terabyte- und Petabyte-Bereiche erreicht, steigen die für das Backup und die Recovery solcher Datenbanken erforderlichen Zeitaufwände und Anforderungen an die Hardwareressourcen enorm. Beim Umgang mit sehr umfangreichen Datenbanken ist es nicht immer die beste Vorgehensweise, jeweils die gesamte Datenbank mit allen Tabellenbereichen zu sichern, da der Speicherbedarf für mehrere Kopien enorm groß ist. Bedenken Sie Folgendes:

- Wenn nur ein kleiner Prozentsatz der Daten in einem Warehouse geändert wird, dürfte es nicht notwendig sein, die gesamte Datenbank zu sichern.
- Das Hinzufügen von Tabellenbereichen zu vorhandenen Datenbanken und das ausschließliche Sichern von Tabellenbereichen ist riskant, da nicht gewährleistet ist, dass zwischen zwei Tabellenbereichsbackups keine weiteren Änderungen an anderen, beim Backup nicht berücksichtigten Tabellenbereichen vorgenommen wurden.

Um dieser Problematik gerecht zu werden, bietet DB2 die Möglichkeit eines inkrementellen Backups und einer inkrementellen Recovery. Ein *inkrementelles Backup* ist ein Backup-Image, das nur die Seiten enthält, die seit dem letzten Backup aktualisiert wurden. Neben aktualisierten Daten und Indexseiten enthält jedes Image eines inkrementellen Backups auch die vollständigen ursprünglichen Metadaten der Datenbank (wie Datenbankkonfiguration, Tabellenbereichsdefinitionen, Datenbankprotokolle usw.), die normalerweise in Images von Gesamtbackups enthalten sind.

Anmerkung:

1. Enthält ein Tabellenbereich Langfeld- oder LOB-Daten (große Objekte) und wird ein inkrementelles Backup erstellt, werden alle Langfeld- oder LOB-Daten in das Backup-Image kopiert, wenn irgendwelche Seiten im betreffenden Tabellenbereich seit dem letzten Backup geändert wurden.
2. Wenn Sie ein inkrementelles Backup eines Tabellenbereichs erstellen, der eine benutzte Seite (d. h. eine Seite mit Daten, die geändert, aber noch nicht auf Platte geschrieben wurden) enthält, werden alle LOB-Daten im Backup gesichert. Normale Daten werden nur gesichert, wenn sie geändert wurden.

Es werden zwei Typen inkrementeller Backups unterstützt:

- *Inkrementell*. Bei einem inkrementellen Backup enthält das Backup-Image eine Kopie aller Datenbankdaten, die seit dem zuletzt erfolgreich durchgeführten Gesamtbackup geändert wurden. Dies wird auch als ein kumulatives Backup-Image bezeichnet, da die über einen gewissen Zeitraum hinweg erstellten inkrementellen Backups jeweils den Inhalt des vorherigen Images eines inkrementellen Backups enthalten. Der Vorgänger eines inkrementellen Backup-Image ist jeweils das neueste erfolgreiche Gesamtbackup desselben Objekts.
- *Delta*. Ein Deltabackup-Image bzw. ein inkrementelles Deltabackup ist eine Kopie aller Datenbankdaten, die seit dem zuletzt erfolgreich durchgeführten Backup (Gesamtbackup, inkrementelles Backup oder Deltabackup) des betreffenden Tabellenbereichs geändert wurden. Dies wird auch als differenzielles oder nicht kumulatives Backup-Image bezeichnet. Der Vorgänger eines Deltabackup-Images ist das jüngste erfolgreiche Backup, das eine Kopie aller im Deltabackup-Image enthaltenen Tabellenbereiche enthält.

Der wesentliche Unterschied zwischen Images von inkrementellen Backups und Delta-Backup-Images wird deutlich beim Erstellen von aufeinander folgenden Backups eines Objekts, das kontinuierlichen Änderungen unterworfen ist. Jedes weitere erstellte inkrementelle Image enthält den vollständigen Inhalt des zuvor erstellten inkrementellen Images sowie alle seit dem letzten Gesamtbackup geänderten oder neu hinzugekommenen Daten. Deltabackup-Images enthalten hingegen nur die seit dem letzten Backup geänderten Seiten, wobei der Typ des Backups keine Rolle spielt.

Kombinationen von inkrementellen Backups von Datenbanken und Tabellenbereichen sind zulässig, sowohl im Online- als auch im Offlinemodus. Gehen Sie bei der Planung Ihrer Backup-Strategie sorgfältig vor, da bei der Kombination aus inkrementellen Backups von Datenbanken und Tabellenbereichen nicht ausgeschlossen werden kann, dass es sich bei dem Vorgänger eines Datenbank-Backups (oder eines Tabellenbereichs-Backups mehrerer Tabellenbereiche) anstatt um ein Einzelimage um eine eindeutige Gruppe von zuvor zu unterschiedlichen Zeitpunkten erstellten Backups von Datenbanken und Tabellenbereichen handelt.

Um die Datenbank oder den Tabellenbereich in einem konsistenten Status wiederherzustellen, müssen Sie zur Recovery ein konsistentes Image des gesamten Objekts (Datenbank oder Tabellenbereich) verwenden und anschließend alle erforderlichen inkrementellen Backups in der nachfolgend beschriebenen Reihenfolge anwenden.

Zur Aktivierung der Überwachung von Datenbankaktualisierungen unterstützt DB2 den neuen Datenbankkonfigurationsparameter *trackmod*. Dieser Parameter kann einen der beiden folgenden Werte haben:

- NO. Inkrementelle Backups sind bei dieser Konfiguration nicht zulässig. Aktualisierungen von Datenbankseiten werden weder verfolgt noch aufgezeichnet. Dies ist der Standardwert.
- YES. Inkrementelle Backups sind bei dieser Konfiguration zulässig. Wird die Aktualisierungsüberwachung aktiviert, wird diese Änderung bei der nächsten erfolgreichen Verbindungsherstellung zur Datenbank wirksam. Bevor für einen bestimmten Tabellenbereich ein inkrementelles Backups durchgeführt werden kann, muss ein Gesamtbackup erstellt werden.

Bei SMS- und DMS-Tabellenbereichen erfolgt die Unterteilung dieser Überwachung auf Tabellenbereichsebene. Bei der Überwachung auf Tabellenbereichsebene zeigt eine Markierung für jeden Tabellenbereich an, ob sich in dem Tabellenbereich Sei-

ten befinden, die gesichert werden müssen. Falls in einem Tabellenbereich keine Seiten gesichert werden müssen, kann die Backup-Operation diesen Tabellenbereich komplett überspringen.

Die Überwachung von Datenbankaktualisierungen kann sich, wenn auch nur minimal, auf die Laufzeitleistung von Transaktionen auswirken, die Daten aktualisieren oder einfügen.

Restore von inkrementellen Backup-Images

- Eine Restoreoperation, bei der inkrementelle Backup-Images verwendet werden, umfasst grundsätzlich die folgenden Schritte:
  1. Identifizieren des inkrementellen Zielimages.

     Ermitteln Sie das endgültige wiederherzustellende Image, und fordern Sie eine Operation zum inkrementellen Restore vom DB2-Restoredienstprogramm an. Dieses Image wird als Zielimage des inkrementellen Restores bezeichnet, da es das letzte Image ist, das wiederhergestellt wird. Das Zielimage des inkrementellen Restores wird im Befehl RESTORE DATABASE mit dem Parameter TAKEN AT angegeben.
  2. Wiederherstellen des neuesten Gesamtbackups der Datenbank oder des Tabellenbereichs, um die Basis für die nachfolgend anzuwendenden inkrementellen Backup-Images zu erstellen.
  3. Wiederherstellen aller erforderlichen Gesamtbackup-Images oder inkrementellen Tabellenbereichsbackup-Images in der Reihenfolge ihrer Erstellung, aufbauend auf dem in Schritt 2 wiederhergestellten Basisimage.
  4. Wiederholen von Schritt 3, bis das Zielimage aus Schritt 1 zum zweiten Mal gelesen wird. Auf das Zielimage wird während einer Operation zum vollständigen inkrementellen Restore zweimal zugegriffen. Beim ersten Zugriff werden noch keine Benutzerdaten, sondern nur die Anfangsdaten aus dem Image gelesen. Erst beim zweiten Zugriff wird das Image vollständig gelesen und verarbeitet. Der zweifache Zugriff auf das Zielimage der Operation zum inkrementellen Restore soll sicherstellen, dass die Datenbank zu Beginn korrekt konfiguriert wird, d. h. mit dem richtigen Protokoll sowie den korrekten Datenbankkonfigurations- und Tabellenbereichsdefinitionen für die Datenbank, die während der Restoreoperation erstellt wird. In Fällen, in denen ein Tabellenbereich seit der Erstellung des ersten Gesamtbackup-Image der Datenbank gelöscht wurde, werden die Tabellenbereichsdaten für dieses Image zwar aus den Backup-Images gelesen, aber bei der Verarbeitung des inkrementellen Restores ignoriert.
- Inkrementelle Backup-Images können auf zwei verschiedene Weisen wiederhergestellt werden.
  - Bei einem automatischen inkrementellen Restore wird der Befehl RESTORE nur einmal abgesetzt, um das gewünschte Zielimage anzugeben. DB2 ermittelt die verbliebenen erforderlichen Backup-Images anhand des Datenbankprotokolls und stellt sie wieder her.
  - Bei einem manuellen inkrementellen Restore muss der Befehl RESTORE einmal für jedes einzelne Backup-Image abgesetzt werden, das wiederhergestellt werden muss (wie in den obigen Schritten erläutert).
- **Automatischer inkrementeller Restore - Beispiel**

  Geben Sie zum Restore einer Gruppe inkrementeller Backup-Images mit dem Befehl RESTORE DATABASE die Option TAKEN AT *zeitmarke* an. Geben Sie die Zeitmarke des Image an, das Sie zuletzt wiederherstellen wollen.

Beispiel:

```
db2 restore db sample incremental automatic taken at 20031228152133
```

Dieser Befehl veranlasst das DB2-Restoredienstprogramm, die am Anfang dieses Abschnitts aufgeführten Schritte automatisch auszuführen. Während der Anfangsphase der Verarbeitung wird das Backup-Image mit der Zeitmarke 20001228152133 gelesen, und das Restoredienstprogramm prüft, ob die Datenbank, ihr Protokoll und die Tabellenbereichsdefinitionen vorhanden und gültig sind.

Während der zweiten Verarbeitungsphase wird das Datenbankprotokoll abgefragt, um eine Kette von Backup-Images zu bilden, die zum Ausführen der angeforderten Restoreoperation erforderlich sind. Wenn dies aus einem bestimmten Grund nicht möglich ist und DB2 keine vollständige Kette der erforderlichen Images erstellen kann, wird die Restoreoperation beendet und eine Fehlernachricht zurückgegeben. In diesem Fall ist ein automatischer inkrementeller Restore nicht möglich, sodass Sie den Befehl RESTORE DATABASE mit der Option INCREMENTAL ABORT absetzen müssen. Dadurch werden alle übrigen Ressourcen bereinigt, sodass Sie mit einem manuellen inkrementellen Restore fortfahren können.

Hinweis: Es wird dringend empfohlen, die Option FORCE des Befehls PRUNE HISTORY nicht zu verwenden. Die Standardoperation dieses Befehls verhindert, dass Sie Protokolleinträge löschen, die möglicherweise für die Recovery unter Verwendung des zuletzt erstellten Gesamtbackup-Image der Datenbank erforderlich sind. Mit der Option FORCE ist es jedoch möglich, Einträge zu löschen, die für eine automatische Restoreoperation benötigt werden.

Während der dritten Verarbeitungsphase stellt DB2 alle verbliebenen Backup-Images in der generierten Kette wieder her. Wenn in dieser Phase ein Fehler auftritt, müssen Sie den Befehl RESTORE DATABASE mit der Option INCREMENTAL ABORT absetzen, um die übrigen Ressourcen zu bereinigen. Anschließend müssen Sie feststellen, ob der Fehler behoben werden kann, bevor Sie den Befehl RESTORE erneut absetzen bzw. erneut einen manuellen inkrementellen Restore versuchen.

- **Manueller inkrementeller Restore - Beispiel**

  Geben Sie zum manuellen Restore einer Gruppe inkrementeller Backup-Images mit dem Befehl RESTORE DATABASE die Option TAKEN AT zeitmarke an, und führen Sie die oben beschriebenen Schritte aus. Beispiel:

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

     Dabei verweist <*ts*> auf das letzte inkrementelle Backup-Image (Ziel-Image), für das ein Restore durchgeführt werden soll.

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

     Dabei verweist <*ts1*> auf das anfängliche vollständige Datenbank-Image (oder Tabellenbereichs-Image).

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

     Dabei verweist <*tsX*> auf alle inkrementellen Backup-Images in der Reihenfolge der Erstellung.

  4. Wiederholen Sie Schritt 3, und führen Sie für jedes inkrementelle Backup-Image einen Restore bis einschließlich des Images <*ts*> aus.

Wenn Sie eine Restoreoperation für eine Datenbank ausführen und zuvor Backup-Images von Tabellenbereichen erstellt wurden, müssen die Tabellenbereichsimages in der chronologischen Reihenfolge der Zeitmarken ihres Backups wiederhergestellt werden.

Mit dem Dienstprogramm db2ckrst können Sie das Datenbankprotokoll abfragen und eine Liste der Zeitmarken der für einen inkrementellen Restore erforderlichen Backup-Images generieren. Außerdem wird eine vereinfachte Restoresyntax für einen manuellen inkrementellen Restore generiert. Es empfiehlt sich, über alle Backups vollständig Buch zu führen und dieses Dienstprogramm nur zur Orientierung zu verwenden.

Automatischer inkrementeller Restore - Einschränkungen

1. Wenn ein Tabellenbereichsname seit dem Backup, von dem Sie einen Restore durchführen wollen, geändert wurde und Sie den neuen Namen in der Restoreoperation auf Tabellenbereichsebene verwenden, wird die erforderliche Kette von Backup-Images nicht korrekt aus dem Datenbankprotokoll generiert, sodass ein Fehler auftritt (SQL2571N).

   Beispiel:

   ```
   db2 backup db sample -> <zm1>
   db2 backup db sample incremental -> <zm2>
   db2 rename tablespace from userspace1 to t1
   db2 restore db sample tablespace ('t1') incremental automatic taken
   at <zm2>

   SQL2571N Ein automatischer inkrementeller Restore kann nicht fortgesetzt werden.
   Ursachencode: "3".
   ```

   Mögliche Lösung: Nehmen Sie einen manuellen inkrementellen Restore vor.

2. Wenn Sie eine Datenbank löschen, wird das Datenbankprotokoll gelöscht. Wenn Sie die gelöschte Datenbank wiederherstellen, wird das Datenbankprotokoll in dem Status wiederhergestellt, den es zum Zeitpunkt der Erstellung des wiederhergestellten Backup-Images hatte. Alle nach diesem Zeitpunkt erstellten Protokolleinträge gehen verloren. Wenn Sie versuchen, einen automatischen inkrementellen Restore durchzuführen, für die einige dieser verloren gegangenen Protokolleinträge benötigt würden, versucht das Dienstprogramm RESTORE, eine falsche Backup-Image-Kette zu erstellen, und gibt eine entsprechende Fehlermeldung ("falsche Reihenfolge") aus (SQL2572N).

   Beispiel:

   ```
   db2 backup db sample -> <zm1>
   db2 backup db sample incremental -> <zm2>
   db2 backup db sample incremental delta -> <zm3>
   db2 backup db sample incremental delta -> <zm4>
   db2 drop db sample
   db2 restore db sample incremental automatic taken at <zm2>
   db2 restore db sample incremental automatic taken at <zm4>
   ```

   Mögliche Lösung:
   - Nehmen Sie einen manuellen inkrementellen Restore vor.
   - Stellen Sie zuerst die Protokolldatei unter Verwendung von Image <zm4> wieder her, bevor Sie einen automatischen inkrementellen Restore starten.

3. Wenn Sie ein Backup-Image aus einer Datenbank in eine andere Datenbank wiederherstellen und anschließend ein inkrementelles Backup oder Deltabackup durchführen, können Sie zum Restore dieses Backup-Images keinen automatischen inkrementellen Restore mehr verwenden.

Beispiel:

```
db2 create db a
db2 create db b

db2 update db cfg for a using trackmod on

db2 backup db a -> zm1
db2 restore db a taken at zm1 into b

db2 backup db b incremental -> zm2

db2 restore db b incremental automatic taken at zm2

SQL2542N Es wurde keine Übereinstimmung für ein Backup-Image der Datenbankdatei
anhand des Aliasnamens der Quellendatenbank "datenbankalias" und der angegebenen
Zeitmarke "zeitmarke" gefunden.
```

Empfohlene Fehlerumgehung:

- Nehmen Sie wie folgt einen manuellen inkrementellen Restore vor:

  ```
  db2 restore db b incremental taken at zm2
  db2 restore db a incremental taken at zm1 into b
  db2 restore db b incremental taken at zm2
  ```

- Nach der manuellen Restoreoperation in Datenbank B führen Sie ein Datenbankgesamtbackup aus, um eine neue Kette von inkrementellen Backups zu beginnen.

Optimieren der Recoveryleistung

Folgendes ist im Hinblick auf die Recoveryleistung zu beachten:

- Sie können die Leistung für Datenbanken, die häufig aktualisiert werden, verbessern, indem Sie die Protokolldateien auf einer separaten Einheit speichern. In einer OLTP-Umgebung (Online Transaction Processing - Onlinetransaktionsverarbeitung) ist oftmals mehr E/A-Aufwand für das Schreiben von Daten in die Protokolle als für das Speichern einer Zeile von Daten erforderlich. Durch das Speichern der Protokolldateien auf einer separaten Einheit wird die Bewegung des Plattenzugriffsarms minimiert, die zum Wechseln zwischen einer Protokolldatei und den Datenbankdateien erforderlich ist.

  Berücksichtigen Sie dabei außerdem, welche anderen Dateien sich auf der Platte befinden. Wenn die Protokolldateien z. B. auf eine Platte versetzt werden, die auch für das System-Paging in einem System verwendet wird, das nicht über genügend Realspeicher verfügt, werden dadurch Ihre Optimierungsversuche zunichte gemacht.

  DB2 versucht automatisch, die zum Ausführen einer Backup- oder Restoreoperation benötigte Zeit zu minimieren, indem es optimale Werte für die Anzahl Puffer, die Puffergröße und die Parallelitätseinstellungen auswählt. Die Werte basieren auf der Menge des für Dienstprogramme verfügbaren Zwischenspeichers, der Anzahl verfügbarer Prozessoren und der Datenbankkonfiguration.
- Verwenden Sie mehrere Quelleneinheiten, um die zur Durchführung einer Restoreoperation benötigte Zeit zu verringern.
- Wenn eine Datenbank große Mengen von Langfeld- und LOB-Daten enthält, kann das Wiederherstellen der Datenbank sehr viel Zeit in Anspruch nehmen. Wenn für die Datenbank die aktualisierende Recovery aktiviert ist, bietet der Befehl RESTORE die Möglichkeit, ausgewählte Tabellenbereiche wiederherzustellen.

Handelt es sich bei Ihren Langfeld- und LOB-Daten um wichtige Unternehmensdaten, sollten Sie den Zeitaufwand für das Wiederherstellen dieser Tabellenbereiche gegen den für die Backup-Operation dieser Tabellenbereiche erforderlichen Zeitaufwand abwägen. Wenn die Langfeld- und LOB-Daten in separaten Tabellenbereichen gespeichert werden, lässt sich der für das Wiederherstellen von Daten erforderliche Zeitaufwand dadurch verringern, dass die Tabellenbereiche mit den Langfeld- und LOB-Daten vom Restore ausgeschlossen werden. Wenn die LOB-Daten von einer getrennten Quelle reproduziert werden können, sollten Sie beim Erstellen oder Ändern einer Tabelle zum Hinzufügen von LOB-Spalten die Option NOT LOGGED verwenden. Wenn Sie die Tabellenbereiche, die Langfeld- und LOB-Daten enthalten, nicht wiederherstellen wollen, aber die Tabellenbereiche wiederherstellen müssen, die die Tabelle enthalten, müssen Sie die aktualisierende Recovery bis zum Ende der Protokolle durchführen, sodass alle Tabellenbereiche, die die Tabellendaten enthalten, konsistent sind.

Anmerkung: Wenn Sie einen Tabellenbereich sichern, der Tabellendaten ohne die zugehörigen LONG- oder LOB-Felder enthält, können Sie keine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt für diesen Tabellenbereich durchführen. Alle Tabellenbereiche für eine Tabelle müssen gleichzeitig bis zum selben Zeitpunkt aktualisierend wiederhergestellt werden.

- Folgendes gilt für Backup- und Restoreoperationen:
  - Es sollten mehrere Einheiten verwendet werden.
  - Überlasten Sie die Bandbreite der E/A-Einheitencontroller nicht.
- DB2 verwendet zur gleichzeitigen Durchführung einer Recovery nach einem Systemabsturz und zur aktualisierenden Datenbankrecovery mehrere Agenten. Sie werden während dieser Operationen eine bessere Leistung feststellen, vor allem auf Maschinen mit symmetrischen Multiprozessoren (SMP). Die Verwendung mehrerer Agenten bei der Datenbankrecovery nutzt die Vorteile der zusätzlichen CPUs auf SMP-Maschinen.

  Der mit der parallelen Recovery eingeführte Agententyp ist 'db2agnsc'. DB2 wählt die bei der Datenbankrecovery zu verwendende Anzahl Agenten auf der Basis der vorhandenen Anzahl CPUs auf der Maschine aus.

  DB2 verteilt Protokollsätze auf diese Agenten, sodass sie nach Bedarf gleichzeitig erneut angewendet werden können. Auf diese Weise kann z. B. die Verarbeitung von Protokollsätzen für Einfügungen, Löschungen, Aktualisierungen sowie das Hinzufügen und Löschen von Schlüsseln parallel ausgeführt werden. Da die Protokollsätze auf Seitenebene parallel ausgeführt werden (Protokollsätze derselben Datenseite werden von demselben Agent verarbeitet), wird die Leistung verbessert, auch wenn alle Prozesse für dieselbe Tabelle ausgeführt werden.
- Wenn Sie eine Recoveryoperation ausführen, wählt DB2 automatisch einen optimalen Wert für die Anzahl der Puffer, die Puffergröße und die Parallelitätseinstellungen aus. Die Werte basieren auf der Menge des für Dienstprogramme verfügbaren Zwischenspeichers, der Anzahl verfügbarer Prozessoren und der Datenbankkonfiguration. Daher sollten Sie je nach der Größe der auf Ihrem System verfügbaren Speicherkapazität in Betracht ziehen, mehr Speicher zuzuordnen, indem Sie den Wert des Konfigurationsparameters UTIL_HEAP_SZ erhöhen.

Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung des Recoverydienstprogramms

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf diese zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie übergeordnete Pflege- und Dienstprogrammoperationen des Datenbankmanagers zusammenzufassen. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf die Objekte zugreifen, für die sie zugriffsberechtigt sind, d. h. für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung SYSADM, SYSCTRL oder SYSMAINT, um das Recoverydienstprogramm verwenden zu können.

Kapitel 12. Restore - Übersicht

In der einfachsten Form des DB2-Befehls RESTORE DATABASE müssen Sie lediglich den Aliasnamen der Datenbank angeben, die Sie wiederherstellen möchten. Beispiel:

```
db2 restore db sample
```

Da die Datenbank SAMPLE vorhanden ist und beim Absetzen des Befehls RESTORE DATABASE ersetzt werden wird, wird in diesem Beispiel die folgende Nachricht zurückgegeben:

```
SQL2539W  Achtung! Restore in eine bestehende Datenbank, die mit der
Datenbank des Backup-Images identisch ist. Die Datenbankdateien werden gelöscht.
Fortfahren ? (j/n)
```

Wenn Sie `j` angeben, sollte die Restoreoperation erfolgreich beendet werden können.

Für den Restore einer Datenbank ist eine exklusive Verbindung erforderlich, d. h., es können keine Anwendungen für die Datenbank ausgeführt werden, sobald die Operation gestartet ist. Das Restoredienstprogramm verhindert, dass andere Anwendungen vor der erfolgreichen Beendigung der Operation auf die Datenbank zugreifen. Der Restore eines Tabellenbereichs kann hingegen online erfolgen.

Ein Tabellenbereich kann erst nach der erfolgreichen Beendigung der Recoveryoperation (und anschließender aktualisierender Recovery) wieder verwendet werden.

Bei Tabellen, die sich über mehrere Tabellenbereiche erstrecken, ist es ratsam, die betreffende Gruppe von Tabellenbereichen zusammen zu sichern (und wiederherzustellen).

Bei der partiellen oder selektiven Restoreoperation können Sie ein Backup-Image auf Tabellenbereichsebene oder ein Image eines vollständigen Datenbank-Backups verwenden, aus dem dem Sie mindestens einen Tabellenbereich auswählen. Alle Protokolldateien, die diesen Tabellenbereichen ab dem Zeitpunkt der Erstellung des Backup-Images zugeordnet wurden, müssen vorhanden sein.

Sie können eine Datenbank aus einem Backup-Image, das auf einer 32-Bit-Ebene erstellt wurde, in einer 64–Bit-Ebene wiederherstellen, nicht jedoch umgekehrt.

Für das Backup und die Wiederherstellung Ihrer Datenbanken sollten Sie die DB2-Dienstprogramme Backup und Restore verwenden. Das Verschieben einer Dateigruppe von einer Maschine auf eine andere wird nicht empfohlen, da dies zu einer Beeinträchtigung der Datenbankintegrität führen kann.

Verwenden von Restore

Verwenden Sie den Befehl RESTORE DATABASE, um eine Datenbank oder einen Tabellenbereich nach einem Problem, zum Beispiel einem Datenträger- oder Speicherfehler, einem Stromausfall oder einem Anwendungsfehler, wiederherzustellen. Wenn Sie Ihre Datenbank oder einzelne Tabellenbereiche mit Backup gesichert haben, können Sie diese wiederherstellen, wenn sie beschädigt oder fehlerhaft sind.

Vorbereitung

Beim Restore in eine *vorhandene* Datenbank sollte noch keine Verbindung zu der wiederherzustellenden Datenbank bestehen. Das Restoredienstprogramm stellt automatisch eine Verbindung zu der angegebenen Datenbank her, die nach Abschluss der Restoreoperation beendet wird. Beim Restore in eine *neue* Datenbank ist zum Erstellen der Datenbank die Herstellung einer Verbindung zu einer Instanz (mit ATTACH) erforderlich. Beim Restore in eine *neue ferne* Datenbank müssen Sie zuerst eine Verbindung zu der Instanz herstellen, in der sich die neue Datenbank befinden soll. Erstellen Sie anschließend die neue Datenbank, indem Sie die Codepage und das Gebiet des Servers angeben. Beim Restore wird die Codepage der Zieldatenbank mit der des Backup-Images überschrieben.

Informationen zu dieser Task

Bei der Datenbank kann es sich um eine lokale oder ferne Datenbank handeln.

Für das Restoredienstprogramm gelten die folgenden Einschränkungen:

- Das Restoredienstprogramm kann nur verwendet werden, wenn die betreffende Datenbank zuvor mit dem DB2-Backup-Dienstprogramm gesichert wurde.
- Wenn Benutzer, bei denen es sich nicht um den Instanzeigner (unter UNIX) oder um Mitglieder der DB2ADMNS- oder Administratorengruppe (unter Windows) handelt, versuchen, ein Backup-Image wiederherzustellen, wird eine Fehlernachricht (SQL2061N) ausgegeben. Wenn andere Benutzer Zugriff auf das Backup-Image benötigen, müssen die Dateizugriffsrechte nach dem Erstellen des Backups geändert werden.
- Es kann keine Operation zum Restore der Datenbank gestartet werden, wenn ein Prozess zur aktualisierenden Recovery aktiv ist.
- Sie können einen Tabellenbereich nur dann in einer vorhandenen Datenbank wiederherstellen, wenn der Tabellenbereich zur Zeit vorhanden und mit dem wiederherzustellenden identisch ist. Hier bedeutet „identisch", dass der Tabellenbereich nicht zwischen der Backup- und der Restoreoperation gelöscht und erneut erstellt wurde. Bei der Datenbank auf der Platte und der Datenbank in dem Backup-Image muss es sich um die gleiche Datenbank handeln.
- Es ist nicht möglich, einen Restore auf Tabellenbereichsebene eines Tabellenbereichsbackups in einer neuen Datenbank auszuführen.
- Es es nicht möglich, eine Online-Restoreoperation auf Tabellenbereichsebene mit den Systemkatalogtabellen auszuführen.
- Sie können keinen Restore eines Backups, das in einer Umgebung mit einer Einzeldatenbankpartition erstellt wurde, in einer vorhandenen Umgebung mit partitionierten Datenbanken durchführen. Stattdessen müssen Sie das Backup in einer Umgebung mit einer Einzeldatenbankpartition wiederherstellen und anschließend die benötigten Datenbankpartitionen hinzufügen. Nach dem Restore in der neuen Umgebung mit einer Einzeldatenbankpartition müssen Sie sicherstellen, dass die Knotennummer der vorhandenen Partition in der Datei `db2nodes.cfg` mit der Knotennummer der Partition der ursprünglichen Datenbank übereinstimmt.
- Beim Restore eines Backup-Images mit einer Codepage in ein System mit einer anderen Codepage wird die Codepage des Systems von der Codepage des Backup-Images überschrieben.
- Mit dem Befehl RESTORE DATABASE können Sie keine durch nicht dynamischen Speicher aktivierten Tabellenbereiche in einen durch dynamischen Speicher aktivierten Tabellenbereich konvertieren.

Das Restoredienstprogramm kann über den Befehlszeilenprozessor (CLP), den Assistenten oder das Notizbuch 'Datenbank wiederherstellen' in der Steuerzentrale oder die Anwendungsprogrammierschnittstelle (API) db2Restore aufgerufen werden.

Das folgende Beispiel zeigt einen Befehl RESTORE DATABASE, der über den CLP abgesetzt wird:

```
db2 restore db sample from D:\DB2Backups taken at 20010320122644
```

Vorgehensweise

Gehen Sie wie folgt vor, um den Restoreassistenten zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis das Datenbank- oder Tabellenbereichsobjekt angezeigt wird, das Sie wiederherstellen möchten.
2. Klicken Sie mit der rechten Maustaste das Objekt an und wählen Sie **Wiederherstellen** im Kontextmenü aus. Der Restoreassistent wird geöffnet.

Weitere Schritte

Detaillierte Informationen können Sie über die Kontexthilfe der Steuerzentrale aufrufen.

Restore für ein Momentaufnahmebackup-Image

Bei einem Restore für ein Momentaufnahmebackup wird zum Kopieren von Daten die leistungsstarke Kopiertechnologie einer Speichereinheit genutzt.

Vorbereitung

Zur Ausführung von Backup- und Restoreoperationen für Momentaufnahmen benötigen Sie einen DB2 ACS-API-Treiber für Ihre Speichereinheit. In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS Series

Sie müssen ein Momentaufnahmebackup durchführen, bevor ein Restore für ein Momentaufnahmebackup möglich ist. Siehe hierzu „Durchführen eines Momentaufnahmebackups” auf Seite 254.

Einschränkungen

Wenn ein Momentaufnahmerestore durchgeführt wird und dabei der Datenbankpfad im Backup-Image nicht mit dem aktuellen Pfad der Datenbank auf der Platte übereinstimmt, werden alle lokalen Aliasnamen für diese Datenbank durch den Restore ungültig. Die unmittelbare Folge ist, dass diese Art des Restores nicht unter Verwendung des Aliasnamens der Datenbank durchgeführt werden kann. Die Ausweichmaßnahme besteht darin, den Restore unter Verwendung des Datenbanknamens durchzuführen und anschließend alle Aliasnamen für diese Datenbank, die vor dem Restore vorhanden waren, erneut zu erstellen. Ferne Clients sind hiervon nicht betroffen.

Vorgehensweise

Sie können den Restore für ein Momentaufnahmebackup unter Verwendung des Befehls RESTORE DATABASE mit dem Parameter **USE SNAPSHOT** oder der API db2Restore mit dem Datenträgertyp SQLU_SNAPSHOT_MEDIA durchführen. Befehl RESTORE DATABASE:

```
db2 restore db sample use snapshot
```

API db2Restore:

```
int sampleRestoreFunction( char dbAlias[],
                           char restoredDbAlias[],
                           char user[],
                           char pswd[],
                           char workingPath[] )
{
  db2MediaListStruct mediaListStruct = { 0 };

  rmediaListStruct.locations = &workingPath;
    rmediaListStruct.numLocations = 1;
  rmediaListStruct.locationType = SQLU_SNAPSHOT_MEDIA;


  db2RestoreStruct restoreStruct = { 0 };

    restoreStruct.piSourceDBAlias = dbAlias;
    restoreStruct.piTargetDBAlias = restoredDbAlias;
  restoreStruct.piMediaList = &mediaListStruct;
    restoreStruct.piUsername = user;
    restoreStruct.piPassword = pswd;
  restoreStruct.iCallerAction = DB2RESTORE_STORDEF_NOINTERRUPT;


  struct sqlca sqlca = { 0 };


  db2Restore(db2Version900, &restoreStruct, &sqlca);

  return 0;
}
```

Restore in einer vorhandenen Datenbank

Sie können einen Restore eines beliebigen Backup-Images einer Datenbank oder eines Tabellenbereichs in einer vorhandenen Datenbank durchführen. Bei einem Restore auf Datenbankebene kann sich das Backup-Image von der vorhandenen Datenbank im Hinblick auf den Aliasnamen, den Datenbanknamen oder die Datenbanknummer unterscheiden. Bei einem Restore auf Tabellenbereichsebene können Sie einen Tabellenbereich nur dann in einer vorhandenen Datenbank wiederherstellen, wenn der Tabellenbereich zurzeit vorhanden und mit dem wiederherzustellenden identisch ist. Hier bedeutet „identisch", dass der Tabellenbereich nicht zwischen der Backup- und der Restoreoperation gelöscht und erneut erstellt wurde. Bei der Datenbank auf der Platte und der Datenbank in dem Backup-Image muss es sich um die gleiche Datenbank handeln.

Eine Datenbank*nummer* ist die eindeutige Kennung einer Datenbank, die sich während der Bestehens der Datenbank nicht ändert. Diese Nummer wird beim Erstellen der Datenbank vom Datenbankmanager zugeordnet. DB2 verwendet immer die Datenbanknummer aus dem Backup-Image.

Beim Restore in eine vorhandene Datenbank führt das Restoredienstprogramm Folgendes aus:

- Löschen der Tabellen-, Index- und Langfelddaten der vorhandenen Datenbank und Ersetzen dieser Informationen durch die Daten des Backup-Images.
- Ersetzen der Tabelleneinträge der wiederherzustellenden Tabellenbereiche.
- Beibehalten der Datei des Recoveryprotokolls, sofern sie nicht beschädigt oder leer ist. Wenn die Datei des Recoveryprotokolls beschädigt ist oder keine Einträge enthält, kopiert der Datenbankmanager die Datei aus dem Backup-Image. Wenn Sie die Datei des Recoveryprotokolls ersetzen möchten, können Sie den Befehl RESTORE mit der Option REPLACE HISTORY FILE ausführen.
- Beibehalten des Authentifizierungstyps für die vorhandene Datenbank.
- Beibehalten der Datenbankverzeichnisse der vorhandenen Datenbank. Die Verzeichnisse definieren die Speicherposition und die Art der Katalogisierung der Datenbank.
- Vergleichen der Datenbanknummern. Bei unterschiedlichen Datenbanknummern:
  - Löschen der Protokolle, die der vorhandenen Datenbank zugeordnet sind.
  - Kopieren der Datenbankkonfigurationsdatei aus dem Backup-Image.
  - Setzen von NEWLOGPATH auf den Wert des Datenbankkonfigurationsparameters *logpath*, falls NEWLOGPATH im Befehl RESTORE DATABASE angegeben wurde.

  Bei identischen Datenbanknummern:
  - Löschen der Protokolle, wenn es sich um ein Image einer nicht wiederherstellbaren Datenbank handelt.
  - Beibehalten der aktuellen Datenbankkonfigurationsdatei.
  - Setzen von NEWLOGPATH auf den Wert des Datenbankkonfigurationsparameters *logpath*, falls NEWLOGPATH im Befehl RESTORE DATABASE angegeben wurde. Andernfalls wird der aktuelle Protokollpfad in die Datenbankkonfigurationsdatei kopiert. Das Dienstprogramm überprüft den Protokollpfad: Falls der Pfad von der Datenbank nicht verwendet werden kann, wird die Datenbankkonfiguration so geändert, dass sie den Standardprotokollpfad verwendet.

Restore in einer neuen Datenbank

Sie können eine neue Datenbank erstellen und anschließend das Backup-Image einer vollständigen Datenbank in diese neue Datenbank wiederherstellen. Wenn Sie keine neue Datenbank erstellen, erstellt das Restoredienstprogramm eine neue Datenbank.

Beim Restore in eine neue Datenbank führt das Restoredienstprogramm Folgendes aus:

- Erstellen einer neuen Datenbank unter Verwendung des Datenbankaliasnamens, der im Parameter für den Aliasnamen der Zieldatenbank angegeben wurde. (Wenn für die Zieldatenbank kein Aliasname angegeben wurde, erstellt das Restoredienstprogramm eine Datenbank mit einem Aliasnamen, der im Parameter für den Aliasnamen der Quellendatenbank angegeben wurde.)
- Wiederherstellen der Konfigurationsdatei der Datenbank aus dem Backup-Image.
- Setzen von NEWLOGPATH auf den Wert des Datenbankkonfigurationsparameters *logpath*, falls NEWLOGPATH im Befehl RESTORE DATABASE angegeben wurde. Das Dienstprogramm überprüft den Protokollpfad: Falls der Pfad von der Datenbank nicht verwendet werden kann, wird die Datenbankkonfiguration so geändert, dass sie den Standardprotokollpfad verwendet.
- Wiederherstellen des Authentifizierungstyps aus dem Backup-Image.

- Wiederherstellen des Inhalts aus den Datenbankverzeichnissen im Backup-Image.
- Wiederherstellen der Datei des Recoveryprotokolls für die Datenbank.
- Überschreiben der Codepage der Datenbank mit der Codepage des Backup-Image.

Verwenden des inkrementellen Restores in einer Test- und Produktionsumgebung

Wenn eine Produktionsdatenbank für inkrementelles Backup und inkrementellen Restore aktiviert wurde, können Sie ein Image eines inkrementellen Backups oder eines Delta-Backups verwenden, um eine Testdatenbank zu erstellen oder zu aktualisieren. Sie können hierzu entweder den manuellen oder den automatischen inkrementellen Restore verwenden. Zum Wiederherstellen des Backup-Images aus der Produktionsdatenbank in die Testdatenbank geben Sie mit dem Befehl RESTORE DATABASE die Option INTO *aliasname-der-zieldatenbank* an. Beispiel: In einer Produktionsdatenbank mit den folgenden Backup-Images:

```
backup db prod
Das Backup war erfolgreich. Die Zeitmarke für dieses Backup-Image ist: <ts1>

   backup db prod incremental
Das Backup war erfolgreich. Die Zeitmarke für dieses Backup-Image ist: <ts2>
```

wäre das folgende ein Beispiel für einen manuellen inkrementellen Restore:

```
restore db prod incremental taken at <zm2> into test without
prompting
DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.

restore db prod incremental taken at <zm1> into test without
prompting
DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.

restore db prod incremental taken at <zm2> into test without
prompting
DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.
```

Falls die Datenbank TEST bereits vorhanden ist, wird die Restoreoperation alle bereits vorhandenen Daten überschreiben. Wenn die Datenbank TEST noch nicht vorhanden ist, wird das Restoredienstprogramm die Datenbank erstellen und anschließend mit den Daten aus den Backup-Images füllen.

Da Operationen zum automatischen inkrementellen Restore vom Datenbankprotokoll abhängen, werden die Restoreschritte leicht abweichen, je nach dem, ob die Testdatenbank vorhanden ist oder nicht. Wenn ein automatischer inkrementeller Restore in die Datenbank TEST durchgeführt werden soll, muss deren Protokoll das Backup-Image-Protokoll für die Datenbank PROD enthalten. Das Datenbankprotokoll für das Backup-Image ersetzt alle Datenbankprotokolle, die bereits für die Datenbank TEST vorhanden sind, wenn:

- Die Datenbank TEST beim Absetzen des Befehls RESTORE DATABASE nicht vorhanden ist.
- Die Datenbank TEST beim Absetzen des Befehls RESTORE DATABASE bereits vorhanden ist und das Protokoll der Datenbank TEST keine Datensätze enthält.

Das folgende Beispiel zeigt einen automatischen inkrementellen Restore in die Datenbank TEST, die noch nicht vorhanden ist:

```
restore db prod incremental automatic taken at <zm2> into test without
prompting
DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.
```

Das Restoredienstprogramm erstellt die Datenbank TEST und füllt sie mit Daten.

Wenn die Datenbank TEST vorhanden ist und das Datenbankprotokoll Einträge enthält, müssen Sie die Datenbank vor dem automatischen inkrementellen Restore wie folgt löschen:

```
drop db test
   DB20000I  Der Befehl DROP DATABASE wurde erfolgreich ausgeführt.

   restore db prod incremental automatic taken at <zm2> into test without
   prompting
   DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.
```

Wenn Sie die Datenbank nicht löschen wollen, können Sie den Befehl PRUNE HISTORY mit einer weit in der Zukunft liegenden Zeitmarke und dem Parameter WITH FORCE OPTION verwenden, bevor Sie den Befehl RESTORE DATABASE absetzen:

```
connect to test
   Datenbankverbindungsinformationen

   Datenbankserver                 = <server-id>
   SQL-Berechtigungs-ID            = <id>
   Aliasname der lokalen Datenbank = TEST

   prune history 9999 with force option
   DB20000I  Der Befehl PRUNE wurde erfolgreich ausgeführt.

   connect reset
   DB20000I  Der SQL-Befehl wurde erfolgreich ausgeführt.
   restore db prod incremental automatic taken at <zm2> into test without
   prompting
   SQL2540W  RESTORE wurde erfolgreich ausgeführt, es wurde jedoch eine Warnung
"2539" beim  Restore der Datenbank im Modus 'No Interrupt'
festgestellt.
```

In diesem Fall arbeitet der Befehl RESTORE DATABASE genau so, wie wenn die Datenbank TEST nicht vorhanden wäre.

Ist die Datenbank TEST vorhanden und enthält das Datenbankprotokoll keine Einträge, müssen Sie die Datenbank TEST nicht löschen, bevor der automatische inkrementelle Restore durchgeführt wird:

```
   restore db prod incremental automatic taken at <zm2> into test without
   prompting
   SQL2540W  RESTORE wurde erfolgreich ausgeführt, es wurde jedoch eine Warnung
"2539" beim  Restore der Datenbank im Modus 'No Interrupt'
festgestellt.
```

Sie können weiter inkrementelle Backups oder Deltabackups der Testdatenbank erstellen, ohne zuvor ein Gesamtbackup der Datenbank vorzunehmen. Wenn es später jedoch erforderlich wird, eines der Images eines inkrementellen Backups oder eines Delta-Backups wiederzuherstellen, müssen Sie einen manuellen inkrementellen Restore durchführen. Dies liegt daran, dass Operationen zum automatischen inkrementellen Restore erfordern, dass alle während eines automatischen inkrementellen Restores wiederhergestellten Backup-Images aus demselben Datenbankaliasnamen erstellt sind.

Wenn Sie ein Datenbankgesamtbackup der Testdatenbank erstellen, nachdem Sie die Restoreoperation unter Verwendung des Produktions-Backup-Images beendet haben, können Sie inkrementelle Backups oder Deltabackups erstellen und diese entweder manuell oder im automatischen Modus wiederherstellen.

Ausführen einer umgeleiteten Restoreoperation

Eine umgeleitete Restoreoperation wird in den folgenden Situationen ausgeführt:

- Wenn Sie für ein Backup-Image ein Restore auf einer Zielmaschine durchführen möchten, die sich von der Quellenmaschine unterscheidet.
- Wenn Sie für Ihre Tabellenbereichscontainer einen Restore an einer anderen physischen Position durchführen möchten.
- Wenn Ihre Restoreoperation fehlgeschlagen ist, da auf mindestens einen Container nicht zugegriffen werden konnte.

Anmerkung: Ein umgeleiteter Restore kann nicht zum Verschieben von Daten von einem Betriebssystem auf ein anderes verwendet werden.

Während einer Operation zum umgeleiteten Restore werden Verzeichnis- und Dateicontainer automatisch erstellt, sofern sie nicht bereits vorhanden sind. Einheitencontainer werden nicht automatisch vom Datenbankmanager erstellt.

DB2 bietet nur Unterstützung für das Hinzufügen, Ändern oder Entfernen von Tabellenbereichscontainern eines DMS-Tabellenbereichs. Bei einem SMS-Tabellenbereich ist ein umgeleiteter Restore die einzige Möglichkeit, die Konfiguration für Tabellenbereichscontainer zu ändern.

Sie können Tabellenbereichscontainer erneut definieren, indem Sie den Befehl RESTORE DATABASE aufrufen und den Parameter REDIRECT angeben, oder indem Sie den Assistenten zur Wiederherstellung einer Datenbank in der Steuerzentrale verwenden. Der Prozess zum Aufrufen eines umgeleiteten Restores eines inkrementellen Backup-Images ist dem Prozess zum Aufrufen eines umgeleiteten Restores eines nicht inkrementellen Backup-Images ähnlich. Führen Sie den Befehl RESTORE DATABASE mit der Option REDIRECT auf, und geben Sie das Backup-Image an, aus dem der inkrementelle Restore der Datenbank erfolgen soll. Alternativ können Sie ein Script zum umgeleiteten Restore aus einem Backup-Image generieren und anschließend das Script nach Bedarf modifizieren. Weitere Informationen hierzu finden Sie in „Ausführen eines umgeleiteten Restores mithilfe eines automatisch generierten Scripts" auf Seite 317.

Die Umleitung von Containern bietet große Flexibilität bei der Verwaltung von Tabellenbereichscontainern. So wird beispielsweise das Hinzufügen von Containern zu SMS-Tabellenbereichen zwar nicht unterstützt, Sie könnten jedoch zu diesem Zweck bei einem umgeleiteten Restore einen zusätzlichen Container angeben.

Beispiel

Eine umgeleitete Restoreoperation besteht aus einem Restoreprozess für Datenbanken, bei dem zwei Schritte ausgeführt werden müssen, darunter ein zwischengeschalteter Definitionsschritt für Tabellenbereichscontainer:

1. Setzen Sie den Befehl RESTORE DATABASE mit der Option REDIRECT ab.
2. Verwenden Sie den Befehl SET TABLESPACE CONTAINERS, um Tabellenbereichscontainer für die wiederhergestellte Datenbank zu definieren (gibt die Tabellenbereichsposition auf dem Zielsystem an).

3. Setzen Sie den Befehl RESTORE DATABASE erneut ab, und geben Sie diesmal die Option CONTINUE an.

Das folgende Beispiel zeigt, wie für die Datenbank SAMPLE ein umgeleiteter Restore durchgeführt wird:

```
db2 restore db sample redirect without prompting
SQL1277W Zurzeit wird eine umgeleitete Restoreoperation ausgeführt.
Die Tabellenbereichskonfiguration kann jetzt angezeigt werden, und
Container für Tabellenbereiche, die keinen dynamischen Speicher
verwenden, können rekonfiguriert werden.

DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.

db2 set tablespace containers for 2 using (path 'userspace1.0', path
'userspace1.1')
DB20000I Der Befehl SET TABLESPACE CONTAINERS wurde erfolgreich ausgeführt.

db2 restore db sample continue
DB20000I  Der Befehl RESTORE DATABASE wurde erfolgreich ausgeführt.
```

Erneutes Definieren von Tabellenbereichscontainern durch Restore einer Datenbank mithilfe eines automatisch generierten Scripts

Wenn Sie eine Datenbank wiederherstellen, nimmt das Restoredienstprogramm an, dass das physische Layout der Container mit dem der Datenbank zum Zeitpunkt des Backups identisch sein soll. Wenn Sie die Position oder Größe eines der physischen Container ändern müssen, müssen Sie den Befehl RESTORE DATABASE mit der Option REDIRECT ausführen. Diese Option verlangt, dass Sie die Positionen physischer Container, die im Backup-Image gespeichert sind, angeben und vollständige Informationen zur Gruppe von Containern für jeden Tabellenbereich mit nicht dynamischem Speicher bereitstellen, den Sie ändern wollen. Sie können die Containerinformationen zum Zeitpunkt des Backups erfassen. Dies kann jedoch etwas aufwendig sein.

Zur Vereinfachung eines umgeleiteten Restores ermöglicht das Restoredienstprogramm Ihnen die Generierung eines Scripts für den umgeleiteten Restore aus einem vorhandenen Backup-Image. Dazu führen Sie den Befehl RESTORE DATABASE mit der Option REDIRECT und der Option GENERATE SCRIPT aus. Das Restoredienstprogramm untersucht das Backup-Image, extrahiert Containerinformationen aus dem Backup-Image und generiert ein CLP-Script, das sämtliche detaillierten Containerinformationen enthält. Anschließend können Sie sämtliche Pfade oder Containergrößen im Script modifizieren und das CLP-Script ausführen, um die Datenbank mit der neuen Gruppe von Containern erneut zu erstellen. Mithilfe des von Ihnen generierten Scripts können Sie eine Datenbank auch dann wiederherstellen, wenn Sie nur über ein Backup-Image verfügen und das Layout der Container nicht kennen. Das Script wird auf dem Client erstellt. Auf der Grundlage des Scripts können Sie festlegen, wo Speicherplatz für Protokolldateien und Container der wiederhergestellten Datenbank benötigt wird, und Sie können die Pfade für Protokolldateien und Container entsprechend ändern.

Das generierte Script besteht aus vier Abschnitten:

Initialisierung

Im ersten Abschnitt werden Befehlsoptionen festgelegt und die Datenbankpartitionen angegeben, in denen der Befehl auszuführen ist. Das folgende Beispiel zeigt, wie dieser erste Abschnitt aussehen kann:

```
UPDATE COMMAND OPTIONS USING S ON Z ON SAMPLE_NODE0000.out V ON;
SET CLIENT ATTACH_DBPARTITIONNUM  0;
SET CLIENT CONNECT_DBPARTITIONNUM 0;
```

Dabei gilt Folgendes:

- `S ON` gibt an, dass die Ausführung des Befehls gestoppt werden soll, wenn ein Befehlsfehler auftritt.
- `Z ON SAMPLE_NODE0000.out` gibt an, dass die Ausgabe in eine Datei mit dem Namen `<aliasname-der-datenbank>_NODE<dbpartitionnum>.out` erfolgen soll.
- `V ON` gibt an, dass der aktuelle Befehl über die Standardausgabe ausgegeben werden soll.

  Wenn das Script in einer Umgebung mit partitionierten Datenbanken ausgeführt wird, ist es wichtig, die Datenbankpartition anzugeben, in der die Scriptbefehle auszuführen sind.

Befehl RESTORE mit der Option REDIRECT

Im zweiten Abschnitt wird der Befehl RESTORE gestartet und die Option REDIRECT verwendet. In diesem Abschnitt können alle Optionen des Befehls RESTORE angegeben werden mit Ausnahme derer, die in Kombination mit der Option REDIRECT nicht verwendbar sind. Das folgende Beispiel zeigt, wie dieser zweite Abschnitt aussehen kann:

```
RESTORE DATABASE SAMPLE
   -- USER '<benutzername>'
   -- USING '<kennwort>'
   FROM '/home/jseifert/backups'
   TAKEN AT 20050906194027
   -- DBPATH ON '<zielverzeichnis>'
   INTO SAMPLE
   -- NEWLOGPATH '/home/jseifert/jseifert/NODE0000/SQL00001/SQLOGDIR/'
   -- WITH <pufferanzahl> BUFFERS
   -- BUFFER <puffergröße>
   -- REPLACE HISTORY FILE
   -- REPLACE EXISTING
REDIRECT
   -- PARALLELISM <n>
   -- WITHOUT ROLLING FORWARD
   -- WITHOUT PROMPTING
;
```

Tabellenbereichsdefinitionen

Dieser Abschnitt enthält Tabellenbereichsdefinitionen für jeden Tabellenbereich, der im Backup-Image enthalten ist oder in der Befehlszeile angegeben wird. Für jeden Tabellenbereich ist ein Abschnitt vorhanden, der aus einem Kommentarblock mit Informationen über Name, Typ und Größe des Tabellenbereichs besteht. Die Informationen sind im gleichen Format wie eine Tabellenbereichsmomentaufnahme aufbereitet. Anhand der gegebenen Informationen können Sie die erforderliche Größe für die einzelnen Tabellenbereiche bestimmen. Wird die Ausgabe zu einem mit dynamischem Speicher erstellten Tabellenbereich angezeigt, ist die Klausel SET TABLESPACE CONTAINERS in der Anzeige nicht enthalten. Das folgende Beispiel zeigt einen Abschnitt einer Tabellenbereichsdefinition:

```
-- *****************************************************************
-- ** Tablespace name                         = SYSCATSPACE
-- **   Tablespace ID                         = 0
-- **   Tablespace Type                       = System managed space
-- **   Tablespace Content Type               = Any data
-- **   Tablespace Page size (bytes)          = 4096
-- **   Tablespace Extent size (pages)        = 32
-- **   Using automatic storage               = No
```

```
-- **   Total number of pages                   = 5572
-- ****************************************************************
SET TABLESPACE CONTAINERS FOR 0
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH   'SQLT0000.0'
);
-- ****************************************************************
-- ** Tablespace name                          = TEMPSPACE1
-- **   Tablespace ID                          = 1
-- **   Tablespace Type                        = System managed space
-- **   Tablespace Content Type                = System Temporary data
-- **   Tablespace Page size (bytes)           = 4096
-- **   Tablespace Extent size (pages)         = 32
-- **   Using automatic storage                = No
-- **   Total number of pages                  = 0
-- ****************************************************************
SET TABLESPACE CONTAINERS FOR 1
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  PATH   'SQLT0001.0'
);
-- ****************************************************************
-- ** Tablespace name                          = DMS
-- **   Tablespace ID                          = 2
-- **   Tablespace Type                        = Database managed space
-- **   Tablespace Content Type                = Any data
-- **   Tablespace Page size (bytes)           = 4096
-- **   Tablespace Extent size (pages)         = 32
-- **   Using automatic storage                = No
-- **   Auto-resize enabled                    = No
-- **   Total number of pages                  = 2000
-- **   Number of usable pages                 = 1960
-- **   High water mark (pages)                = 96
-- ****************************************************************
SET TABLESPACE CONTAINERS FOR 2
-- IGNORE ROLLFORWARD CONTAINER OPERATIONS
USING (
  FILE   '/tmp/dms1'                                          1000
, FILE   '/tmp/dms2'                                          1000
);
```

Befehl RESTORE mit der Option CONTINUE

Im letzten Abschnitt wird der Befehl RESTORE mit der Option CONTINUE abgesetzt, um den umgeleiteten Restore abzuschließen. Das folgende Beispiel zeigt, wie dieser letzte Abschnitt aussehen kann:

```
RESTORE DATABASE SAMPLE CONTINUE;
```

Ausführen eines umgeleiteten Restores mithilfe eines automatisch generierten Scripts

Wenn Sie eine umgeleitete Restoreoperation durchführen, müssen Sie die Positionen physischer Container angeben, die im Backup-Image gespeichert sind, und vollständige Informationen zur Gruppe von Containern für jeden Tabellenbereich bereitstellen, den Sie ändern wollen. Verwenden Sie die folgende Prozedur, um ein Script für den umgeleiteten Restore auf der Basis eines vorhandenen Backup-Images zu generieren, das Script zu modifizieren und es anschließend zur Durchführung der umgeleiteten Restoreoperation auszuführen.

Sie können einen umgeleiteten Restore nur dann ausführen, wenn die Datenbank zuvor mit dem DB2-Backup-Dienstprogramm gesichert wurde.

- Wenn die Datenbank vorhanden ist, müssen Sie eine Verbindung zu ihr herstellen können, um das Script zu generieren. Wenn daher die Datenbank eine Migration oder eine Recovery nach Systemabsturz erfordert, muss diese ausgeführt werden, bevor Sie versuchen, ein Script für den umgeleiteten Restore zu generieren.
- Wenn Sie in einer Umgebung mit partitionierten Datenbanken arbeiten und die Zieldatenbank nicht vorhanden ist, können Sie den Befehl zur Generierung des Scripts für den umgeleiteten Restore nicht gleichzeitig in allen Datenbankpartitionen ausführen. Stattdessen muss der Befehl zur Generierung des Scripts für den umgeleiteten Restore jeweils nur in einer Datenbankpartition gleichzeitig, beginnend mit der Katalogpartition, ausgeführt werden.

  Alternativ können Sie zuerst eine Pseudodatenbank mit demselben Namen wie Ihre Zieldatenbank erstellen. Nach der Erstellung der Pseudodatenbank können Sie das Script für den umgeleiteten Restore in allen Datenbankpartitionen gleichzeitig generieren.
- Auch wenn Sie die Option REPLACE EXISTING bei der Ausführung des Befehls RESTORE zur Generierung des Scripts angeben, wird die Option REPLACE EXISTING in dem Script auf Kommentar gesetzt.
- Aus Sicherheitsgründen wird Ihr Kennwort nicht in das generierte Script eingetragen. Sie müssen das Kennwort manuell einfügen.
- Sie können das Script für einen umgeleiteten Restore nicht über den Restoreassistenten in der Steuerzentrale generieren.

Gehen Sie zur Durchführung eines umgeleiteten Restores mithilfe eines Scripts wie folgt vor:

1. Generieren Sie mithilfe des Restoredienstprogramms ein Script für den umgeleiteten Restore. Das Restoredienstprogramm kann über den Befehlszeilenprozessor (CLP) oder die Anwendungsprogrammierschnittstelle **db2Restore** aufgerufen werden. Das folgende Beispiel zeigt einen Befehl RESTORE DATABASE mit der Option REDIRECT und der Option GENERATE SCRIPT:

   ```
   db2 restore db test from /home/jseifert/backups taken at 20050304090733
                  redirect generate script test_node0000.clp
   ```

   Dieser Befehl erstellt ein Script für den umgeleiteten Restore auf dem Client mit dem Namen `test_node0000.clp`.
2. Öffnen Sie das Script für den umgeleiteten Restore in einem Texteditor, um alle erforderlichen Modifikationen vorzunehmen. Folgende Angaben können modifiziert werden:
   - RESTORE-Optionen
   - Pfade für dynamischen Speicher
   - Layout und Pfade von Containern
3. Führen Sie das modifizierte Script für den umgeleiteten Restore aus. Beispiel:

   ```
   db2 -tvf test_node0000.clp
   ```

Erneutes Erstellen einer Datenbank

Unter der erneuten Erstellung (Rebuild) einer Datenbank ist der Prozess der Wiederherstellung einer Datenbank bzw. einer Teilgruppe ihrer Tabellenbereiche durch eine Reihe von Restoreoperationen zu verstehen. Die zur erneuten Datenbankerstellung bereitgestellte REBUILD-Funktionalität macht DB2 zu einem leistungsfähigeren und vielseitigeren Produkt, das Ihnen eine komplettere Recoverylösung bietet.

Die Fähigkeit, eine Datenbank aus Backup-Images von Tabellenbereichen erneut erstellen zu können, bedeutet, dass Sie Datenbankgesamtbackups nicht mehr so häufig zu erstellen brauchen. Bei zunehmenden Datenbankgrößen finden sich immer seltener geeignete Gelegenheiten zur Erstellung von Datenbankgesamtbackups. Die alternative Verwendung von Tabellenbereichsbackups bietet die Möglichkeit, mit weniger häufigen Datenbankgesamtbackups zu arbeiten. Stattdessen können Sie häufiger Tabellenbereichsbackups erstellen und diese zusammen mit Protokolldateien in Ihrem Plan für den Einsatz bei einem Ausfall vorsehen.

Wenn Sie im Recoveryfall eine Teilgruppe von Tabellenbereichen schneller als andere wieder online verfügbar machen müssen, können Sie die REBUILD-Funktionalität zu diesem Zweck nutzen. Die Möglichkeit, nur eine Teilgruppe von Tabellenbereichen online verfügbar zu machen, ist insbesondere in Test- und Produktionsumgebungen nützlich.

Die erneute Erstellung einer Datenbank beinhaltet eine potenziell größere Anzahl von Restoreoperationen. Eine REBUILD-Operation kann ein Datenbankimage oder Tabellenbereichsimage (oder beides) verwenden. Sie kann Gesamtbackups oder inkrementelle Backups (oder beides) verwenden. Die einleitende Restoreoperation stellt das Zielimage wieder her, das die Struktur der Datenbank definiert, die wiederhergestellt werden kann (z. B. die Tabellenbereichsgruppe und die Datenbankkonfiguration). Bei wiederherstellbaren Datenbanken ermöglicht die REBUILD-Operation die Erstellung einer Datenbank, zu der eine Verbindung herstellbar ist und die eine Teilgruppe der Tabellenbereiche enthält, die Sie online benötigen, während Tabellenbereiche, die später offline wiederhergestellt werden können, zunächst unverändert beibehalten werden.

Die Methode, die Sie zur erneuten Erstellung Ihrer Datenbank verwenden, hängt davon ab, ob sie wiederherstellbar oder nicht wiederherstellbar ist.

- Wenn die Datenbank wiederherstellbar ist, verwenden Sie eine der folgenden Methoden:
  - Führen Sie mithilfe eines Images eines Gesamtbackups oder eines inkrementellen Backup-Images einer Datenbank oder von Tabellenbereichen als Ziel die REBUILD-Operation für Ihre Datenbank aus, indem Sie den Tabellenbereich SYSCATSPACE und alle anderen Tabellenbereiche aus dem Zielimage nur mit der Option REBUILD im Befehl RESTORE wiederherstellen. Anschließend können Sie für Ihre Datenbank eine aktualisierende Recovery (ROLLFORWARD) bis zu einem bestimmten Zeitpunkt ausführen.
  - Führen Sie mithilfe eines Images eines Gesamtbackups oder eines inkrementellen Backup-Images einer Datenbank oder von Tabellenbereichen als Ziel die REBUILD-Operation für Ihre Datenbank aus, indem Sie in der Option REBUILD die Gruppe von wiederherzustellenden Tabellenbereichen angeben, die in der Datenbank zum Zeitpunkt des Zielimages definiert waren. Der Tabellenbereich SYSCATSPACE muss zu dieser Gruppe gehören. Durch diese Operation werden die Tabellenbereiche wiederhergestellt, die im Zielimage definiert sind. Anschließend werden mithilfe der Datei des Recoveryprotokolls alle anderen erforderlichen Backup-Images für die verbleibenden Tabellenbereiche, die sich nicht im Zielimage befinden, gesucht und wiederhergestellt. Wenn diese Restoreoperationen abgeschlossen sind, führen Sie eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt aus.
- Wenn die Datenbank nicht wiederherstellbar ist:
  - Führen Sie mithilfe eines Images eines Gesamtbackups oder eines inkrementellen Backup-Images einer Datenbank als Ziel die REBUILD-Operation für Ihre Datenbank aus, indem Sie den Tabellenbereich SYSCATSPACE und alle anderen Tabellenbereiche aus dem Zielimage mit der entsprechenden RE-

BUILD-Syntax im Befehl RESTORE wiederherstellen. Wenn die Restoreoperation abgeschlossen ist, können Sie eine Verbindung zur Datenbank herstellen.

Angeben des Zielimages

Zur Ausführung einer erneuten Erstellung einer Datenbank beginnen Sie, indem Sie den Befehl RESTORE unter Angabe des aktuellsten Backup-Images absetzen, das Sie als Ziel der Restoreoperation verwenden. Dieses Image wird als Zielimage der REBUILD-Operation bezeichnet, weil es die Struktur der wiederherzustellenden Datenbank einschließlich der Tabellenbereiche, die wiederherstellbar sind, der Datenbankkonfiguration und der Protokollfolge definiert. Das Zielimage für die Rebuildoperation wird mit dem Parameter TAKEN AT im Befehl RESTORE DATABASE angegeben. Bei dem Zielimage kann es sich um jeden Typ von Backup (Gesamtbackup, Tabellenbereichsbackup, inkrementelles, Online- oder Offline-Backup) handeln. Zur erneuten Erstellung der Datenbank sind alle die Tabellenbereiche verfügbar, die in der Datenbank zum Zeitpunkt der Erstellung des Zielimages definiert waren.

Sie müssen die Tabellenbereiche, die wiederhergestellt werden sollen, mit einer der folgenden Methoden angeben:
- Geben Sie an, dass alle in der Datenbank definierten Tabellenbereiche wiederherzustellen sind, und geben Sie eine Ausnahmeliste an, falls Tabellenbereiche ausgeschlossen werden sollen.
- Geben Sie an, dass alle Tabellenbereiche wiederherzustellen sind, die Benutzerdaten im Zielimage haben, und geben Sie eine Ausnahmeliste an, falls Tabellenbereiche ausgeschlossen werden sollen.
- Geben Sie die Liste der in der Datenbank definierten Tabellenbereiche an, die wiederhergestellt werden sollen.

Wenn Sie die Tabellenbereiche kennen, welche die erneut erstellte Datenbank enthalten soll, führen Sie den Befehl RESTORE mit der entsprechenden REBUILD-Option aus, indem Sie das zu verwendende Zielimage angeben.

Rebuildphase

Wenn der Befehl RESTORE mit der entsprechenden REBUILD-Option abgesetzt und das Zielimage erfolgreich wiederhergestellt wurde, wird die Datenbank als in der Rebuildphase befindlich betrachtet. Nach der Restoreoperation des Zielimages stellen alle weiteren Restoreoperationen für Tabellenbereiche Daten in vorhandenen Tabellenbereichen, so wie sie in der erneut erstellten Datenbank definiert sind, wieder her. Anschließend werden diese Tabellenbereiche bei Abschluss der Rebuildoperation zusammen mit der Datenbank aktualisierend wiederhergestellt (ROLLFORWARD).

Wenn Sie den Befehl RESTORE mit der entsprechenden REBUILD-Option absetzen und die Datenbank nicht vorhanden ist, wird eine neue Datenbank auf der Basis der Attribute im Zielimage erstellt. Ist die Datenbank vorhanden, empfangen Sie eine Warnung, die besagt, dass die Phase der Neuerstellung (Rebuildphase) gestartet wird. Sie werden zur Angabe aufgefordert, ob die Rebuildoperation fortgesetzt werden soll oder nicht.

Die Rebuildoperation stellt alle Anfangsmetadaten aus dem Zielimage wieder her. Dies schließt auch alle Daten mit ein, die zur Datenbank gehören, jedoch nicht zu den Tabellenbereichsdaten oder Protokolldateien. Solche Anfangsmetadaten sind zum Beispiel:
- Tabellenbereichsdefinitionen

- Die Verlaufsdatei, bei der es sich um eine Datenbankdatei handelt, in der Verwaltungsoperationen aufgezeichnet werden

Die Rebuildoperation stellt außerdem die Datenbankkonfiguration wieder her. Das Zielimage legt die Protokollkette fest, die bestimmt, welche Images für die verbleibenden Restores in der Rebuildphase verwendet werden können. Nur Images in derselben Protokollkette sind verwendbar.

Wenn eine Datenbank bereits auf Platte vorhanden ist und die Verlaufsdatei aus dem Zielimage genommen werden soll, müssen Sie die Option REPLACE HISTORY FILE angeben. Die zu diesem Zeitpunkt auf Platte befindliche Verlaufsdatei wird durch die automatische Logik verwendet, um die verbleibenden Images zu suchen, die zur erneuten Erstellung der Datenbank erforderlich sind.

Wenn das Zielimage wiederhergestellt wurde, geschieht Folgendes:

- Wenn die Datenbank wiederherstellbar ist, wird sie in den Status *aktualisierende Recovery anstehend* versetzt und alle Tabellenbereiche, die Sie mit RESTORE wiederherstellen, werden ebenfalls in den Status *aktualisierende Recovery anstehend* versetzt. Alle Tabellenbereiche, die in der Datenbank definiert sind, jedoch nicht wiederhergestellt wurden, werden in den Status *Restore anstehend* versetzt.
- Wenn die Datenbank nicht wiederherstellbar ist, werden die Datenbank und die Tabellenbereiche, die wiederhergestellt wurden, in den Normalstatus versetzt. Alle Tabellenbereiche, die nicht wiederhergestellt wurden, werden in den Status *Löschen anstehend* versetzt, da sie nicht mehr zurückgeschrieben werden können. Für diesen Typ von Datenbank ist damit die Rebuildphase abgeschlossen.

Für wiederherstellbare Datenbanken endet die Rebuildphase, wenn der erste Befehl ROLLFORWARD DATABASE absetzt wird und das Dienstprogramm zur aktualisierenden Recovery mit der Verarbeitung von Protokollsätzen beginnt. Wenn eine aktualisierende Recoveryoperation nach dem Start der Verarbeitung von Protokollsätzen fehlschlägt und als Nächstes eine Restoreoperation abgesetzt wird, wird die Restoreoperation nicht als Teil der Rebuildphase interpretiert. Solche Restoreoperationen sollten als normale Tabellenbereichswiederherstellungen betrachtet werden, die nicht zur Rebuildphase gehören.

Automatische Verarbeitung

Nach dem Restore des Zielimages stellt das Restoredienstprogramm fest, ob verbleibende Tabellenbereiche vorhanden sind, die wiederherzustellen sind. Wenn dies der Fall ist, werden sie über dieselbe Verbindung wiederhergestellt, die zur Ausführung des Befehls RESTORE DATABASE mit der Option REBUILD verwendet wurde. Das Dienstprogramm verwendet die Verlaufsdatei auf Platte, um die aktuellsten Backup-Images zu finden, die vor dem Zielimage erstellt wurden, das alle verbleibenden wiederherzustellenden Tabellenbereiche enthält. Das Restoredienstprogramm verwendet die Positionsdaten der Backup-Images, die in der Verlaufsdatei gespeichert sind, um jedes einzelne dieser Images automatisch wiederherzustellen. Diese nachfolgenden Restoreoperationen, die auf Tabellenbereichsebene erfolgen, können nur offline ausgeführt werden. Wenn das ausgewählte Image nicht zur aktuellen Protokollkette gehört, wird ein Fehler zurückgegeben. Jeder Tabellenbereich, der aus diesem Image wiederhergestellt wird, wird in den Status *aktualisierende Recovery anstehend* versetzt.

Das Restoredienstprogramm versucht, alle erforderlichen Tabellenbereiche automatisch wiederherzustellen. In einigen Fällen ist es jedoch aufgrund von Problemen mit der Verlaufsdatei nicht in der Lage, einige Tabellenbereiche wiederherzustellen,

oder es tritt ein Fehler beim Restore eines der erforderlichen Images auf. In einem solchen Fall können Sie die Rebuildoperation entweder manuell zu Ende führen oder das Problem beheben und die Rebuildoperation erneut ausführen.

Wenn die automatische erneute Erstellung nicht erfolgreich abgeschlossen werden kann, schreibt das Restoredienstprogramm alle Informationen in das Diagnoseprotokoll (db2diag.log), die es für die verbleibenden Restoreschritte gesammelt hat. Mithilfe dieser Informationen können Sie die erneute Erstellung manuell zu Ende führen.

Wenn eine Datenbank erneut erstellt wird, werden nur Container, die zu den Tabellenbereichen gehören, die Teil des Rebuildprozesses sind, angefordert.

Wenn Container durch einen umgeleiteten Restore umdefiniert werden müssen, müssen Sie den neuen Pfad und die Größe des neuen Containers für die verbleibenden Restoreoperationen und die nachfolgende aktualisierende Recoveryoperation festlegen.

Wenn die Daten für einen Tabellenbereich, der aus einem dieser verbleibenden Images wiederhergestellt wird, nicht in die Definitionen des neuen Containers passt, wird der Tabellenbereich in den Status *Restore anstehend* versetzt und am Ende der Restoreoperation eine Warnung ausgegeben. Weitere Informationen zu dem aufgetretenen Problem finden Sie im Diagnoseprotokoll.

Abschließen der Rebuildphase

Wenn alle vorgesehenen Tabellenbereiche wiederhergestellt wurden, haben je nach Konfiguration der Datenbank zwei Optionen. Wenn die Datenbank nicht wiederherstellbar ist, kann eine Verbindung zu dieser Datenbank hergestellt werden und alle wiederhergestellten Tabellenbereiche werden online verfügbar. Alle Tabellenbereiche, die sich im Status Löschen anstehend befinden, können nicht mehr wiederhergestellt werden und sollten gelöscht werden, wenn geplant ist, in Zukunft Backups für die Datenbank auszuführen.

Wenn die Datenbank wiederherstellbar ist, können Sie den Befehl ROLLFORWARD ausführen, um die Tabellenbereiche, die wiederhergestellt wurden, online verfügbar zu machen. Wenn der Tabellenbereich SYSCATSPACE nicht wiederhergestellt wurde, schlägt die aktualisierende Recovery fehl, sodass dieser Tabellenbereich wiederhergestellt werden muss, bevor die aktualisierende Recoveryoperation gestartet werden kann. Dies bedeutet, dass der Tabellenbereich SYSCATSPACE während der Rebuildphase wiederhergestellt werden muss.

Anmerkung: In einer Umgebung mit partitionierten Datenbanken ist der Tabellenbereich SYSCATSPACE in Nichtkatalogpartitionen nicht vorhanden, sodass er dort nicht erneut erstellt werden kann. Allerdings muss der Tabellenbereich SYSCATSPACE in der Katalogpartition einer der Tabellenbereiche sein, die erneut erstellt werden, da ansonsten die aktualisierende Recoveryoperation nicht erfolgreich ausgeführt werden kann.

Die aktualisierende Recovery der Datenbank nimmt die Datenbank aus dem Status *aktualisierende Recovery anstehend* heraus und stellt alle Tabellenbereiche im Status *aktualisierende Recovery anstehend* aktualisierend wieder her. Das Dienstprogramm für aktualisierende Recovery (Rollforward) arbeitet nicht mit Tabellenbereichen im Status *Restore anstehend*.

Die Stoppzeit der aktualisierenden Recoveryoperation muss ein Zeitpunkt sein, der nach dem Ende des aktuellsten Backup-Images liegt, das in der Rebuildphase wiederhergestellt wird. Wenn ein anderer Zeitpunkt angegeben wird, tritt ein Fehler auf. Wenn die aktualisierende Recoveryoperation die Backupzeit des ältesten Images, das wiederhergestellt wurde, nicht erreichen kann, ist das Dienstprogramm für aktualisierende Recovery nicht in der Lage, die Datenbank bis zu einem konsistenten Punkt wiederherzustellen, sodass die Operation fehlschlägt.

Sie müssen sämtliche Protokolldateien für den Zeitraum zwischen dem frühesten und dem aktuellsten Backup-Image für die Verwendung durch das Dienstprogramm für aktualisierende Recovery zur Verfügung haben. Die erforderlichen Protokolle sind die Protokolle, die auf die Protokollkette aus dem frühesten Backup-Image bis zu dem Ziel-Backup-Image folgen, wie sie durch den Abschneidebereich im Zielimage definiert sind. Ansonsten schlägt die aktualisierende Recoveryoperation fehl. Wenn während der Rebuildphase irgendwelche Backup-Images wiederhergestellt wurden, die aktueller waren als das Zielimage, sind die zusätzlichen Protokolle vom Zielimage bis zum aktuellsten wiederhergestellten Backup-Image erforderlich. Wenn die Protokolle nicht verfügbar gemacht werden, werden die durch die Protokolle nicht erreichten Tabellenbereiche von der aktualisierenden Recoveryoperation in den Status *Restore anstehend* versetzt. Sie können den Befehl LIST HISTORY ausführen, um den RESTORE-REBUILD-Eintrag mit dem Protokollbereich anzuzeigen, der für die aktualisierende Recovery erforderlich ist.

Die korrekten Protokolldateien müssen verfügbar sein. Wenn das Dienstprogramm für aktualisierende Recovery die Protokolle abrufen soll, müssen Sie sicherstellen, dass der DB2-Protokollmanager so konfiguriert ist, dass er die Position angibt, von der die Protokolldateien abgerufen werden können. Wenn der Protokollpfad oder der Archivierungspfad geändert wurden, müssen Sie die Option OVERFLOW LOG PATH des Befehls ROLLFORWARD DATABASE verwenden.

Verwenden Sie die Option AND STOP des Befehls ROLLFORWARD DATABASE, um die Datenbank verfügbar zu machen, wenn der ROLLFORWARD-Befehl erfolgreich beendet wird. An diesem Punkt befindet sich die Datenbank nicht mehr im Status *aktualisierende Recovery anstehend*. Wenn die aktualisierende Recoveryoperation beginnt, jedoch vor dem erfolgreichen Abschluss ein Fehler auftritt, stoppt die aktualisierende Recoveryoperation am Fehlerpunkt und gibt eine Fehlernachricht zurück. Die Datenbank verbleibt im Status *aktualisierende Recovery anstehend*. Sie müssen Schritte zur Behebung des Problems (z. B. Korrektur der Protokolldatei) ausführen und einen weiteren ROLLFORWARD-Befehl absetzen, um die Verarbeitung fortzusetzen. Wenn sich der Fehler nicht beheben lässt, können Sie die Datenbank am Fehlerpunkt verfügbar machen, indem Sie den Befehl ROLLFORWARD STOP ausführen. Nach Verwendung der Option STOP sind alle eventuell über diesen Punkt hinaus vorhandenen Protokolldaten nicht mehr verfügbar. Die Datenbank wird an diesem Punkt wieder verfügbar, und alle Tabellenbereiche, die wiederhergestellt wurden, werden online zugänglich. Tabellenbereiche, die noch nicht wiederhergestellt wurden, befinden sich im Status *Restore anstehend*. Die Datenbank selbst befindet sich im Normalstatus.

Sie müssen entscheiden, welches die beste Methode zur Wiederherstellung der im Status *Restore anstehend* verbleibenden Tabellenbereiche ist. Dies könnte eine neue Restore- und aktualisierende Recoveryoperation für einen Tabellenbereich oder auch eine Wiederholung der gesamten Operation zur erneuten Erstellung sein. Diese Entscheidung hängt vom Typ der aufgetretenen Probleme ab. In Fällen, in denen der Tabellenbereich SYSCATSPACE zu den Tabellenbereichen im Status *Restore anstehend* gehört, kann keine Verbindung zur Datenbank hergestellt werden.

Erneutes Erstellen und Tabellenbereichscontainer

Bei einer erneuten Erstellung werden nur die Container angefordert, deren Tabellenbereiche Teil des Rebuildprozesses sind. Die Container, die zu den einzelnen Tabellenbereichen gehören, werden angefordert, wenn die Benutzerdaten der Tabellenbereiche aus einem Image wiederhergestellt werden.

Nach dem Restore des Zielimages sind nur die Definitionen für jeden Tabellenbereich, der der Datenbank zum Zeitpunkt des Backups bekannt war, wiederhergestellt. Dies bedeutet, dass der Datenbank, die durch den Rebuildprozess erstellt wird, die gleichen Tabellenbereiche bekannt sind, die sie zum Zeitpunkt des Backups hatte. Für diejenigen Tabellenbereiche, für die auch die Benutzerdaten aus dem Zielimage wiederhergestellt werden sollten, wurden zu diesem Zeitpunkt auch die zugehörigen Container angefordert.

Für alle verbleibenden Tabellenbereiche, die durch direkte Restoreoperationen für diese Tabellenbereiche wiederhergestellt werden, werden die zugehörigen Container angefordert, wenn das Image wiederhergestellt wird, das die Daten der Tabellenbereiche enthält.

Erneutes Erstellen mit umgeleitetem Restore

Im Fall eines umgeleiteten Restores müssen alle Tabellenbereichscontainer beim Restore des Zielimages definiert werden. Wenn Sie die Option REDIRECT angeben, wird die Steuerung an Sie zurückgegeben, sodass Sie Ihre Tabellenbereichscontainer neu definieren können. Wenn Sie Tabellenbereichscontainer mithilfe des Befehls SET TABLESPACE CONTAINERS erneut definiert haben, werden diese neuen Container zu diesem Zeitpunkt angefordert. Jeder Tabellenbereichscontainer, der von Ihnen nicht neu definiert wird, wird normal angefordert, wenn die Benutzerdaten des Tabellenbereichs aus einem Image wiederhergestellt werden.

Wenn die Daten für einen Tabellenbereich, der wiederhergestellt wird, nicht in die neuen Containerdefinitionen passen, wird der Tabellenbereich in den Status *Restore anstehend* versetzt und am Ende der Restoreoperation eine Warnung an Sie zurückgegeben. Im DB2-Diagnoseprotokoll finden Sie in diesem Fall eine Nachricht mit Zusatzinformationen zu dem Problem.

Erneutes Erstellen und Tabellenbereiche für temporäre Tabellen

Im Allgemeinen besteht ein DB2-Backup-Image aus den folgenden Komponenten:

- Anfangsmetadaten für die Datenbank, wie zum Beispiel Tabellenbereichsdefinitionen, die Datenbankkonfigurationsdatei und die Verlaufsdatei.
- Daten für nicht temporäre Tabellenbereiche, die dem Dienstprogramm BACKUP angegeben wurden
- Endmetadaten für die Datenbank, wie zum Beispiel der Protokolldateiheader
- Protokolldateien (wenn die Option INCLUDE LOGS angegeben wurde)

In jedem Backup-Image, unabhängig davon, ob es sich um ein Datenbank- oder Tabellenbereichsbackup, ein Gesamtbackup oder ein inkrementelles (Delta-)Backup handelt, sind in jedem Fall diese Kernkomponenten enthalten.

Ein Backup-Image einer Datenbank enthält alle der oben aufgeführten Komponenten sowie Daten für alle in der Datenbank zum Zeitpunkt des Backups definierten Tabellenbereiche.

Ein Backup-Image von Tabellenbereichen enthält stets die oben aufgeführten Datenbankmetadaten, darüber hinaus jedoch nur die Daten der Tabellenbereiche, die bei der Ausführung des Backup-Dienstprogramms angegeben wurden.

Tabellenbereiche für temporäre Tabellen werden anders als Tabellenbereiche für nicht temporäre Tabellen behandelt. Die Daten in Tabellenbereichen für temporäre Tabellen werden in keinem Fall durch ein Backup gesichert, jedoch spielt ihr Vorhandensein für die Rahmendefinition der Datenbank eine wichtige Rolle. Obwohl die Daten solcher Tabellenbereiche selbst nie gesichert werden, werden die Tabellenbereiche für temporäre Tabellen als Teil der Datenbank betrachtet. Daher werden sie in den Metadaten, die zusammen mit einem Backup-Image gespeichert werden, speziell markiert. Dadurch sieht es so aus, als ob sie sich im Backup-Image befänden. Darüber hinaus enthalten die Tabellenbereichsdefinitionen Informationen zum Vorhandensein aller Tabellenbereiche für temporäre Tabellen.

Obwohl kein Backup-Image Daten eines Tabellenbereichs für temporäre Tabellen enthält, werden bei der erneuten Erstellung (Rebuild), wenn das Zielimage wieder hergestellt wird, unabhängig vom Typ des Backup-Image Tabellenbereiche für temporäre Tabellen nur insofern wiederhergestellt, als dass ihre Container angefordert und zugeordnet werden. Die Anforderung und Zuordnung von Containern erfolgt automatisch im Rahmen der Rebuildverarbeitung. Infolgedessen können beim erneuten Erstellen einer Datenbank Tabellenbereiche für temporäre Tabellen nicht ausgeschlossen werden.

Auswählen eines Zielimages für die erneute Erstellung einer Datenbank

Das Zielimage für die erneute Erstellung sollte das aktuellste Backup-Image sein, das Sie als Ausgangspunkt für Ihre Restoreoperation verwenden möchten. Dieses Image wird als Zielimage der REBUILD-Operation bezeichnet, weil es die Struktur der wiederherzustellenden Datenbank einschließlich der Tabellenbereiche, die wiederherstellbar sind, der Datenbankkonfiguration und der Protokollfolge definiert. Es kann sich um jeden Typ von Backup (Gesamtbackup, Tabellenbereichsbackup, inkrementelles, Online- oder Offline-Backup) handeln.

Das Zielimage legt die Protokollfolge (oder Protokollkette) fest, die bestimmt, welche Images für die verbleibenden Restores in der Rebuildphase verwendet werden können. Nur Images in derselben Protokollkette sind verwendbar.

Die folgenden Beispiele veranschaulichen, wie das Image ausgewählt wird, das als Zielimage für die Rebuildoperation zu verwenden ist.

Nehmen Sie an, eine Datenbank mit dem Namen SAMPLE enthält die folgenden Tabellenbereiche:

- SYSCATSPACE (Systemkataloge)
- USERSP1 (Tabellenbereich für Benutzerdaten)
- USERSP2 (Tabellenbereich für Benutzerdaten)
- USERSP3 (Tabellenbereich für Benutzerdaten)

Abb. 23 zeigt die folgenden Backups auf Datenbankebene und Backups auf Tabellenbereichsebene, die erstellt wurden, in chronologischer Reihenfolge:

1. Datenbankgesamtbackup DB1
2. Tabellenbereichsgesamtbackup TS1
3. Tabellenbereichsgesamtbackup TS2
4. Tabellenbereichsgesamtbackup TS3
5. Restore der Datenbank und aktualisierende Recovery bis zu einem Zeitpunkt zwischen TS1 und TS2
6. Tabellenbereichsgesamtbackup TS4
7. Tabellenbereichsgesamtbackup TS5

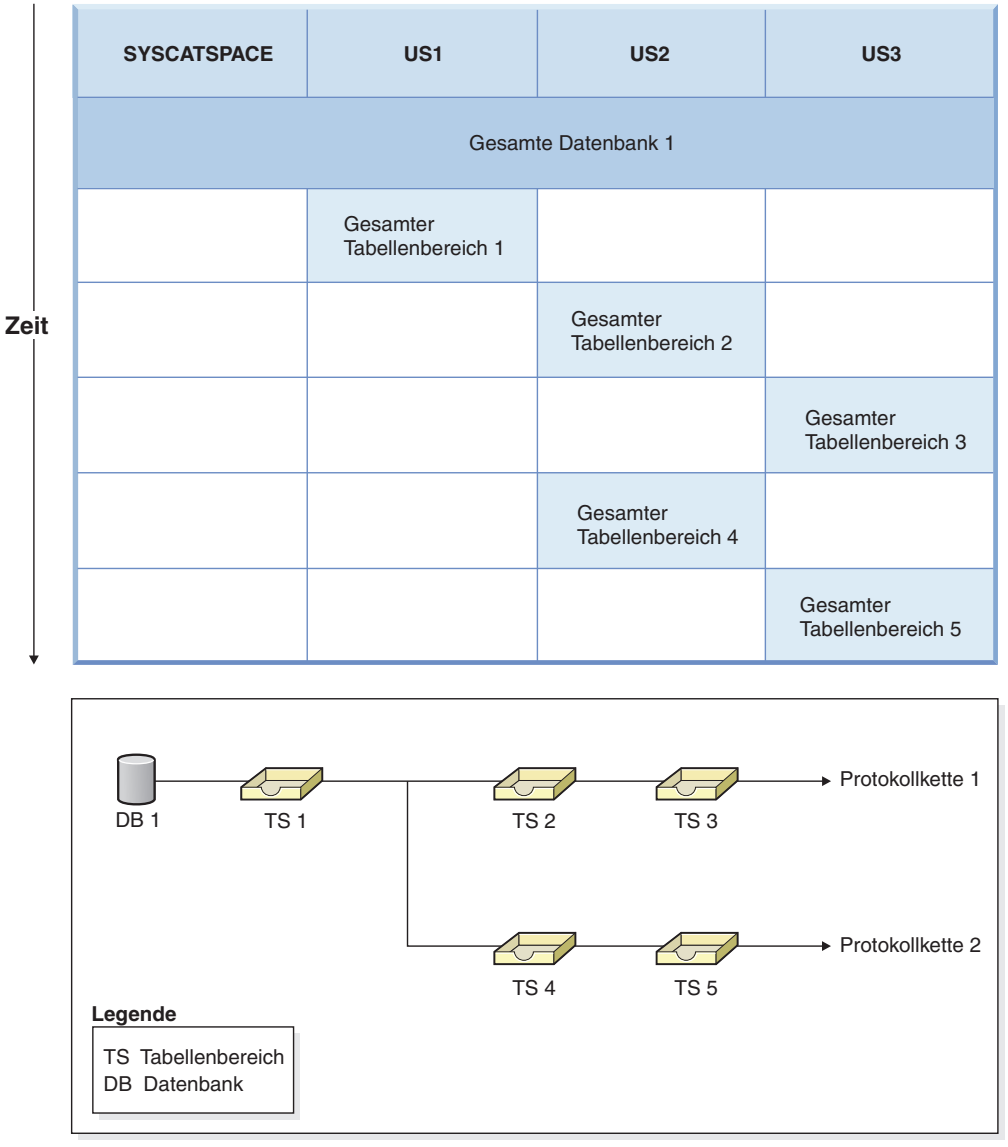


Abbildung 23. Backups auf Datenbank- und Tabellenbereichsebene der Datenbank SAMPLE

Beispiel 1

Das folgende Beispiel veranschaulicht die Befehle des Befehlszeilenprozessors (CLP), die Sie ausführen müssen, um die Datenbank SAMPLE bis zum aktuellen Zeitpunkt erneut zu erstellen. Als Erstes müssen Sie die Tabellenbereiche auswählen, die Sie erneut erstellen wollen. Da Sie die Datenbank bis zum aktuellen Zeitpunkt erneut erstellen wollen, müssen Sie das aktuellste Backup-Image als Zielimage auswählen. Das aktuellste Backup-Image ist Image TS5, das in Protokollkette 2 enthalten ist:

```
    db2 restore db sample rebuild with all tablespaces in database taken at
          TS5 without prompting
    db2 rollforward db sample to end of logs
db2 rollforward db sample stop
```

Diese Befehle stellen die Backup-Images TS5, TS4, TS1 und DB1 automatisch wieder her und führen eine aktualisierende Recovery der Datenbank bis zum Ende von Protokollkette 2 aus.

Anmerkung: Alle Protokolle, die zu Protokollkette 2 gehören, müssen für die vollständige Ausführung der aktualisierenden Recoveryoperationen zugänglich sein.

Beispiel 2

Dieses zweite Beispiel veranschaulicht die CLP-Befehle, die Sie ausführen müssen, um die Datenbank SAMPLE bis zum Ende von Protokollkette 1 erneut zu erstellen. Das Zielimage, das Sie dazu auswählen, sollte das aktuellste Backup-Image in Protokollkette 1 sein. Dies ist TS3:

```
    db2 restore db sample rebuild with all tablespaces in database
          taken at TS3 without prompting
    db2 rollforward db sample to end of logs
db2 rollforward db sample stop
```

Diese Befehle stellen die Backup-Images TS3, TS2, TS1 und DB1 automatisch wieder her und führen eine aktualisierende Recovery der Datenbank bis zum Ende von Protokollkette 1 aus.

Anmerkung: Alle Protokolle, die zu Protokollkette 1 gehören, müssen für die vollständige Ausführung der aktualisierenden Recoveryoperationen zugänglich sein.

Auswählen des falschen Zielimages zur erneuten Erstellung

Nehmen Sie an, eine Datenbank mit dem Namen SAMPLE2 enthält die folgenden Tabellenbereiche:

- SYSCATSPACE (Systemkataloge)
- USERSP1 (Tabellenbereich für Benutzerdaten)
- USERSP2 (Tabellenbereich für Benutzerdaten)

Abb. 24 auf Seite 328 zeigt die Backup-Protokollkette für SAMPLE2, die aus den folgenden Backups besteht:

1. BK1 ist ein Datenbankgesamtbackup mit allen Tabellenbereichen.
2. BK2 ist ein Tabellenbereichsgesamtbackup von USERSP1.
3. BK3 ist ein Tabellenbereichsgesamtbackup von USERSP2.

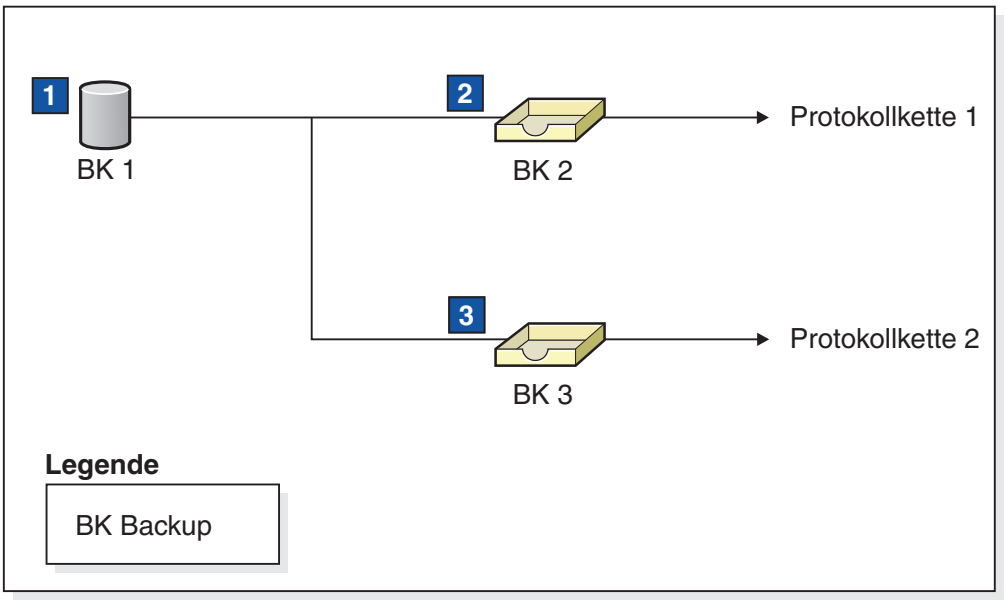


Abbildung 24. Backup-Protokollkette für die Datenbank SAMPLE2

Das folgende Beispiel zeigt den CLP-Befehl, den Sie ausführen müssen, um die Datenbank aus BK3 mit den Tabellenbereichen SYSCATSPACE und USERSP2 erneut zu erstellen:

```
db2 restore db sample2 rebuild with tablespace (SYSCATSPACE,
      USERSP2) taken at BK3 without prompting
```

Nehmen Sie nun an, dass nach Abschluss dieser Restoreoperation auch der Tabellenbereich USERSP1 wiederhergestellt werden soll und daher der folgende Befehl ausgeführt wird:

```
db2 restore db sample2 tablespace (USERSP1) taken at BK2
```

Diese Restoreoperation schlägt fehl und gibt eine Nachricht aus, die besagt, dass BK2 aus der falschen Protokollkette stammt (SQL2154N). Wie Abb. 24 zu entnehmen ist, kann nur das Image BK1 für den Restore von USERSP1 verwendet werden. Daher muss der folgende Befehl eingegeben werden:

```
db2 restore db sample2 tablespace (USERSP1) taken at BK1
```

Dieser Befehl wird erfolgreich ausgeführt, sodass für die Datenbank eine entsprechende aktualisierende Recovery erfolgen kann.

Erneutes Erstellen ausgewählter Tabellenbereiche

Die erneute Erstellung einer Datenbank bietet Ihnen die Möglichkeit, eine Datenbank zu erstellen, die eine Untergruppe der Tabellenbereiche enthält, die Bestandteil der ursprünglichen Datenbank sind. Die erneute Erstellung nur einer Untergruppe von Tabellenbereichen in einer Datenbank kann in folgenden Situationen nützlich sein:

- In einer Test- und Entwicklungsumgebung, in der Sie nur mit einem Teil der Tabellenbereiche arbeiten möchten.
- In einem Recoveryfall, in dem Sie Tabellenbereiche, die kritischer sind, schneller wieder online verfügbar machen müssen als andere, können Sie zunächst eine Teilgruppe von Tabellenbereichen wiederherstellen und die Wiederherstellung anderer Tabellenbereiche zu einem späteren Zeitpunkt nachholen.

Betrachten Sie das folgende Beispiel zur erneuten Erstellung einer Datenbank, die eine Teilgruppe der Tabellenbereiche enthält, die Bestandteil der ursprünglichen Datenbank sind.

Für dieses Beispiel wird angenommen, dass eine Datenbank mit dem Namen MYDB vorhanden ist und die folgenden Tabellenbereiche enthält:

- SYSCATSPACE (Systemkataloge)
- USERSP1 (Tabellenbereich für Benutzerdaten)
- USERSP2 (Tabellenbereich für Benutzerdaten)
- USERSP3 (Tabellenbereich für Benutzerdaten)

Die folgenden Backups wurden erstellt:

- BK1 ist ein Backup von SYSCATSPACE und USERSP1.
- BK2 ist ein Backup von USERSP2 und USERSP3.
- BK3 ist ein Backup von USERSP3.

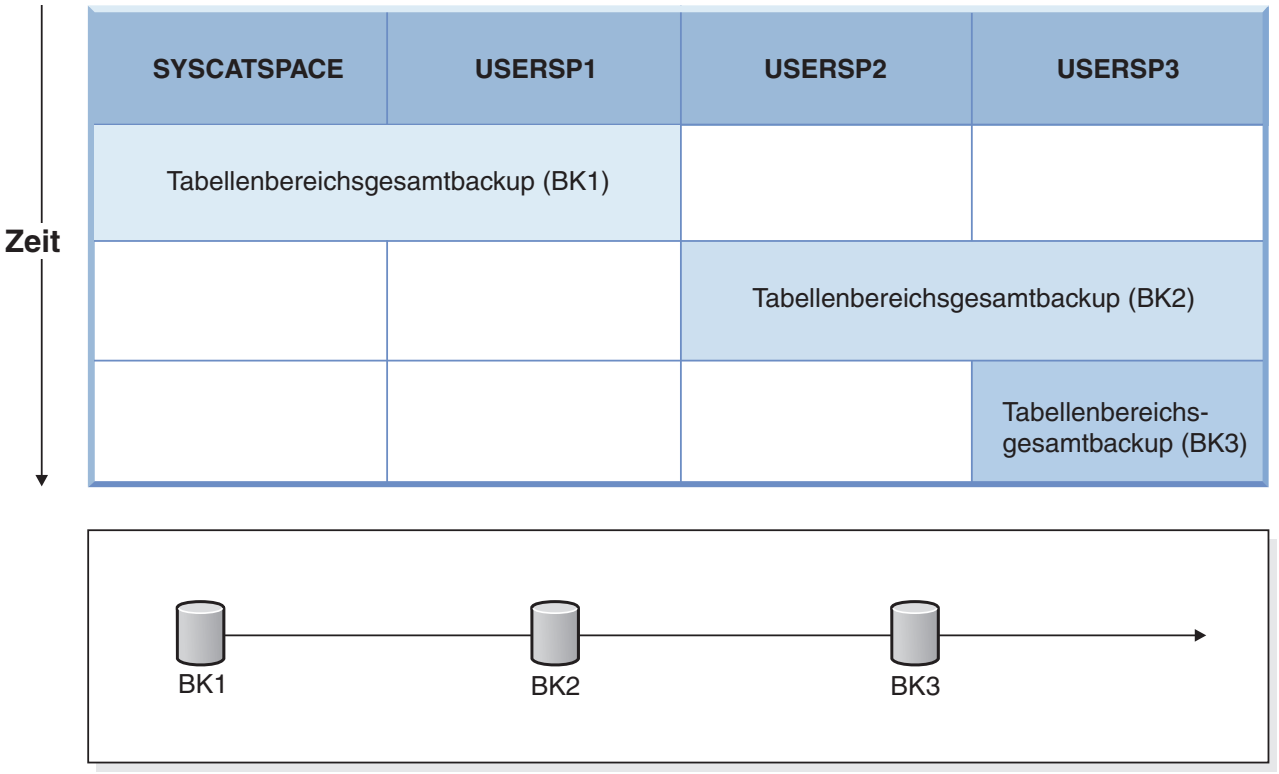


Abbildung 25. Für die Datenbank SAMPLE verfügbare Backup-Images

Die folgende Prozedur veranschaulicht die Verwendung der Befehle RESTORE DATABASE und ROLLFORWARD DATABASE über den Befehlszeilenprozessor (CLP) zur erneuten Erstellung nur der Tabellenbereiche SYSCATSPACE und USERSP1 bis zum Ende der Protokolle:

```
db2 restore db mydb rebuild with all tablespaces in image
             taken at BK1 without prompting
   db2 rollforward db mydb to end of logs
   db2 rollforward db mydb stop
```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und nur die Tabellenbereiche SYSCATSPACE und USERSP1 befinden sich im Status NORMAL. Die Tabellenbereiche USERSP2 und USERSP3 befinden sich im Status *Restore anstehend*. Zu einem späteren Zeitpunkt können Sie weiterhin auch die Tabellenbereiche USERSP2 und USERSP3 wiederherstellen.

Erneutes Erstellen mit Images inkrementeller Backups

Sie können eine Datenbank mithilfe inkrementeller Backup-Images erneut erstellen. Standardmäßig versucht das Restoredienstprogramm, einen automatischen inkrementellen Restore für alle inkrementellen Images auszuführen. Dies bedeutet, dass, wenn Sie die Option INCREMENTAL des Befehls RESTORE DATABASE nicht verwenden, jedoch das Zielimage ein inkrementelles Backup-Image ist, das Restoredienstprogramm eine Rebuildoperation unter Verwendung des automatischen inkrementellen Restores ausführt. Wenn das Zielimage kein inkrementelles Image ist, jedoch ein anderes erforderliches Image ein inkrementelles Image ist, stellt das Restoredienstprogramm sicher, dass solche inkrementellen Images durch den automatischen inkrementellen Restore wiederhergestellt werden. Das Restoredienstprogramm verhält sich immer gleich, unabhängig davon, ob Sie die Option INCREMENTAL mit der Option AUTOMATIC angeben oder nicht.

Wenn Sie die Option INCREMENTAL, jedoch nicht die Option AUTOMATIC angeben, müssen Sie den gesamten Prozess zur erneuten Erstellung (Rebuild) manuell ausführen. Das Restoredienstprogramm stellt in diesem Fall nur die Anfangsmetadaten aus dem Zielimage wieder her, wie es dies auch bei einem regulären manuellen inkrementellen Restore tun würde. Anschließend müssen Sie dann den Restore des Zielimages unter Verwendung der erforderlichen inkrementellen Restorekette zu Ende führen. Und schließlich müssen Sie die verbleibenden Images wiederherstellen, um die Datenbank erneut zu erstellen.

Es wird empfohlen, den automatischen inkrementellen Restore zur erneuten Erstellung einer Datenbank zu verwenden. Nur bei einem Restorefehler sollten Sie versuchen, eine Datenbank mithilfe manueller Methoden erneut zu stellen.

Erneutes Erstellen von partitionierten Datenbanken

Zur erneuten Erstellung (Rebuild) einer partitionierten Datenbank erstellen Sie jede Datenbankpartition separat erneut. Für jede Datenbankpartition, angefangen bei der Katalogpartition, stellen Sie zunächst alle Tabellenbereiche, die Sie benötigen, mit RESTORE wieder her. Alle Tabellenbereiche, die nicht wiederhergestellt werden, werden in den Status *Restore anstehend* versetzt. Wenn alle Datenbankpartitionen wiederhergestellt sind, setzen Sie den Befehl ROLLFORWARD DATABASE in der Katalogpartition ab, um eine aktualisierende Recovery aller Datenbankpartitionen auszuführen.

Anmerkung: Wenn Sie zu einem späteren Zeitpunkt Tabellenbereiche wiederherstellen müssen, die ursprünglich nicht in die Rebuildphase mit eingeschlossen waren, müssen Sie sicherstellen, dass bei der nachfolgenden aktualisierenden Recovery des Tabellenbereichs das Dienstprogramm für aktualisierende Recovery (Rollforward) alle Daten über die Datenbankpartitionen hinweg synchron hält. Wenn ein Tabellenbereich bei der ursprünglichen Restore- und aktualisierenden Recoveryoperation versehentlich unberücksichtigt bleibt, wird sein Fehlen unter Umständen erst erkannt, wenn bei einem Versuch, auf die Daten zuzugreifen, ein Datenzugriffsfehler auftritt. Sie müssen den fehlenden Tabellenbereich mit Restore wiederherstellen und eine aktualisierende Recovery für ihn ausführen, um ihn mit den übrigen Partitionen zu resynchronisieren.

Betrachten Sie das folgende Beispiel im Hinblick auf die erneute Erstellung einer partitionierten Datenbank mithilfe von Backup-Images, die auf Tabellenbereichsebene erstellt wurden.

Für dieses Beispiel wird angenommen, dass eine wiederherstellbare Datenbank mit dem Namen SAMPLE vorhanden ist und drei Datenbankpartitionen enthält:

- Datenbankpartition 1 enthält die Tabellenbereiche SYSCATSPACE, USERSP1 und USERSP2 und ist die Katalogpartition.
- Datenbankpartition 2 enthält die Tabellenbereiche USERSP1 und USERSP3.
- Datenbankpartition 3 enthält die Tabellenbereiche USERSP1, USERSP2 und USERSP3.

Die folgenden Backups wurden erstellt, wobei BKxy die Backupnummer x auf Partition y bezeichnet:

- BK11 ist ein Backup der Tabellenbereiche SYSCATSPACE, USERSP1 und USERSP2.
- BK12 ist ein Backup der Tabellenbereiche USERSP2 und USERSP3.
- BK13 ist ein Backup der Tabellenbereiche USERSP1, USERSP2 und USERSP3.
- BK21 ist ein Backup des Tabellenbereichs USERSP1.
- BK22 ist ein Backup des Tabellenbereichs USERSP1.
- BK23 ist ein Backup des Tabellenbereichs USERSP1.
- BK31 ist ein Backup des Tabellenbereichs USERSP2.
- BK33 ist ein Backup des Tabellenbereichs USERSP2.
- BK42 ist ein Backup des Tabellenbereichs USERSP3.
- BK43 ist ein Backup des Tabellenbereichs USERSP3.

Die folgende Prozedur veranschaulicht die Verwendung der Befehle RESTORE DATABASE und ROLLFORWARD DATABASE über den Befehlszeilenprozessor (CLP) zur erneuten Erstellung der gesamten Datenbank bis zum Ende der Protokolle.

1. Führen Sie in Datenbankpartition 1 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

   ```
   db2 restore db sample rebuild with all tablespaces in database
         taken at BK31 without prompting
   ```

2. Führen Sie in Datenbankpartition 2 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

   ```
   db2 restore db sample rebuild with tablespaces in database
         taken at BK42 without prompting
   ```

3. Führen Sie in Datenbankpartition 3 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

   ```
   db2 restore db sample rebuild with all tablespaces in database
         taken at BK43 without prompting
   ```

4. Führen Sie in der Katalogpartition einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus:

   ```
   db2 rollforward db sample to end of logs
   ```

5. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

   ```
   db2 rollforward db sample stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank in allen Datenbankpartitionen hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Einschränkungen für das erneute Erstellen von Datenbanken

Die folgende Liste enthält eine Zusammenfassung der Einschränkungen für die erneute Erstellung (Option REBUILD):

- Einer der Tabellenbereiche, die erneut erstellt werden, muss der Tabellenbereich SYSCATSPACE in der Katalogpartition sein.
- Sie können eine erneute Erstellung nicht über die GUI-Tools der Steuerzentrale ausführen. Sie müssen entweder Befehle über den Befehlszeilenprozessor (CLP) ausführen oder die entsprechenden Anwendungsprogrammierschnittstellen (APIs) verwenden.
- Die Option REBUILD kann nicht für ein Zielimage aus einer Version vor Version 9.1 verwendet werden, es sei denn, es handelt sich um ein Image eines Offline-Backups der Datenbank. Wenn das Zielimage ein Offline-Backup der Datenbank ist, können nur die Tabellenbereiche in diesem Image für die erneute Erstellung verwendet werden. Die Datenbank muss nach einem erfolgreichen Abschluss der Rebuildoperation migriert werden. Versuche, eine erneute Erstellung mit irgendeinem anderen Typ von Zielimage aus einer Version vor Version 9.1 durchzuführen, schlagen fehl.
- Die Option REBUILD kann nicht für ein Zielimage angegeben werden, das unter einem anderen Betriebssystem erstellt wurde als dem, unter dem es wiederhergestellt wird, sofern es sich nicht um ein Datenbankgesamtbackup handelt. Wenn das Zielimage ein Gesamtbackup der Datenbank ist, können nur die Tabellenbereiche in diesem Image für die erneute Erstellung verwendet werden. Versuche, eine erneute Erstellung mit irgendeinem anderen Typ von Zielimage durchzuführen, das unter einem anderen Betriebssystem erstellt wurde als dem, unter dem es wiederhergestellt wird, schlagen fehl.

Optimieren der Leistung des Restoredienstprogramms

Wenn Sie eine Restoreoperation ausführen, wählt DB2 automatisch optimale Werte für die Anzahl Puffer, die Puffergröße und die Parallelitätseinstellungen aus. Die Werte basieren auf der Menge des für Dienstprogramme verfügbaren Zwischenspeichers, der Anzahl verfügbarer Prozessoren und der Datenbankkonfiguration. Daher sollten Sie je nach der Größe der auf Ihrem System verfügbaren Speicherkapazität in Betracht ziehen, mehr Speicher zuzuordnen, indem Sie den Wert des Konfigurationsparameters UTIL_HEAP_SZ erhöhen. Dadurch soll die zum Ausführen einer Restoreoperation erforderliche Zeit minimiert werden. Wenn Sie für die folgenden Parameter des Befehls RESTORE DATABASE nicht explizit Werte angeben, wählt DB2 einen Wert für die Parameter aus:

- WITH anzahl-puffer BUFFERS
- PARALLELISM n
- BUFFER puffergröße

Für Restoreoperationen wird immer ein Vielfaches der von der Backup-Operation verwendeten Puffergröße verwendet. Sie können beim Absetzen des Befehls RESTORE DATABASE eine Puffergröße angeben, müssen jedoch sicherstellen, dass es sich dabei um ein Vielfaches der Backup-Puffergröße handelt.

Sie können auch eine der folgenden Aktionen ausführen, um die für eine Restoreoperation erforderliche Zeit zu reduzieren:

- Vergrößern Sie die Restorepuffer.

  Die Größe der Restorepuffer muss eine positive ganze Zahl und ein Vielfaches der Backup-Puffergröße sein, die bei der Backup-Operation angegeben wurde. Wird eine nicht korrekte Puffergröße angegeben, erhalten die zugeordneten Puffer die kleinstmögliche Größe.

- Erhöhen Sie die Anzahl der Puffer.

  Der Wert, den Sie angeben, muss ein Vielfaches der Anzahl der Seiten sein, die Sie für den Backup-Puffer angegeben haben. Die Mindestanzahl beträgt 8 Seiten.
- Erhöhen Sie den Wert des Parameters PARALLELISM.

  Dadurch wird die Anzahl der Puffermanipulatoren erhöht, die verwendet werden, um während der Restoreoperation in die Datenbank zu schreiben.
- Erhöhen Sie den Wert für die Größe des Zwischenspeichers für Dienstprogramme.

  Dies vergrößert den Speicher, der gleichzeitig durch andere Dienstprogramme genutzt werden kann.

Erforderliche Zugriffsrechte und Berechtigungen zur Verwendung von Restore

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf diese zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie übergeordnete Pflege- und Dienstprogrammoperationen des Datenbankmanagers zusammenzufassen. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf die Objekte zugreifen, für die sie zugriffsberechtigt sind, d. h. für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung SYSADM, SYSCTRL oder SYSMAINT, um eine *vorhandene* Datenbank mithilfe einer Backupkopie der gesamten Datenbank wiederherstellen zu können. Für das Wiederherstellen in eine *neue* Datenbank ist die Berechtigung SYSADM oder SYSCTRL erforderlich.

Restore - Beispiele

Sitzungen für umgeleiteten Restore - CLP-Beispiele

Beispiel 1

Es folgt ein typisches Beispielszenario für den umgeleiteten, nicht inkrementellen Restore einer Datenbank mit dem Aliasnamen MEINEDB:

1. Setzen Sie einen Befehl RESTORE DATABASE mit der Option REDIRECT ab.

   ```
   db2 restore db meinedb replace existing redirect
   ```

2. Setzen Sie für jeden Tabellenbereich, dessen Container Sie erneut definieren möchten, einen Befehl SET TABLESPACE CONTAINERS ab. Beispiel für eine Windows-Umgebung:

   ```
   db2 set tablespace containers for 5 using
             (file 'f:\ts3con1' 20000, file 'f:\ts3con2' 20000)
   ```

   Setzen Sie für jeden Tabellenbereich, dessen Containerpositionen erneut definiert werden, den Befehl LIST TABLESPACE CONTAINERS ab, um sicherzustellen, dass es sich bei den Containern der wiederhergestellten Datenbank um die in diesem Schritt angegebenen Container handelt.

3. Nach der erfolgreichen Durchführung der Schritte 1 und 2 setzen Sie den folgenden Befehl ab:

   ```
   db2 restore db mydb continue
   ```

   Dies ist der letzte Schritt des umgeleiteten Restores.

4. Falls Schritt 3 fehlschlägt oder die Restoreoperation abgebrochen wurde, kann der umgeleitete Restore erneut gestartet werden, indem Sie wieder bei Schritt 1 beginnen.

Anmerkung:

1. Nach der erfolgreichen Ausführung von Schritt 1 und vor der Vollendung von Schritt 3 kann die Restoreoperation durch Absetzen des folgenden Befehls abgebrochen werden:

   ```
   db2 restore db meinedb abort
   ```

2. Falls Schritt 3 fehlschlägt oder die Restoreoperation abgebrochen wurde, kann der umgeleitete Restore erneut gestartet werden, indem Sie wieder bei Schritt 1 beginnen.

Beispiel 2

Es folgt ein typisches Beispielszenario für den manuellen, umgeleiteten inkrementellen Restore einer Datenbank mit dem Aliasnamen MEINEDB und den folgenden Backup-Images:

```
   backup db meinedb
Das Backup war erfolgreich. Die Zeitmarke für dieses Backup-Image ist: <ts1>

   backup db meinedb incremental
Das Backup war erfolgreich. Die Zeitmarke für dieses Backup-Image ist: <ts2>
```

1. Setzen Sie einen Befehl RESTORE DATABASE mit den Optionen INCREMENTAL und REDIRECT ab.

   ```
   db2 restore db meinedb incremental taken at <zm2> replace existing redirect
   ```

2. Setzen Sie für jeden Tabellenbereich, dessen Container erneut definiert werden müssen, einen Befehl SET TABLESPACE CONTAINERS ab. Beispiel für eine Windows-Umgebung:

   ```
   db2 set tablespace containers for 5 using
             (file 'f:\ts3con1' 20000, file 'f:\ts3con2' 20000)
   ```

   Setzen Sie den Befehl LIST TABLESPACE CONTAINERS ab, um sicherzustellen, dass es sich bei den Containern der wiederhergestellten Datenbank um die in diesem Schritt angegebenen Container handelt.

3. Nach der erfolgreichen Durchführung der Schritte 1 und 2 setzen Sie den folgenden Befehl ab:

   ```
   db2 restore db mydb continue
   ```

4. Die übrigen Befehle zum inkrementellen Restore können nun wie folgt abgesetzt werden:

   ```
   db2 restore db meinedb incremental taken at <zm1>
   db2 restore db meinedb incremental taken at <ts2>
   ```

   Dies ist der letzte Schritt des umgeleiteten Restores.

Anmerkung:

1. Nach der erfolgreichen Ausführung von Schritt 1 und vor der Vollendung von Schritt 3 kann die Restoreoperation durch Absetzen des folgenden Befehls abgebrochen werden:

   ```
   db2 restore db meinedb abort
   ```

2. Nach der erfolgreichen Ausführung von Schritt 3 und vor dem Absetzen aller erforderlichen Befehle in Schritt 4 kann die Restoreoperation durch Absetzen des folgenden Befehls abgebrochen werden:

   ```
   db2 restore db meinedb incremental abort
   ```

3. Falls Schritt 3 fehlschlägt oder die Restoreoperation abgebrochen wurde, kann der umgeleitete Restore erneut gestartet werden, indem Sie wieder bei Schritt 1 beginnen.
4. Wenn der Befehl zum Restore in Schritt 4 fehlschlägt, kann der fehlgeschlagene Befehl erneut abgesetzt werden, um den Restoreprozess fortzusetzen.

Beispiel 3

Es folgt ein typisches Beispielszenario für den automatischen, umgeleiteten inkrementellen Restore derselben Datenbank:

1. Setzen Sie einen Befehl RESTORE DATABASE mit den Optionen INCREMENTAL AUTOMATIC und REDIRECT ab.

   ```
   db2 restore db meinedb incremental automatic taken at <zm2>
                  replace existing redirect
   ```

2. Setzen Sie für jeden Tabellenbereich, dessen Container erneut definiert werden müssen, einen Befehl SET TABLESPACE CONTAINERS ab. Beispiel für eine Windows-Umgebung:

   ```
   db2 set tablespace containers for 5 using
             (file 'f:\ts3con1' 20000, file 'f:\ts3con2' 20000)
   ```

   Setzen Sie den Befehl LIST TABLESPACE CONTAINERS ab, um sicherzustellen, dass es sich bei den Containern der wiederhergestellten Datenbank um die in diesem Schritt angegebenen Container handelt.
3. Nach der erfolgreichen Durchführung der Schritte 1 und 2 setzen Sie den folgenden Befehl ab:

   ```
   db2 restore db mydb continue
   ```

   Dies ist der letzte Schritt des umgeleiteten Restores.

Anmerkung:

1. Nach der erfolgreichen Ausführung von Schritt 1 und vor der Vollendung von Schritt 3 kann die Restoreoperation durch Absetzen des folgenden Befehls abgebrochen werden:

   ```
   db2 restore db meinedb abort
   ```

2. Falls Schritt 3 fehlschlägt oder die Restoreoperation abgebrochen wurde, kann der umgeleitete Restore erneut gestartet werden, indem Sie wieder bei Schritt 1 beginnen, nachdem Sie den folgenden Befehl abgesetzt haben:

   ```
   db2 restore db meinedb incremental abort
   ```

Sitzungen zur erneuten Erstellung - Beispiele für CLP

Szenario 1

Für die folgenden Beispiele wird angenommen, dass eine wiederherstellbare Datenbank des Namens MYDB vorhanden ist, die folgende Tabellenbereiche enthält:

- SYSCATSPACE (Systemkataloge)
- USERSP1 (Tabellenbereich für Benutzerdaten)
- USERSP2 (Tabellenbereich für Benutzerdaten)
- USERSP3 (Tabellenbereich für Benutzerdaten)

Die folgenden Backups wurden erstellt:

- BK1 ist ein Backup von SYSCATSPACE und USERSP1.
- BK2 ist ein Backup von USERSP2 und USERSP3.

- BK3 ist ein Backup von USERSP3.

Beispiel 1

Mit der folgenden Prozedur wird die gesamte Datenbank bis zum aktuellsten Zeitpunkt erneut erstellt:

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus:
   ```
   db2 restore db mydb rebuild with all tablespaces in database
            taken at BK3 without prompting
   ```
2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):
   ```
   db2 rollforward db mydb to end of logs
   ```
3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:
   ```
   db2 rollforward db mydb stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Beispiel 2

Mit der folgenden Prozedur werden nur die Tabellenbereiche SYSCATSPACE und USERSP2 bis zu einem bestimmten Zeitpunkt erneut erstellt (wobei das Ende von BK3 weiter zurückliegt als der bestimmte Zeitpunkt, der wiederum weiter zurückliegt als das Ende der Protokolle):

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus, indem Sie die Tabellenbereiche angeben, die wiederhergestellt werden sollen.
   ```
   db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP2)
         taken at BK2 without prompting
   ```
2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO PIT aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):
   ```
   db2 rollforward db mydb to PIT
   ```
3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:
   ```
   db2 rollforward db mydb stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und nur die Tabellenbereiche SYSCATSPACE und USERSP2 befinden sich im Status NORMAL. Die Tabellenbereiche USERSP1 und USERSP3 befinden sich im Status *Restore anstehend* (RESTORE_PENDING).

Gehen Sie zur Wiederherstellung der Tabellenbereiche USERSP1 und USERSP3 zu einem späteren Zeitpunkt durch normale Restoreoperationen für die Tabellenbereiche (ohne Option REBUILD) wie folgt vor:

1. Führen Sie den Befehl RESTORE DATABASE *ohne* die Option REBUILD aus, indem Sie den wiederherzustellenden Tabellenbereich angeben. Stellen Sie zunächst USERSP1 wieder her:
   ```
   db2 restore db mydb tablespace (USERSP1) taken at BK1 without prompting
   ```
2. Stellen Sie anschließend USERSP3 wieder her:
   ```
   db2 restore db mydb tablespace taken at BK3 without prompting
   ```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus, indem Sie die wiederherzustellenden Tabellenbereiche angeben (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

   ```
   db2 rollforward db mydb to end of logs tablespace (USERSP1, USERSP3)
   ```

   Die aktualisierende Recovery (Rollforward) wendet alle Protokolle bis zum Zeitpunkt (PIT) an und stoppt dann für diese beiden Tabellenbereiche, da an ihnen seit der ersten aktualisierenden Recovery keine Arbeiten ausgeführt wurden.
4. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

   ```
   db2 rollforward db mydb stop
   ```

Beispiel 3

Mit der folgenden Prozedur werden nur die Tabellenbereiche SYSCATSPACE und USERSP1 bis zum Ende der Protokolle erneut erstellt:

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

   ```
   db2 restore db mydb rebuild with all tablespaces in image
            taken at BK1 without prompting
   ```
2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

   ```
   db2 rollforward db mydb to end of logs
   ```
3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

   ```
   db2 rollforward db mydb stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und nur die Tabellenbereiche SYSCATSPACE und USERSP1 befinden sich im Status NORMAL. Die Tabellenbereiche USERSP2 und USERSP3 befinden sich im Status *Restore anstehend* (RESTORE_PENDING).

Beispiel 4

Im folgenden Beispiel befinden sich die Backups BK1 und BK2 nicht mehr an derselben Position wie in der Datei des Recoveryprotokolls angeben. Dies ist jedoch bei der Ausführung der Rebuildoperation nicht bekannt.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus, indem Sie angeben, dass die gesamte Datenbank bis zum aktuellsten Zeitpunkt erneut erstellt werden soll:

   ```
   db2 restore db mydb rebuild with all tablespaces in database
            taken at BK3 without prompting
   ```

   An diesem Punkt ist das Zielimage erfolgreich wiederhergestellt. Jedoch wird ein Fehler aus dem Restoredienstprogramm zurückgegeben, der besagt, dass ein erforderliches Image nicht gefunden werden konnte.
2. In diesem Fall müssen Sie die Rebuildoperation manuell ausführen. Da sich die Datenbank in der Rebuildphase befindet, kann dies wie folgt geschehen:
   a. Führen Sie einen Befehl RESTORE DATABASE aus, indem Sie die Position des Backup-Image BK1 angeben:

      ```
      db2 restore db mydb tablespace taken at BK1 from <position>
               without prompting
      ```

b. Führen Sie einen Befehl RESTORE DATABASE aus, indem Sie die Position des Backup-Image BK2 angeben:

```
db2 restore db mydb tablespace (USERSP2) taken at BK2 from
        <position> without prompting
```

c. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

```
db2 rollforward db mydb to end of logs
```

d. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Beispiel 5

In diesem Beispiel enthält der Tabellenbereich USERSP3 unabhängige Daten, die zur Generierung eines speziellen Berichts benötigt werden, jedoch wünschen Sie nicht, dass die Berichtsgenerierung die ursprüngliche Datenbank stört. Um Zugriff auf die Daten zu erhalten, ohne die ursprüngliche Datenbank zu beeinträchtigen, können Sie mit REBUILD eine neue Datenbank nur mit diesem Tabellenbereich und dem Tabellenbereich SYSCATSPACE zu generieren. Der Tabellenbereich SYSCATSPACE ist ebenfalls erforderlich, um nach der Restore- und der aktualisierenden Recoveryoperation die Herstellung einer Verbindung zu der Datenbank zu ermöglichen.

Gehen Sie zum Erstellen einer neuen Datenbank mit den aktuellsten Daten in den Tabellenbereichen SYSCATSPACE und USERSP3 wie folgt vor:

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus, und geben Sie an, dass die Tabellenbereiche SYSCATSPACE und USERSP3 in einer neuen Datenbank namens NEWDB wiederhergestellt werden sollen:

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP3)
      taken at BK3 into newdb without prompting
```

2. Führen Sie einen Befehl ROLLFORWARD DATABASE für NEWDB mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

```
db2 rollforward db newdb to end of logs
```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db newdb stop
```

An diesem Punkt kann eine Verbindung zur neuen Datenbank hergestellt werden, und nur die Tabellenbereiche SYSCATSPACE und USERSP3 befinden sich im Status NORMAL. Die Tabellenbereiche USERSP1 und USERSP2 befinden sich im Status *Restore anstehend* (RESTORE_PENDING).

Anmerkung: Wenn die Containerpfade zwischen der aktuellen Datenbank und der neuen Datenbank problematisch sind (zum Beispiel wenn die Container für die ursprüngliche Datenbank geändert werden müssen, weil das Dateisystem nicht vorhanden ist, oder wenn die Container bereits von der ursprünglichen Datenbank verwendet werden), müssen Sie einen umgeleiteten Restore ausführen. Das obige Beispiel geht von der Annahme aus, dass die Standarddatenbankpfade für dynamischen Speicher für die Tabellenbereiche verwendet werden.

Szenario 2

Für das folgende Beispiel wird angenommen, dass eine wiederherstellbare Datenbank des Namens MYDB vorhanden ist, die den Tabellenbereich SYSCATSPACE und eintausend Benutzertabellenbereiche mit Namen der Form Txxxx besitzt. Dabei stellt xxxx die Tabellenbereichsnummer (z. B. T0001) dar. Es ist ein Datenbankgesamtbackup (BK1) vorhanden.

Beispiel 6

Mit der folgenden Prozedur werden alle Tabellenbereiche außer T0999 und T1000 wiederhergestellt:

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

   ```
   db2 restore db mydb rebuild with all tablespaces in image except
         tablespace (T0999, T1000) taken at BK1 without prompting
   ```

2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

   ```
   db2 rollforward db mydb to end of logs
   ```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

   ```
   db2 rollforward db mydb stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und alle Tabellenbereiche außer T0999 und T1000 befinden sich im Status NORMAL. Die Tabellenbereiche T0999 und T1000 befinden sich im Status *Restore anstehend* (RESTORE_PENDING).

Szenario 3

Die Beispiele in diesem Szenario veranschaulichen, wie eine wiederherstellbare Datenbank mithilfe inkrementeller Backups erneut erstellt wird. Für die folgenden Beispiele wird angenommen, dass eine Datenbank des Namens MYDB vorhanden ist, die folgende Tabellenbereiche enthält:

- SYSCATSPACE (Systemkataloge)
- USERSP1 (Datentabellenbereich)
- USERSP2 (Tabellenbereich für Benutzerdaten)
- USERSP3 (Tabellenbereich für Benutzerdaten)

Die folgenden Backups wurden erstellt:

- FULL1 ist ein Gesamtbackup der Tabellenbereiche SYSCATSPACE, USERSP1, USERSP2 und USERSP3.
- DELTA1 ist ein Deltabackup der Tabellenbereiche SYSCATSPACE und USERSP1.
- INCR1 ist ein inkrementelles Backup der Tabellenbereiche USERSP2 und USERSP3.
- DELTA2 ist ein Deltabackup der Tabellenbereiche SYSCATSPACE, USERSP1, USERSP2 und USERSP3.
- DELTA3 ist Deltabackup des Tabellenbereichs USERSP2.
- FULL2 ist ein Gesamtbackup des Tabellenbereichs USERSP1.

Beispiel 7

Mit der folgenden Prozedur werden nur die Tabellenbereiche SYSCATSPACE und USERSP2 bis zum aktuellsten Zeitpunkt durch eine inkrementelle automatische Restoreoperation erneut erstellt.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus. Die Option INCREMENTAL AUTO ist optional. Das Restoredienstprogramm erkennt die Granularität des Images und verwendet einen automatischen inkrementellen Restore, wenn er erforderlich ist.

   ```
   db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP2)
         incremental auto taken at DELTA3 without prompting
   ```

2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

   ```
   db2 rollforward db mydb to end of logs
   ```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

   ```
   db2 rollforward db mydb stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und nur die Tabellenbereiche SYSCATSPACE und USERSP2 befinden sich im Status NORMAL. Die Tabellenbereiche USERSP1 und USERSP3 befinden sich im Status *Restore anstehend* (RESTORE_PENDING).

Beispiel 8

Mit der folgenden Prozedur wird die gesamte Datenbank bis zum aktuellsten Zeitpunkt durch einen inkrementellen automatischen Restore erneut erstellt.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus. Die Option INCREMENTAL AUTO ist optional. Das Restoredienstprogramm erkennt die Granularität des Images und verwendet einen automatischen inkrementellen Restore, wenn er erforderlich ist.

   ```
   db2 restore db mydb rebuild with all tablespaces in database
         incremental auto taken at DELTA3 without prompting
   ```

2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

   ```
   db2 rollforward db mydb to end of logs
   ```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

   ```
   db2 rollforward db mydb stop
   ```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Beispiel 9

Mit der folgenden Prozedur wird die gesamte Datenbank außer Tabellenbereich USERSP3 bis zum aktuellsten Zeitpunkt erneut erstellt.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus. Obwohl es sich bei dem Zielimage um ein nicht inkrementelles Image handelt, erkennt das Restoredienstprogramm, dass die erforderliche Neuerstellungskette inkrementelle Images enthält, und stellt diese Images automatisch inkrementell wieder her.

```
db2 restore db mydb rebuild with all tablespaces in database except
        tablespace (USERSP3) taken at FULL2 without prompting
```

2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

```
db2 rollforward db mydb to end of logs
```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

Szenario 4

Die Beispiele in diesem Szenario veranschaulichen, wie eine wiederherstellbare Datenbank mithilfe von Backup-Images erneut erstellt wird, die Protokolldateien enthalten. Für die folgenden Beispiele wird angenommen, dass eine Datenbank des Namens MYDB vorhanden ist, die folgende Tabellenbereiche enthält:

- SYSCATSPACE (Systemkataloge)
- USERSP1 (Tabellenbereich für Benutzerdaten)
- USERSP2 (Tabellenbereich für Benutzerdaten)

Beispiel 10

Mit der folgenden Prozedur wird die Datenbank nur mit den Tabellenbereichen SYSCATSPACE und USERSP2 bis zum aktuellsten Zeitpunkt erneut erstellt. Es ist ein online erstelltes Image eines Datenbankgesamtbackups (BK1) vorhanden, das Protokolldateien enthält.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP2)
      taken at BK1 logtarget /logs without prompting
```

2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle nach dem Ende von BK1 gesichert wurden und zugänglich sind):

```
db2 rollforward db mydb to end of logs overflow log path (/logs)
```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und nur die Tabellenbereiche SYSCATSPACE und USERSP2 befinden sich im Status NORMAL. Der Tabellenbereich USERSP1 befindet sich im Status *Restore anstehend* (RESTORE_PENDING).

Beispiel 11

Mit der folgenden Prozedur wird die Datenbank bis zum aktuellsten Zeitpunkt erneut erstellt. Es sind zwei online erstellte Images von Tabellenbereichsgesamtbackups vorhanden, die Protokolldateien enthalten:

- BK1 ist ein Backup von SYSCATSPACE mit den Protokolldateien 10-45.
- BK2 ist ein Backup von USERSP1 und USERSP2 mit den Protokolldateien 64-80.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in database
        taken at BK2 logtarget /logs without prompting
```

Die aktualisierende Recoveryoperation beginnt mit Protokolldatei 10, die sie immer im Überlaufprotokollpfad findet, wenn sie sich nicht im primären Protokolldateipfad befindet. Der Protokollbereich 46-63 muss für die aktualisierende Recovery verfügbar gemacht werden, da diese Protokolldateien in keinem Backup-Image enthalten sind.

2. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS unter Verwendung des Überlaufprotokollpfads für die Protokolldateien 64-80 aus:

```
db2 rollforward db mydb to end of logs overflow log path (/logs)
```

3. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Szenario 5

Für die folgenden Beispiele wird angenommen, dass eine wiederherstellbare Datenbank des Namens MYDB vorhanden ist, die folgende Tabellenbereiche enthält:

- SYSCATSPACE (0), SMS-Systemkatalog (relativ definierter Container)
- USERSP1 (1), SMS-Tabellenbereich für Benutzerdaten (relativ definierter Container)
- USERSP2 (2) DMS-Tabellenbereich für Benutzerdaten (absolut definierter Container /usersp2)
- USERSP3 (3) DMS-Tabellenbereich für Benutzerdaten (absolut definierter Container /usersp3)

Die folgenden Backups wurden erstellt:

- BK1 ist ein Backup von SYSCATSPACE und USERSP1.
- BK2 ist ein Backup von USERSP2 und USERSP3.
- BK3 ist ein Backup von USERSP3.

Beispiel 12

Mit der folgenden Prozedur wird die gesamte Datenbank bis zum aktuellsten Zeitpunkt durch einen umgeleiteten Restore erneut erstellt.

1. Führen Sie einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in database
         taken at BK3 redirect without prompting
```

2. Setzen Sie für jeden Tabellenbereich, dessen Container Sie erneut definieren möchten, einen Befehl SET TABLESPACE CONTAINERS ab. Beispiel:

```
db2 set tablespace containers for 3 using (file '/newusersp2' 10000)
```

3.

```
db2 set tablespace containers for 4 using (file '/newusersp3' 15000)
```

4. Führen Sie einen Befehl RESTORE DATABASE mit der Option CONTINUE aus:

```
db2 restore db mydb continue
```

5. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und zugänglich sind):

```
db2 rollforward db mydb to end of logs
```

6. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

An diesem Punkt kann eine Verbindung zur Datenbank hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Szenario 6

Für die folgenden Beispiele wird angenommen, dass eine Datenbank des Namens MYDB mit drei Datenbankpartitionen vorhanden ist:

- Datenbankpartition 1 enthält die Tabellenbereiche SYSCATSPACE, USERSP1 und USERSP2 und ist die Katalogpartition.
- Datenbankpartition 2 enthält die Tabellenbereiche USERSP1 und USERSP3.
- Datenbankpartition 3 enthält die Tabellenbereiche USERSP1, USERSP2 und USERSP3.

Die folgenden Backups wurden erstellt, wobei BKxy die Backupnummer x auf Partition y bezeichnet:

- BK11 ist ein Backup der Tabellenbereiche SYSCATSPACE, USERSP1 und USERSP2.
- BK12 ist ein Backup der Tabellenbereiche USERSP2 und USERSP3.
- BK13 ist ein Backup der Tabellenbereiche USERSP1, USERSP2 und USERSP3.
- BK21 ist ein Backup des Tabellenbereichs USERSP1.
- BK22 ist ein Backup des Tabellenbereichs USERSP1.
- BK23 ist ein Backup des Tabellenbereichs USERSP1.
- BK31 ist ein Backup des Tabellenbereichs USERSP2.
- BK33 ist ein Backup des Tabellenbereichs USERSP2.
- BK42 ist ein Backup des Tabellenbereichs USERSP3.
- BK43 ist ein Backup des Tabellenbereichs USERSP3.

Beispiel 13

Mit der folgenden Prozedur wird die gesamte Datenbank bis zum Ende der Protokolle erneut erstellt.

1. Führen Sie in Datenbankpartition 1 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in database
         taken at BK31 without prompting
```

2. Führen Sie in Datenbankpartition 2 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with tablespaces in database taken at
         BK42 without prompting
```

3. Führen Sie in Datenbankpartition 3 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in database
         taken at BK43 without prompting
```

4. Führen Sie in der Katalogpartition einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus (unter der Voraussetzung, dass alle Protokolle gesichert wurden und in allen Datenbankpartitionen zugänglich sind):

```
db2 rollforward db mydb to end of logs
```

5. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

An diesem Punkt kann eine Verbindung zur Datenbank in allen Datenbankpartitionen hergestellt werden, und alle Tabellenbereiche befinden sich im Status NORMAL.

Beispiel 14

Mit der folgenden Prozedur werden die Tabellenbereiche SYSCATSPACE, USERSP1 und USERSP2 bis zum aktuellsten Zeitpunkt erneut erstellt.

1. Führen Sie in Datenbankpartition 1 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in database
         taken at BK31 without prompting
```

2. Führen Sie in Datenbankpartition 2 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in image taken at
         BK22 without prompting
```

3. Führen Sie in Datenbankpartition 3 einen Befehl RESTORE DATABASE mit der Option REBUILD aus:

```
db2 restore db mydb rebuild with all tablespaces in image taken at
         BK33 without prompting
```

   Hinweis: Dieser Befehl lässt den Tabellenbereich USERSP1 aus, der zum Abschließen der Rebuildoperation erforderlich ist.

4. Führen Sie in der Katalogpartition einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS aus:

```
db2 rollforward db mydb to end of logs
```

5. Führen Sie einen Befehl ROLLFORWARD DATABASE mit der Option STOP aus:

```
db2 rollforward db mydb stop
```

   Die aktualisierende Recovery wird erfolgreich beendet. Zur Datenbank kann eine Verbindung in allen Datenbankpartitionen hergestellt werden. Alle Tabellenbereiche mit folgenden Ausnahmen befinden sich im Status NORMAL: Der Tabellenbereich USERSP3 befindet sich in allen Datenbankpartitionen, in denen er enthalten ist, im Status *Restore anstehend* (RESTORE PENDING), und der Tabellenbereich USERSP1 hat in Datenbankpartition 3 den Status *Restore anstehend* (RESTORE PENDING).

   Wenn ein Versuch unternommen wird, auf Daten in USERSP1 in Datenbankpartition 3 zuzugreifen, tritt ein Datenzugriffsfehler auf. Um diesen zu beheben, muss der Tabellenbereich USERSP1 wiederhergestellt werden:

   a. Führen Sie in Datenbankpartition 3 einen Befehl RESTORE DATABASE aus, indem Sie ein Backup-Image angeben, das den Tabellenbereich USERSP1 enthält:

```
db2 restore db mydb tablespace taken at BK23 without prompting
```

   b. Führen Sie in der Katalogpartition einen Befehl ROLLFORWARD DATABASE mit der Option TO END OF LOGS und der Option AND STOP aus:

```
db2 rollforward db mydb to end of logs on dbpartitionnum (3) and stop
```

An diesem Punkt kann auf die Daten des Tabellenbereichs USERSP1 in Datenbankpartition 3 zugegriffen werden, da sich der Tabellenbereich im Status NORMAL befindet.

Szenario 7

Für die folgenden Beispiele wird angenommen, dass eine *nicht wiederherstellbare* Datenbank des Namens MYDB mit den folgenden Tabellenbereichen vorhanden ist:

- SYSCATSPACE (0), SMS-Systemkatalog
- USERSP1 (1), SMS-Tabellenbereich für Benutzerdaten
- USERSP2 (2), DMS-Tabellenbereich für Benutzerdaten
- USERSP3 (3), DMS-Tabellenbereich für Benutzerdaten

Es ist nur ein Backup (BK1) der Datenbank vorhanden.

Beispiel 15

Die folgende Prozedur veranschaulicht die Verwendung der Rebuildoperation für eine nicht wiederherstellbare Datenbank.

Erstellen Sie die Datenbank nur mit den Tabellenbereichen SYSCATSPACE und USERSP1 erneut:

```
db2 restore db mydb rebuild with tablespace (SYSCATSPACE, USERSP1)
   taken at BK1 without prompting
```

Im Anschluss an die Restoreoperation kann eine Verbindung zur Datenbank hergestellt werden. Wenn Sie den Befehl LIST TABLESPACES ausführen, sehen Sie, dass die Tabellenbereiche SYSCATSPACE und USERSP1 im Status NORMAL sind, während sich die Tabellenbereiche USERSP2 und USERSP3 im Status *Löschen anstehend* bzw. *Offline* (DELETE_PENDING/OFFLINE) befinden. Sie können nun mit den beiden Tabellenbereichen arbeiten, die sich im Status NORMAL befinden.

Wenn Sie ein Datenbankbackup erstellen wollen, müssen Sie zunächst die Tabellenbereiche USERSP2 und USERSP3 mithilfe des Befehls DROP TABLESPACE löschen, da das Backup ansonsten fehlschlägt.

Wenn Sie die Tabellenbereiche zu einem späteren Zeitpunkt wiederherstellen wollen, müssen Sie erneut einen Befehl RESTORE DATABASE mit BK1 ausführen.

Kapitel 13. Aktualisierende Recovery - Übersicht

In der einfachsten Form des DB2-Befehls ROLLFORWARD DATABASE müssen Sie lediglich den Aliasnamen der Datenbank angeben, die Sie aktualisierend wiederherstellen wollen. Beispiel:

```
db2 rollforward db sample to end of logs and stop
```

In diesem Beispiel gibt der Befehl Folgendes zurück:

```
                                Status der aktualisierenden Recovery

 Aliasname der Eingabedatenbank                       = sample
    Anzahl der Knoten, die Status zurückgaben         = 1

   Knotennummer                               = 0
    Aktualisierende Recovery: Status          = nicht anstehend
    Nächste zu lesende Protokolldatei         =
Verarbeitete Protokolldateien                 =  -
Letzte festgeschriebene Transaktion           = 2001-03-11-02.39.48.000000

 DB20000I  Der Befehl ROLLFORWARD wurde erfolgreich ausgeführt.
```

Im Folgenden wird eine mögliche Vorgehensweise zur Durchführung einer aktualisierenden Recovery beschrieben:

1. Rufen Sie das Dienstprogramm zur aktualisierenden Recovery ohne die Option STOP auf.
2. Rufen Sie das Dienstprogramm zur aktualisierenden Recovery mit der Option QUERY STATUS auf.

   Wenn Sie eine Recovery bis zum Ende der Protokolle angeben, kann die Option QUERY STATUS angeben, dass eine oder mehrere Protokolldateien fehlen, falls der zurückgegebene Zeitpunkt vor dem erwarteten Zeitpunkt liegt.

   Wenn Sie eine Recovery bis zu einem bestimmten Zeitpunkt angeben, können Sie mit der Option QUERY STATUS sicherstellen, dass die aktualisierende Recovery am richtigen Zeitpunkt beendet wird.
3. Rufen Sie das Dienstprogramm zur aktualisierenden Recovery mit der Option STOP auf. Nach Beendigung dieser Operation können keine weiteren Änderungen mehr aktualisierend wiederhergestellt werden.

Eine mögliche Alternative besteht darin, die aktualisierende Recovery folgendermaßen auszuführen:

1. Rufen Sie das Dienstprogramm zur aktualisierenden Recovery mit der Option AND STOP auf.
2. Die Notwendigkeit weiterer Schritte hängt vom Ergebnis der aktualisierenden Recoveryoperation ab:
   - Wenn sie erfolgreich ist, wurde die aktualisierende Recovery abgeschlossen. Eine Verbindung zur Datenbank kann hergestellt und die Datenbank selbst genutzt werden. An diesem Punkt können keine weiteren Änderungen mehr aktualisierend wiederhergestellt werden.
   - Wenn Fehler zurückgemeldet werden, führen Sie die erforderlichen Maßnahmen zur Korrektur des Problems aus (z. B. wenn eine Protokolldatei fehlt, suchen Sie die Protokolldatei, oder wenn Abruffehler vorliegen, stellen Sie si-

cher, dass die Protokollarchivierung funktioniert). Anschließend rufen Sie das Dienstprogramm zur aktualisierenden Recovery mit der Option AND STOP erneut auf.

Eine Datenbank muss erst erfolgreich wiederhergestellt werden (mit dem Restoredienstprogramm), bevor sie aktualisierend wiederhergestellt werden kann. Bei einem Tabellenbereich ist dies jedoch nicht erforderlich. Ein Tabellenbereich kann zeitweilig in den Status *Aktualisierende Recovery anstehend* versetzt werden, erfordert aber keine Restoreoperation, um diesen Status aufzuheben (z. B. nach einem Stromausfall).

Wenn das Dienstprogramm zur aktualisierenden Recovery aufgerufen wird, geschieht Folgendes:

- Wenn sich die Datenbank im Status *Aktualisierende Recovery anstehend* befindet, wird die Datenbank aktualisierend wiederhergestellt. Wenn auch Tabellenbereiche diesen Status haben, müssen Sie nach der aktualisierenden Recovery der Datenbank das Dienstprogramm zur aktualisierenden Recovery erneut aufrufen, um diese Tabellenbereiche aktualisierend wiederherzustellen.
- Wenn sich die Datenbank *nicht* im Status "Aktualisierende Recovery anstehend" befindet, die Tabellenbereiche in der Datenbank sich jedoch im Status "Aktualisierende Recovery anstehend" *befinden*:
  - Wenn Sie eine Liste mit Tabellenbereichen angeben, werden nur diese Tabellenbereiche aktualisierend wiederhergestellt.
  - Wenn Sie keine Liste mit Tabellenbereichen angeben, werden alle Bereiche aktualisierend wiederhergestellt, die den Status *Aktualisierende Recovery anstehend* haben.

Die aktualisierende Recovery einer Datenbank wird offline durchgeführt. Die Datenbank steht erst dann wieder zur Verfügung, wenn die aktualisierende Recovery erfolgreich beendet wurde. Diese Operation kann jedoch nur beendet werden, wenn beim Aufrufen des Dienstprogramms die Option STOP angegeben wurde.

Die aktualisierende Recovery von Tabellenbereichen kann offline durchgeführt werden. Die Datenbank steht erst dann wieder zur Verfügung, wenn die aktualisierende Recovery erfolgreich beendet wurde. Dies geschieht, wenn das Ende der Protokolle erreicht wird oder wenn beim Aufrufen des Dienstprogramms die Option STOP angegeben wurde.

Wenn SYSCATSPACE nicht betroffen ist, können Sie die aktualisierende Recovery von Tabellenbereichen auch *online* durchführen. Wenn Sie eine aktualisierende Recovery für einen Tabellenbereich online durchführen, steht dieser während der Operation nicht zur Verfügung. Die anderen Tabellenbereiche in der Datenbank *sind* jedoch verfügbar.

Wenn Sie eine Datenbank erstellen, wird für diese Datenbank zunächst nur die Umlaufprotokollierung aktiviert. Das bedeutet, dass die Protokolle wiederverwendet und nicht gespeichert oder archiviert werden. Bei Verwendung der Umlaufprotokollierung ist eine aktualisierende Recovery nicht möglich. Es kann lediglich eine Recovery nach einem Systemabsturz bzw. eine Versionsrecovery durchgeführt werden. Archivierte Protokolldateien erfassen die Änderungen, die nach dem Erstellen eines Backup-Images an einer Datenbank vorgenommen werden. Sie können diese Protokollierung (und aktualisierende Recovery) aktivieren, indem Sie den Datenbankkonfigurationsparameter *logarchmeth1* auf einen anderen Wert als den Standardwert `OFF` setzen. Wenn Sie *logarchmeth1* auf einen Wert ungleich `OFF` setzen,

wird die Datenbank in den Status "Backup anstehend" gesetzt, und Sie müssen ein Offline-Backup der Datenbank ausführen, bevor sie wiederverwendet werden kann.

Anmerkung: Für jede Protokolldatei, die in einer Operation zur aktualisierenden Recovery verwendet wird, wird in der Datei des Recoveryprotokolls ein Eintrag erstellt.

Verwenden der aktualisierenden Recovery

Verwenden Sie den Befehl ROLLFORWARD DATABASE, um Transaktionen, die in den Datenbankprotokolldateien aufgezeichnet wurden, auf ein mit RESTORE wiederhergestelltes Backup-Image einer Datenbank bzw. eines Tabellenbereichs anzuwenden.

Es sollte noch keine Verbindung zu der Datenbank bestehen, die aktualisierend wiederhergestellt werden soll: Das Dienstprogramm zur aktualisierenden Recovery stellt automatisch eine Verbindung zu der angegebenen Datenbank her, die nach Abschluss der aktualisierenden Recovery beendet wird.

Stellen Sie nur dann Tabellenbereiche wieder her, wenn die aktuell laufende Operation zur aktualisierenden Recovery abgebrochen wurde. Andernfalls erhalten Sie möglicherweise eine Tabellenbereichsgruppe, in der sich einige Tabellenbereiche im Status *Aktualisierende Recovery läuft* befinden und andere im Status *Aktualisierende Recovery anstehend*. Eine aktive Operation zur aktualisierenden Recovery wirkt sich nur auf Tabellenbereiche aus, die sich im Status *Aktualisierende Recovery läuft* befinden.

Bei der Datenbank kann es sich um eine lokale oder ferne Datenbank handeln.

Für das Dienstprogramm zur aktualisierenden Recovery gelten die folgenden Einschränkungen:

- Sie können jeweils nur eine Operation zur aktualisierenden Recovery aufrufen. Wenn Sie mehrere Tabellenbereiche wiederherstellen wollen, können Sie alle Bereiche in derselben Operation angeben.
- Wenn Sie nach der letzten Backup-Operation einen Tabellenbereich umbenannt haben, müssen Sie bei der aktualisierenden Recovery sicherstellen, dass der neue Name verwendet wird. Der vorherige Tabellenbereichsname wird nicht erkannt.
- Sie können eine laufende Operation zur aktualisierenden Recovery nicht abbrechen. Es können nur Operationen zur aktualisierenden Recovery abgebrochen werden, die zwar beendet wurden, für die jedoch die Option STOP nicht angegeben wurde, oder Operationen, die vor der erfolgreichen Beendigung fehlgeschlagen sind.
- Sie können eine aktualisierende Recovery eines Tabellenbereichs nicht bis zu einem bestimmten Zeitpunkt *fortsetzen*, wenn die von Ihnen angegebene Zeitmarke vor der vorherigen Zeitmarke liegt. Wenn kein bestimmter Zeitpunkt angegeben wird, wird der vorherige verwendet. Sie können eine aktualisierende Recovery ausgeben, die an einem bestimmten Zeitpunkt dadurch beendet wird, dass Sie lediglich die Option STOP angeben. Dies ist jedoch nur dann zulässig, wenn die betroffenen Tabellenbereiche zuvor alle aus demselben Offline-Backup-Image wiederhergestellt wurden. In diesem Fall ist keine Protokollverarbeitung erforderlich. Wenn Sie für eine Liste anderer Tabellenbereiche eine weitere Operation zur aktualisierenden Recovery starten, bevor die laufende Operation beendet oder abgebrochen wurde, wird eine Fehlernachricht (SQL4908) zurückgegeben. Rufen Sie für alle Datenbankpartitionen den Befehl LIST TABLESPACES auf, um

festzustellen, welche Tabellenbereiche aktualisierend wiederhergestellt werden (Status *Aktualisierende Recovery läuft*) und für welche Tabellenbereiche die aktualisierende Recovery noch durchgeführt werden muss (Status *Aktualisierende Recovery anstehend*). Sie haben folgende Möglichkeiten:

  – Beenden Sie die aktuelle Operation zur aktualisierenden Recovery für alle Tabellenbereiche.
  – Beenden Sie die aktuelle Operation zur aktualisierenden Recovery für eine Untergruppe von Tabellenbereichen. (Dies ist evtl. nicht möglich, wenn die Operation zur aktualisierenden Recovery bis zu einem bestimmten Zeitpunkt ausgeführt werden soll, wofür die Beteiligung aller Datenbankpartitionen erforderlich ist.)
  – Brechen Sie die aktuelle Operation zur aktualisierenden Recovery ab.

- In einer Umgebung mit partitionierten Datenbanken muss das Dienstprogramm zur aktualisierenden Recovery von der Katalogpartition der Datenbank aus aufgerufen werden.
- Die zeitpunktgesteuerte aktualisierende Recovery eines Tabellenbereichs ist nur für DB2-Clients der Version 9 verfügbar. Sie sollten für sämtliche Clients, die mit einer früheren Version des Datenbankprodukts ausgeführt werden, eine Migration auf Version 9 ausführen, damit für einen Tabellenbereich eine zeitpunktgesteuerte aktualisierende Recovery durchgeführt werden kann.
- Für Protokolle aus einem früheren Release kann keine aktualisierende Recovery durchgeführt werden.

Das Dienstprogramm für die aktualisierende Recovery kann über den Befehlszeilenprozessor (CLP), den Restoreassistenten in der Steuerzentrale oder die Anwendungsprogrammierschnittstelle db2Rollforward aufgerufen werden.

Das folgende Beispiel zeigt einen Befehl ROLLFORWARD DATABASE, der über den CLP abgesetzt wird:

```
db2 rollforward db sample to end of logs and stop
```

Gehen Sie wie folgt vor, um den Restoreassistenten zu öffnen:

1. Erweitern Sie in der Steuerzentrale die Objektbaumstruktur, bis das Datenbank- oder Tabellenbereichsobjekt angezeigt wird, das Sie wiederherstellen möchten.
2. Klicken Sie mit der rechten Maustaste das Objekt an, und wählen Sie **Aktualisierend wiederherstellen** im Kontextmenü aus. Der Assistent zur aktualisierenden Recovery wird geöffnet.

Detaillierte Informationen hierzu können Sie über die Kontexthilfe der Steuerzentrale aufrufen.

Aktualisierende Recovery von Änderungen in einem Tabellenbereich

Wenn die Datenbank zur aktualisierenden Recovery aktiviert ist, können Sie Backups, Restores und aktualisierende Recoverys nicht nur für die gesamte Datenbank, sondern auch für Tabellenbereiche durchführen. Es kann sinnvoll sein, eine Recoverystrategie für einzelne Tabellenbereiche zu implementieren, da sich dadurch möglicherweise Zeit einsparen lässt: Eine Recovery eines Teils der Datenbank benötigt weniger Zeit als eine Recovery der gesamten Datenbank. Wenn zum Beispiel eine Festplatte fehlerhaft ist und nur einen Tabellenbereich enthält, kann der Tabellenbereich von einem Backup wiederhergestellt und aktualisierend wiederhergestellt werden, ohne dass die gesamte Datenbank wiederhergestellt werden muss und ohne den Benutzerzugriff auf die übrigen Teile der Datenbank zu beeinträchti-

gen. Wenn der beschädigte Tabellenbereich jedoch die Systemkatalogtabellen enthält, können Sie in diesem Fall keine Verbindung zur Datenbank herstellen. (Der Tabellenbereich mit den Systemkatalogtabellen kann unabhängig von der Datenbank wiederhergestellt werden, wenn für diesen Tabellenbereich ein Backup auf Tabellenbereichsebene verfügbar ist.) Außerdem bieten Backups auf Tabellenbereichsebene die Möglichkeit, kritische Teile der Datenbank häufiger als andere Teile zu sichern, und sind weniger zeitintensiv als Backups der gesamten Datenbank.

Nach dem Restore eines Tabellenbereichs befindet sich dieser immer im Status *Aktualisierende Recovery anstehend*. Um den Tabellenbereich verwendbar zu machen, müssen Sie eine aktualisierende Recovery für ihn durchführen. In den meisten Fällen haben Sie dabei die Möglichkeit, die aktualisierende Recovery bis zum Ende der Protokolldateien oder bis zu einem bestimmten Zeitpunkt auszuführen. Eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt kann jedoch nicht für Tabellenbereiche ausgeführt werden, die Systemkatalogtabellen enthalten. Diese Tabellen müssen bis zum Ende der Protokolle aktualisierend wiederhergestellt werden, um sicherzustellen, dass alle Tabellenbereiche in der Datenbank konsistent bleiben.

Wenn ein Tabellenbereich aktualisierend wiederhergestellt wird, verarbeitet der DB2-Datenbankmanager alle Protokolldateien, auch wenn sie keine Protokollsätze enthalten, die diesen Tabellenbereich betreffen. Wenn Sie möchten, dass die Protokolldateien übersprungen werden, die keine Protokollsätze für diesen Tabellenbereich enthalten, setzen Sie die Registrierdatenbankvariable DB2_COLLECT_TS_REC_INFO auf *ON*. Dies ist der Standardwert. Die Registrierdatenbankvariable muss vor der Erstellung und Verwendung der Protokolldateien gesetzt werden, um sicherzustellen, dass die zum Überspringen der Protokolldateien erforderlichen Informationen erfasst werden.

Die im Datenbankverzeichnis vorhandene Protokolldatei der Tabellenbereichsänderungen (DB2TSCHG.HIS) enthält Informationen dazu, welche Protokolle für die einzelnen Tabellenbereiche verarbeitet werden sollten. Sie können den Inhalt dieser Datei mithilfe des Dienstprogramms db2logsForRfwd anzeigen und mit dem Befehl PRUNE HISTORY Einträge aus der Datei löschen. Während einer Restoreoperation für eine Datenbank wird die Datei DB2TSCHG.HIS aus dem Backup-Image wiederhergestellt und während der aktualisierenden Recovery aktualisiert. Falls für eine Protokolldatei keine Informationen verfügbar sind, wird die Datei behandelt, als ob sie für die Recovery aller Tabellenbereiche erforderlich wäre.

Da die Informationen für jede Protokolldatei nach der Inaktivierung des Protokolls auf Platte geschrieben werden, können diese Informationen infolge eines Absturzes verloren gehen. Wenn eine Recoveryoperation in der Mitte einer Protokolldatei beginnt, wird daher zur Kompensation das gesamte Protokoll behandelt, als ob es Änderungen an allen Tabellenbereichen im System enthielte. Anschließend werden die aktiven Protokolle verarbeitet, und die Informationen zu diesen Protokollen werden erneut erstellt. Falls Informationen für ältere Protokolle oder archivierte Protokolldateien bei einem Absturz verloren gehen und in der Datendatei keine Informationen für die Protokolle vorhanden sind, werden die Protokolldateien während der Recoveryoperation für den Tabellenbereich behandelt, als ob sie Änderungen für alle Tabellenbereiche enthielten.

Führen Sie vor der aktualisierenden Recovery eines Tabellenbereichs den Befehl LIST TABLESPACES SHOW DETAIL aus. Dieser Befehl gibt den *Mindestrecoveryzeitpunkt* zurück, d. h. den frühesten Zeitpunkt, bis zu dem der Tabellenbereich aktualisierend wiederhergestellt werden kann. Der Mindestrecoveryzeitpunkt wird aktualisiert, wenn DDL-Anweisungen (DDL - Datendefinitionssprache) für den Ta-

bellenbereich oder für Tabellen in diesem Tabellenbereich ausgeführt werden. Der Tabellenbereich muss mindestens bis zum Mindestrecoveryzeitpunkt aktualisierend wiederhergestellt werden, um mit den Informationen in den Systemkatalogtabellen synchron zu sein. Wenn Sie mehr als einen Tabellenbereich wiederherstellen, müssen die Tabellenbereiche bis zum frühesten Mindestrecoveryzeitpunkt für alle Tabellenbereiche aktualisierend wiederhergestellt werden. Tabellenbereiche können nicht bis zu einem Zeitpunkt aktualisierend wiederhergestellt werden, der vor der Zeitmarke des Backups liegt. Setzen Sie in einer Umgebung mit partitionierten Datenbanken für alle Datenbankpartitionen den Befehl LIST TABLESPACES SHOW DETAIL ab. Die Tabellenbereiche müssen bis zum frühesten Mindestrecoveryzeitpunkt für alle Tabellenbereiche auf allen Datenbankpartitionen aktualisierend wiederhergestellt werden.

Wenn Sie Tabellenbereiche bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen und eine Tabelle in mehreren Tabellenbereichen enthalten ist, müssen alle Tabellenbereiche, die die Tabelle enthalten, gleichzeitig aktualisierend wiederhergestellt werden. Wenn zum Beispiel die Tabellendaten in einem Tabellenbereich gespeichert sind und der Index für die Tabelle sich in einem anderen Tabellenbereich befindet, müssen beide Tabellenbereiche gleichzeitig bis zum selben Zeitpunkt aktualisierend wiederhergestellt werden.

Wenn die Daten und langen Objekte einer Tabelle in getrennten Tabellenbereichen gespeichert sind und die LOB-Daten reorganisiert wurden, müssen die Tabellenbereiche sowohl für die Daten als auch für die langen Objekte gemeinsam von einem Backup wiederhergestellt und aktualisierend wiederhergestellt werden. Es ist ratsam, nach dem Reorganisieren der Tabelle ein Backup der betroffenen Tabellenbereiche zu erstellen.

Angenommen, Sie wollen einen Tabellenbereich bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen, und eine Tabelle im Tabellenbereich entspricht einem der beiden folgenden Typen:

- Eine zugrunde liegende Tabelle für eine MQT (Materialized Query Table - gespeicherte Abfragetabelle) oder Zwischenspeichertabelle, die sich in einem anderen Tabellenbereich befindet
- Eine MQT oder Zwischenspeichertabelle für eine Tabelle, die sich in einem anderen Tabellenbereich befindet

In diesem Fall müssen Sie beide Tabellenbereiche bis zum selben Zeitpunkt aktualisierend wiederherstellen. Wenn Sie dies nicht tun, wird die MQT oder Zwischenspeichertabelle am Ende der aktualisierenden Recovery in den Status *Festlegen der Integrität anstehend* versetzt. Die MQT muss vollständig aktualisiert werden, und die Zwischenspeichertabelle wird als unvollständig markiert.

Wenn Sie einen Tabellenbereich bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen wollen und eine Tabelle in dem Tabellenbereich an einer referenziellen Integritätsbeziehung mit einer anderen Tabelle beteiligt ist, die in einem anderen Tabellenbereich enthalten ist, sollten Sie beide Tabellenbereiche gleichzeitig bis zum selben Zeitpunkt aktualisierend wiederherstellen. Wenn Sie dies nicht tun, wird die untergeordnete Tabelle in der referenziellen Integritätsbeziehung am Ende der aktualisierenden Recovery in den Status *Festlegen der Integrität anstehend* versetzt. Wenn die untergeordnete Tabelle später auf ungültige Integritätsbedingungen hin überprüft wird, ist eine Überprüfung der gesamten Tabelle erforderlich.

Wenn eine der folgenden Tabellen existiert, werden sie ebenfalls zusammen mit der untergeordneten Tabelle in den Status Festlegen der Integrität anstehend versetzt:

- Jede untergeordnete MQT für die untergeordnete Tabelle
- Jede untergeordnete Zwischenspeichertabelle für die untergeordnete Tabelle
- Jede untergeordnete Fremdschlüsseltabelle der untergeordneten Tabelle

Wenn Sie diese Tabellen aus dem Status *Festlegen der Integrität anstehend* herausnehmen wollen, ist eine vollständige Integritätsverarbeitung erforderlich. Wenn Sie beide Tabellenbereiche zur selben Zeit aktualisierend wiederherstellen, bleibt die Integritätsbedingung am Ende der aktualisierenden Recovery bis zu einem bestimmten Zeitpunkt aktiv.

Stellen Sie sicher, dass die aktualisierende Recovery von Tabellenbereichen bis zu einem bestimmten Zeitpunkt nicht dazu führt, dass eine Transaktion in einigen Tabellenbereichen rückgängig gemacht und in anderen festgeschrieben wird. Dies kann in den folgenden Fällen geschehen:

- Eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt wird für eine Untergruppe der Tabellenbereiche durchgeführt, die durch eine Transaktion aktualisiert wurden, und dieser Zeitpunkt liegt vor dem Zeitpunkt, an dem die Transaktion festgeschrieben wurde.
- Eine Tabelle, die in dem Tabellenbereich enthalten ist, der bis zu einem bestimmten Zeitpunkt aktualisierend wiederhergestellt wird, hat einen zugeordneten Trigger oder wird von einem Trigger aktualisiert, der auf andere Tabellenbereiche als den zugreift, der aktualisierend wiederhergestellt wird.

Sie sollten einen geeigneten Zeitpunkt suchen, der dies verhindert.

Sie können den Befehl QUIESCE TABLESPACES FOR TABLE absetzen, um einen transaktionskonsistenten Zeitpunkt für die aktualisierende Recovery von Tabellenbereichen zu erstellen. Diese Wartezeitanforderung (im Modus SHARE, INTENT TO UPDATE oder EXCLUSIVE) wartet durch Sperrung, bis alle aktiven Transaktionen für diese Tabellenbereiche beendet wurden und blockiert neue Anforderungen. Wenn die Wartezeitanforderung bestätigt wird, befinden sich die Tabellenbereiche in einem konsistenten Status. Sie können in der Datei des Recoveryprotokolls nach Wartezeitpunkten suchen und prüfen, ob sie nach dem Mindestrecoveryzeitpunkt liegen, um einen geeigneten Zeitpunkt für den Stopp der aktualisierenden Recovery festzulegen.

Nach Beendigung der aktualisierenden Recovery bis zu einem bestimmten Zeitpunkt wird der Tabellenbereich wieder in den Status *Backup anstehend* versetzt. Sie müssen ein Backup des Tabellenbereichs erstellen, weil alle Aktualisierungen für den Zeitraum zwischen dem Zeitpunkt, bis zu dem die aktualisierende Recovery erfolgte, und dem aktuellen Zeitpunkt entfernt wurden. Sie können den Tabellenbereich nicht mehr von einem vorherigen Backup der Datenbank bzw. des Tabellenbereichs bis zum aktuellen Zeitpunkt aktualisierend wiederherstellen. Das folgende Beispiel zeigt, warum das Backup des Tabellenbereichs erforderlich ist und wie es verwendet wird. (Um den Tabellenbereich verfügbar zu machen, können Sie entweder die gesamte Datenbank sichern, oder nur den Tabellenbereich, der den Status *Backup anstehend* hat, oder eine Gruppe von Tabellenbereichen, die diesen letztgenannten Tabellenbereich enthält.)

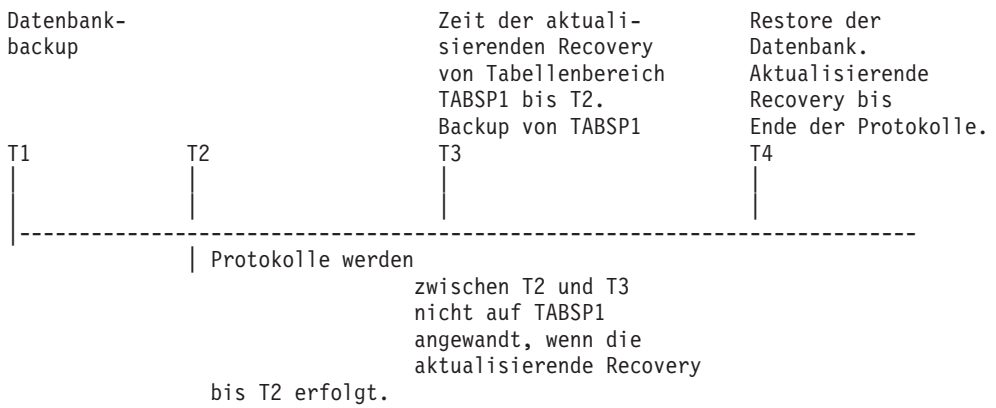


Abbildung 26. Backup-Anforderung für Tabellenbereiche

Im vorstehenden Beispiel wird die Datenbank zum Zeitpunkt T1 gesichert. Anschließend erfolgt für den Tabellenbereich TABSP1 zum Zeitpunkt T3 eine aktualisierende Recovery bis zu einem bestimmten Zeitpunkt (T2). Der Tabellenbereich wird nach Zeitpunkt T3 gesichert. Da sich der Tabellenbereich im Status *Backup anstehend* befindet, muss ein Backup-Image erstellt werden. Die Zeitmarke des Backup-Images des Tabellenbereichs weist eine Zeit nach T3 aus, während sich der Tabellenbereich in dem Status von Zeitpunkt T2 befindet. Die Protokollsätze für den Zeitraum zwischen T2 und T3 werden auf TABSP1 nicht angewendet. Die Datenbank wird unter Verwendung des bei T1 erstellten Backup-Images zum Zeitpunkt T4 wiederhergestellt und bis zum Ende der Protokolle aktualisierend wiederhergestellt. Der Tabellenbereich TABSP1 wird zum Zeitpunkt T3 in den Status *Restore anstehend* versetzt, da der Datenbankmanager annimmt, dass zwischen T3 und T4 Operationen an TABSP1 ausgeführt wurden, ohne dass die Protokolländerungen zwischen T2 und T3 auf den Tabellenbereich angewendet wurden. Wären diese Protokolländerungen tatsächlich als Teil der aktualisierenden Recovery der Datenbank angewendet worden, wäre diese Annahme falsch. Das erforderliche Backup eines Tabellenbereichs, das nach der aktualisierenden Recovery bis zu einem bestimmten Zeitpunkt erstellt werden muss, ermöglicht es, diesen Tabellenbereich bis nach dem Zeitpunkt einer vorherigen aktualisierenden Recovery (in diesem Beispiel T3) aktualisierend wiederherzustellen.

Wenn Sie den Tabellenbereich TABSP1 beispielsweise bis zum Zeitpunkt T4 wiederherstellen wollen, würden Sie den Tabellenbereich von einem Backup wiederherstellen, das nach T3 erstellt wurde (entweder das erforderliche Backup oder ein anderes), und anschließend den Tabellenbereich TABSP1 bis zum Ende der Protokolle aktualisierend wiederherstellen.

Im vorangegangenen Beispiel wäre die effizienteste Vorgehensweise zum Restore des Tabellenbereichs bis zum Zeitpunkt T4 die Ausführung der erforderlichen Schritte in folgender Reihenfolge:

1. Stellen Sie die Datenbank von einem Backup wieder her.
2. Stellen Sie den Tabellenbereich von einem Backup wieder her.
3. Stellen Sie die Datenbank aktualisierend wieder her.
4. Stellen Sie den Tabellenbereich aktualisierend wieder her.

Da Sie den Tabellenbereich vor der aktualisierenden Recovery der Datenbank von einem Backup wiederherstellen, werden keine Ressourcen zum Anwenden der Protokollsätze auf den Tabellenbereich verwendet, wenn die Datenbank aktualisierend wiederhergestellt wird.

Wenn Sie das Backup-Image von TABSP1 für einen Zeitpunkt nach T3 nicht mehr finden können oder den Tabellenbereich TABSP1 auf einem Stand vor oder bis T3 wiederherstellen wollen, haben Sie folgende Möglichkeiten:

- Stellen Sie den Tabellenbereich aktualisierend wieder her bis zum Zeitpunkt T3. Sie brauchen den Tabellenbereich nicht erneut wiederherzustellen, weil er vom Datenbank-Backup wiederhergestellt wurde.
- Stellen Sie den Tabellenbereich erneut von dem Datenbank-Backup-Image wieder her, das Sie zum Zeitpunkt T1 erstellt haben, und stellen Sie anschließend den Tabellenbereich bis zu einem Zeitpunkt vor T3 aktualisierend wieder her.
- Löschen Sie den Tabellenbereich.

In einer Umgebung mit partitionierten Datenbanken müssen Sie Folgendes beachten:

- Sie müssen alle Teile des Tabellenbereichs bis zum selben Zeitpunkt gleichzeitig aktualisierend wiederherstellen. Dadurch wird sichergestellt, dass der Tabellenbereich datenbankpartitionsübergreifend konsistent ist.
- Wenn sich einige Datenbankpartitionen im Status *Aktualisierende Recovery anstehend* befinden und in anderen Datenbankpartitionen einige Tabellenbereiche ebenfalls diesen Status haben (die Datenbankpartitionen selbst jedoch nicht), müssen Sie zuerst die Datenbankpartitionen und anschließend die Tabellenbereiche aktualisierend wiederherstellen.
- Wenn Sie einen Tabellenbereich bis zum Ende der Protokolle aktualisierend wiederherstellen wollen, müssen Sie ihn nicht in jeder Datenbankpartition wiederherstellen, sondern nur in den Partitionen, für die eine Recovery erforderlich ist. Wenn Sie einen Tabellenbereich jedoch bis zu einem bestimmten Zeitpunkt aktualisierend wiederherstellen wollen, müssen Sie ihn in jeder Datenbankpartition wiederherstellen.

In einer Datenbank mit partitionierten Tabellen:

- Wenn Sie einen Tabellenbereich, der einen Teil einer partitionierten Tabelle enthält, aktualisierend bis zu einem bestimmten Zeitpunkt wiederherstellen, müssen Sie auch alle anderen Tabellenbereiche, in denen sich die Tabelle befindet, bis zu demselben Zeitpunkt aktualisierend wiederherstellen. Die aktualisierende Recovery eines einzelnen Tabellenbereichs, der einen Teil einer partitionierten Tabelle enthält, bis zum Ende der Protokolle ist jedoch zulässig. Wenn eine partitionierte Tabelle (mit ATTACH) zugeordnete, (mit DETACH) getrennte oder (mit DROP) gelöschte Datenpartitionen hat, muss die aktualisierende Recovery bis zu einem bestimmten Zeitpunkt auch alle Tabellenbereiche für diese Datenpartitionen berücksichtigen. Um zu bestimmen, ob eine partitionierte Tabelle zugeordnete, getrennte oder gelöschte Datenpartitionen hat, fragen Sie die Katalogsicht SYSCAT.DATAPARTITIONS ab.

Erforderliche Berechtigungen für aktualisierende Recovery

Zugriffsrechte ermöglichen es Benutzern, Datenbankressourcen zu erstellen oder auf sie zuzugreifen. Berechtigungsstufen stellen eine Methode dar, um Berechtigungen sowie übergeordnete Pflege- und Dienstprogrammoperationen des Datenbankmanagers zusammenzufassen. Sie dienen zusammen zur Steuerung des Zugriffs auf den Datenbankmanager und seine Datenbankobjekte. Benutzer können nur auf die Objekte zugreifen, für die sie zugriffsberechtigt sind, d. h. für die sie über das erforderliche Zugriffsrecht oder die erforderliche Berechtigung verfügen.

Sie benötigen die Berechtigung SYSADM, SYSCTRL oder SYSMAINT, um das Dienstprogramm zur aktualisierenden Recovery verwenden zu können.

Sitzungen zur aktualisierenden Recovery - Beispiele für CLP

Beispiel 1

Der Befehl ROLLFORWARD DATABASE ermöglicht die Spezifikation mehrerer Operationen gleichzeitig, die jeweils mit dem Schlüsselwort AND voneinander getrennt werden müssen. Sie benötigen z. B. die folgenden Einzelbefehle, um eine aktualisierende Recovery bis zum Ende der Protokolle auszuführen und zu beenden:

```
db2 rollforward db sample to end of logs
db2 rollforward db sample complete
```

Diese Befehle können wie folgt kombiniert werden:

```
db2 rollforward db sample to end of logs and complete
```

Obwohl die beiden Befehle gleichbedeutend sind, wird empfohlen, solche Operationen in zwei Schritten auszuführen. Es ist wichtig, zu überprüfen, ob die aktualisierende Recoveryoperation wie erwartet fortgeschritten ist, bevor Sie sie stoppen, sodass keine Protokolle fehlen.

Wenn der ROLLFORWARD-Befehl auf einen Fehler stößt, wird die aktualisierende Recoveryoperation nicht beendet. Der Fehler wird zurückgegeben, sodass Sie den Fehler beheben und den Befehl erneut ausführen können. Wenn Sie den Fehler jedoch nicht beheben können, können Sie die Beendigung der aktualisierenden Recovery mit dem folgenden Befehl erzwingen:

```
db2 rollforward db sample complete
```

Dieser Befehl macht die Datenbank an dem Punkt in den Protokollen vor dem Fehler online verfügbar.

Beispiel 2

Geben Sie zur Ausführung einer aktualisierenden Recovery der Datenbank bis zum Protokollende (zwei Tabellenbereiche wurden wiederhergestellt) folgende Befehle ein:

```
db2 rollforward db sample to end of logs
db2 rollforward db sample to end of logs and stop
```

Diese beiden Anweisungen sind gleichbedeutend. Weder AND STOP noch AND COMPLETE sind für eine aktualisierende Tabellenbereichsrecovery bis zum Protokollende erforderlich. Tabellenbereichsnamen sind nicht erforderlich. Falls keine Namen angegeben werden, werden alle Tabellenbereiche, für die eine aktualisierende Recovery erforderlich ist, mit einbezogen. Wenn nur eine Untermenge dieser Tabellenbereiche aktualisierend wiederhergestellt werden soll, müssen die Namen der Tabellenbereiche angegeben werden.

Beispiel 3

Führen Sie, nachdem drei Tabellenbereiche wiederhergestellt wurden, für einen Tabellenbereich die aktualisierende Recovery bis zum Ende der Protokolldateien und für die anderen beiden bis zu einem bestimmten Zeitpunkt aus. Beides muss online erfolgen:

```
db2 rollforward db sample to end of logs tablespace(TBS1) online
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
    tablespace(TBS2, TBS3) online
```

Beachten Sie, dass zwei aktualisierende Recoveryoperationen nicht gleichzeitig ausgeführt werden können. Sie können den zweiten Befehl erst aufrufen, nachdem die erste aktualisierende Recoveryoperation erfolgreich beendet wurde.

Beispiel 4

Führen Sie nach dem Restore der Datenbank die aktualisierende Recovery bis zu einem bestimmten Zeitpunkt unter Verwendung von OVERFLOW LOG PATH aus, womit Sie das Verzeichnis angeben, indem das Benutzerexitprogramm archivierte Protokolldateien speichert:

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
         overflow log path (/logs)
```

Beispiel 5

Im folgenden Beispiel wird angenommen, dass eine Datenbank namens 'sample' vorhanden ist. Für die Datenbank wird ein Backup durchgeführt, und die Recoveryprotokolle werden in das Backup-Image eingefügt; die Datenbank wird wiederhergestellt; dann wird eine aktualisierende Recovery bis zum Ende der Zeitmarke des Backups ausgeführt.

Durchführen eines Backups der Datenbank, einschließlich der Recoveryprotokolle im Backup-Image:

```
db2 backup db sample online include logs
```

Wiederherstellen der Datenbank mit diesem Backup-Image:

```
db2 restore db sample
```

Ausführen einer aktualisierenden Recovery der Datenbank bis zum Ende der Zeitmarke des Backups:

```
db2 rollforward db sample to end of backup
```

Beispiel 6 (Umgebungen mit partitionierten Datenbanken)

Es gibt drei Datenbankpartitionen: 0, 1 und 2. Der Tabellenbereich TBS1 ist in allen Datenbankpartitionen definiert, der Tabellenbereich TBS2 nur in den Datenbankpartitionen 0 und 2. Nachdem Sie die Datenbank in Datenbankpartition 1 und TBS1 in den Datenbankpartitionen 0 und 2 wiederhergestellt haben, führen Sie eine aktualisierende Recovery der Datenbank in Datenbankpartition 1 wie folgt aus:

```
db2 rollforward db sample to end of logs and stop
```

Dieser Befehl gibt die Warnung SQL1271 („Die Datenbank "name" wurde wiederhergestellt. Auf dem bzw. den Knoten 0 und 2 ist jedoch mindestens ein Tabellenbereich offline.”) zurück.

```
db2 rollforward db sample to end of logs
```

Durch diesen Befehl wird TBS1 in den Datenbankpartitionen 0 und 2 aktualisierend wiederhergestellt. In diesem Fall ist die Klausel TABLESPACE(TBS1) optional.

Beispiel 7 (Umgebungen mit partitionierten Datenbanken)

Im folgenden Beispiel wird angenommen, dass eine partitionierte Datenbank namens 'sample' vorhanden ist. Für alle Datenbankpartitionen wird ein SSV-Backup (SSV = Single System View, Einzelsystemsicht) durchgeführt; die Datenbank wird

auf allen Datenbankpartitionen wiederhergestellt; dann wird eine aktualisierende Recovery der Datenbank bis zum Ende der Zeitmarke des Backups ausgeführt.

Durchführen eines SSV-Backups:

```
db2 backup db sample on all nodes online include logs
```

Wiederherstellen der Datenbank auf allen Datenbankpartitionen:

```
db2_all "db2 restore db sample taken at 1998-04-03-14.21.56"
```

Ausführen einer aktualisierenden Recovery der Datenbank bis zum Ende der Zeitmarke des Backups:

```
db2 rollforward db sample to end of backup on all nodes
```

Beispiel 8 (Umgebungen mit partitionierten Datenbanken)

Im folgenden Beispiel wird angenommen, dass eine partitionierte Datenbank namens 'sample' vorhanden ist. Für alle Datenbankpartitionen wird mit einem Befehl ein Backup unter Verwendung von 'db2_all' durchgeführt; die Datenbank wird auf allen Datenbankpartitionen wiederhergestellt; dann wird eine aktualisierende Recovery der Datenbank zum Ende der Zeitmarke des Backups ausgeführt.

Durchführen eines Backups für alle Datenbankpartitionen mit einem Befehl unter Verwendung von 'db2_all':

```
db2_all "db2 backup db sample include logs to /shared/dir/"
```

Wiederherstellen der Datenbank auf allen Datenbankpartitionen:

```
db2_all "db2 restore db sample from /shared/dir/"
```

Ausführen einer aktualisierenden Recovery der Datenbank bis zum Ende der Zeitmarke des Backups:

```
db2 rollforward db sample to end of backup on all nodes
```

Beispiel 9 (Umgebungen mit partitionierten Datenbanken)

Nachdem Sie Tabellenbereich TBS1 nur in den Datenbankpartitionen 0 und 2 wiederhergestellt haben, stellen Sie TBS1 wie folgt in den Datenbankpartitionen 0 und 2 aktualisierend wieder her:

```
db2 rollforward db sample to end of logs
```

Datenbankpartition 1 wird ignoriert.

```
db2 rollforward db sample to end of logs tablespace(TBS1)
```

Dieser Befehl schlägt fehl, weil TBS1 auf Datenbankpartition 1 nicht für eine aktualisierende Recovery bereit ist. SQL4906N wird dokumentiert.

```
db2 rollforward db sample to end of logs on
         dbpartitionnums (0, 2) tablespace(TBS1)
```

Dieser Befehl wird erfolgreich beendet.

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
         tablespace(TBS1)
```

Dieser Befehl schlägt fehl, weil TBS1 in Datenbankpartition 1 nicht für eine aktualisierende Recovery bereit ist. Alle Teile müssen gemeinsam aktualisierend wiederhergestellt werden.

Anmerkung: Bei der aktualisierenden Tabellenbereichsrecovery bis zu einem Zeitpunkt wird die DBPARTITIONNUM-Klausel nicht akzeptiert. Die aktualisierende Recoveryoperation muss in allen Datenbankpartitionen ausgeführt werden, in denen sich der Tabellenbereich befindet.

Geben Sie nach dem Restore von TBS1 in Datenbankpartition 1 Folgendes ein:

```
db2 rollforward db sample to 1998-04-03-14.21.56 and stop
         tablespace(TBS1)
```

Dieser Befehl wird erfolgreich beendet.

Beispiel 10 (Umgebungen mit partitionierten Datenbanken)

Nachdem Sie einen Tabellenbereich in allen Datenbankpartitionen wiederhergestellt haben, stellen Sie ihn wie folgt aktualisierend bis zum Zeitpunkt PIT2 wieder her. Geben Sie jedoch AND STOP nicht an. Die aktualisierende Recoveryoperation wird weiterhin ausgeführt. Brechen Sie sie wie folgt ab, und führen Sie eine aktualisierende Recovery bis zum Zeitpunkt PIT1 aus:

```
db2 rollforward db sample to pit2 tablespace(TBS1)
   db2 rollforward db sample cancel tablespace(TBS1)

** TBS1 in allen Datenbankpartitionen wiederherstellen **

   db2 rollforward db sample to pit1 tablespace(TBS1)
   db2 rollforward db sample stop tablespace(TBS1)
```

Beispiel 11 (Umgebungen mit partitionierten Datenbanken)

Stellen Sie wie folgt einen Tabellenbereich wieder her, der sich in den in der Datei db2nodes.cfg aufgelisteten acht Datenbankpartitionen (3 bis 10) befindet:

```
db2 rollforward database dwtest to end of logs tablespace (tssprodt)
```

Diese Operation bis zum Protokollende (nicht bis zu einem Zeitpunkt) wird erfolgreich beendet. Die Datenbankpartitionen, in denen sich die Tabellenbereiche befinden, müssen nicht angegeben werden. Das Dienstprogramm verwendet standardmäßig die Datei db2nodes.cfg.

Beispiel 12 (Umgebungen mit partitionierten Datenbanken)

Stellen Sie wie folgt sechs kleine Tabellenbereiche aktualisierend wieder her, die sich in einer Datenbankpartitionsgruppe mit nur einer Datenbankpartition befinden (in Datenbankpartition 6):

```
db2 rollforward database dwtest to end of logs on dbpartitionnum (6)
         tablespace(tsstore, tssbuyer, tsstime, tsswhse, tsslscat, tssvendor)
```

Diese Operation bis zum Protokollende (nicht bis zu einem Zeitpunkt) wird erfolgreich beendet.

Beispiel 13 (partitionierte Tabellen - aktualisierende Recovery bis zum Ende der Protokolle in allen Datenbankpartitionen)

Eine partitionierte Tabelle wird in den Tabellenbereichen tbsp1, tbsp2, tbsp3 mit einem Index in tbsp0 erstellt. Später fügt ein Benutzer der Tabelle Datenpartitionen in tbsp4 hinzu und ordnet (ATTACH) Datenpartitionen aus der Tabelle in tbsp5 zu. Alle Tabellenbereiche können bis zum Ende der Protokolle aktualisierend wiederhergestellt werden.

```
db2 rollforward db PBARDB to END OF LOGS and stop
        tablespace(tbsp0, tbsp1, tbsp2, tbsp3, tbsp4, tbsp5)
```

Dieser Befehl wird erfolgreich beendet.

Beispiel 14 (partitionierte Tabellen - aktualisierende Recovery bis zum Ende der Protokolle in einem Tabellenbereich)

Eine partitionierte Tabelle wird zu Anfang in den Tabellenbereichen tbsp1, tbsp2, tbsp3 mit einem Index in tbsp0 erstellt. Später fügt ein Benutzer der Tabelle Datenpartitionen in tbsp4 hinzu und ordnet (ATTACH) Datenpartitionen aus der Tabelle in tbsp5 zu. Der Tabellenbereich tbsp4 wird beschädigt und erfordert einen Restore sowie eine aktualisierende Recovery bis zum Ende der Protokolle.

```
db2 rollforward db PBARDB to END OF LOGS and stop tablespace(tbsp4)
```

Dieser Befehl wird erfolgreich beendet.

Beispiel 15 (partitionierte Tabellen - aktualisierende Recovery bis zu einem Zeitpunkt aller Datenpartitionen einschließlich der hinzugefügten (ADD), zugeordneten (ATTACH), getrennten (DETACH) oder derer mit Indizes)

Eine partitionierte Tabelle wird in den Tabellenbereichen tbsp1, tbsp2, tbsp3 mit einem Index in tbsp0 erstellt. Später fügt ein Benutzer der Tabelle Datenpartitionen in tbsp4 hinzu, ordnet (ATTACH) Datenpartitionen aus der Tabelle in tbsp5 zu und hebt die Zuordnung von Datenpartitionen aus tbsp1 auf (DETACH). Der Benutzer führt eine aktualisierende Recovery bis zu einem Zeitpunkt mit allen Tabellenbereichen aus, die von der partitionierten Tabelle verwendet werden, einschließlich der Tabellenbereiche, die in der Klausel INDEX IN angegeben wurden.

```
db2 rollforward db PBARDB to 2005-08-05-05.58.53 and stop
        tablespace(tbsp0, tbsp1, tbsp2, tbsp3, tbsp4, tbsp5)
```

Dieser Befehl wird erfolgreich beendet.

Beispiel 16 (partitionierte Tabellen - aktualisierende Recovery bis zu einem Zeitpunkt für eine Untergruppe der Tabellenbereiche)

Eine partitionierte Tabelle wird in drei Tabellenbereichen (tbsp1, tbsp2, tbsp3) erstellt. Später hebt der Benutzer die Zuordnung aller Datenpartitionen aus tbsp3 auf. Die aktualisierende Recovery bis zu einem Zeitpunkt ist nur für die Tabellenbereiche tbsp1 und tbsp2 zulässig.

```
db2 rollforward db PBARDB to 2005-08-05-06.02.42 and stop
         tablespace( tbsp1, tbsp2)
```

Dieser Befehl wird erfolgreich beendet.

Kapitel 14. Datenrecovery mit IBM Tivoli Storage Manager (TSM)

Wenn Sie den Befehl BACKUP DATABASE oder RESTORE DATABASE aufrufen, können Sie angeben, dass Sie die Backup- oder Restoreoperation für die Datenbank bzw. den Tabellenbereich mit dem Produkt IBM Tivoli Storage Manager (TSM) verwalten wollen. Der erforderliche Mindeststand der TSM-Client-API ist Version 4.2.0 mit Ausnahme folgender Systeme, die eine andere Version erfordern:

- 64-Bit-Solaris-Systeme, für die Version 4.2.1 der TSM-Client-API erforderlich ist.
- 64-Bit-Windows-Betriebssysteme, für die Version 5.1 der TSM-Client-API erforderlich ist.
- Alle Windows X64-Systeme, für die Version 5.3.2 der TSM-Client-API erforderlich ist.
- 32-Bit-Linux für System i und pSeries, für die mindestens Version 5.1.5 der TSM-Client-API erforderlich ist.
- 64-Bit-Linux für System i und pSeries, für die mindestens Version 5.2.2 der TSM-Client-API erforderlich ist.
- 64-Bit-Linux auf AMD Opteron-Systeme, für die mindestens Version 5.2.0 der TSM-Client-API erforderlich ist.
- Linux für zSeries-Systeme, für die mindestens Version 5.2.2 der TSM-Client-API erforderlich ist.

Konfigurieren eines Tivoli Storage Manager-Clients

Bevor der DB2-Datenbankmanager einen TSM-Client (TSM = IBM Tivoli Storage Manager) zur Verwaltung von Backup- bzw. Restoreoperationen für Datenbanken oder Tabellenbereiche verwenden kann, müssen Sie die TSM-Umgebung konfigurieren.

Vorbereitung

Es muss ein funktionierender TSM-Client und -Server installiert und konfiguriert werden. Darüber hinaus muss auf jedem DB2-Datenbankserver, auf dem TSM verwendet werden soll, die TSM-Client-API installiert sein. TSM-Client-Proxy-Knoten werden unterstützt, wenn der TSM-Server entsprechend konfiguriert ist. Informationen zur Serverkonfiguration und zur Unterstützung von Proxy-Knoten finden Sie im Abschnitt „Aspekte der Verwendung von Tivoli Storage Manager” auf Seite 364 oder in der Tivoli-Dokumentation.

Vorgehensweise

Gehen Sie wie folgt vor, um die TSM-Umgebung für die Verwendung durch DB2-Datenbanksysteme zu konfigurieren:

1. Definieren Sie die von der TSM-Client-API verwendeten Umgebungsvariablen:

   DSMI_DIR
   Gibt den benutzerdefinierten Verzeichnispfad an, in dem sich die API-Datei für den Trusted Agent (`dsmtca`) befindet.

   DSMI_CONFIG
   Gibt den benutzerdefinierten Verzeichnispfad zur Datei `dsm.opt` an, die

die TSM-Benutzeroptionen enthält. Im Unterschied zu den beiden anderen Variablen müssen Sie bei dieser Variablen einen vollständig qualifizierten Pfad und einen Dateinamen angeben.

DSMI_LOG
Gibt den benutzerdefinierten Verzeichnispfad an, in dem das Fehlerprotokoll (`dsierror.log`) erstellt wird.

Anmerkung: In einer Umgebung mit partitionierter Datenbank müssen diese Einstellungen in der Datei `sqllib/userprofile` angegeben werden.

2. Falls an diesen Umgebungsvariablen Änderungen vorgenommen werden, während der Datenbankmanager ausgeführt wird, stoppen Sie den Datenbankmanager und starten Sie ihn erneut. Beispiel:
   - Stoppen Sie den Datenbankmanager mit dem Befehl db2stop.
   - Starten Sie den Datenbankmanager mit dem Befehl db2start.
3. Abhängig von der Konfiguration des Servers benötigt der Tivoli-Client für den Datenaustausch mit dem TSM-Server möglicherweise ein Kennwort.

   Falls die TSM-Umgebung für die Verwendung von `PASSWORDACCESS=generate` konfiguriert ist, muss für den Tivoli-Client ein Kennwort eingerichtet sein.

   Die ausführbare Datei `dsmapipw` ist im Verzeichnis `sqllib/adsm` des Instanzeigners installiert. Mithilfe dieser ausführbaren Datei können Sie das TSM-Kennwort definieren und zurücksetzen.

   Damit Sie den Befehl dsmapipw ausführen können, müssen Sie als lokaler Administrator oder als Root angemeldet sein. Bei der Ausführung dieses Befehls werden Sie aufgefordert, die folgenden Informationen einzugeben:
   - *Altes Kennwort*, d. h. das aktuelle (vom TSM-Server erkannte) Kennwort für den TSM-Knoten. Wenn Sie diesen Befehl zum ersten Mal ausführen, ist dies das Kennwort, das vom TSM-Administrator bei der Registrierung Ihres Knotens auf dem TSM-Server vergeben wurde.
   - *Neues Kennwort*, d. h. das neue Kennwort für den TSM-Knoten, das auf dem TSM-Server gespeichert wird. (Sie werden zweimal zur Eingabe des neuen Kennworts aufgefordert, um Eingabefehler festzustellen.)

   Anmerkung: Benutzer, die den Befehl BACKUP DATABASE oder den Befehl RESTORE DATABASE aufrufen, brauchen dieses Kennwort nicht zu kennen. Sie müssen den Befehl dsmapipw nur ausführen, wenn Sie für die erste Verbindung ein Kennwort festlegen und nachdem das Kennwort auf dem TSM-Server zurückgesetzt wurde.

Entsprechend Ihrer Strategie für die Backup- und Protokollarchivierung müssen Sie möglicherweise weitere Konfigurationsschritte für die TSM-Clients ausführen, um Proxy-Knoten zu verwenden. Proxy-Knoten ermöglichen die Konsolidierung von Backups und Protokollarchiven für Datenbanken auf mehreren Clientknoten oder für mehrere Benutzer unter einem gemeinsamen Zielknotennamen auf dem TSM-Server. Diese Konfiguration ist hilfreich, wenn das Backup von verschiedenen Administratoren oder Computern ausgeführt werden kann (z. B. innerhalb eines Clusters). Die Option asnodename ermöglicht außerdem die Datenwiederherstellung von einem anderen Computer oder Benutzer, der nicht das Backup ausgeführt hat.

Wenn die Proxy-Knoten nicht standardmäßig verwendet werden sollen, ist keine zusätzliche Clientkonfiguration erforderlich. Wenn Sie Backup- oder Restore-Operationen über Proxy-Knoten ausführen möchten, geben Sie den Wert asnodename im Parameter **OPTIONS** beim Aufrufen des Befehls BACKUP DATABASE oder RESTORE DATABASE an.

Verwenden Sie die folgenden Methoden, wenn TSM-Proxy-Knoten standardmäßig verwendet werden sollen:

- Aktualisieren Sie die Datenbankkonfigurationsparameter so, dass verschiedene Proxy-Knoten für verschiedene Datenbanken verwendet werden.
- Aktualisieren Sie die Datei dsm.sys so, dass für alle Benutzer und Datenbanken auf einer Maschine derselbe Proxy-Knoten verwendet wird. Alle Benutzer auf der Maschine werden in der Konfiguration des Proxy-Knotens wie ein eindeutiger Benutzer angezeigt. Verwenden Sie diese Methode nur, wenn jeder Benutzer bzw. jede Instanz eine separate Datenbank verwendet. Andernfalls werden möglicherweise Dateien überschrieben und es können Daten verloren gehen.

TSM-Clientkonfiguration mit VENDOROPT, LOGARCHOPT1 und LOGARCHOPT2

Sie können einen oder mehrere der folgenden Datenbankkonfigurationsparameter festlegen, um unterschiedliche Proxy-Knoteneinstellungen für die einzelnen Datenbanken zu verwenden:

- Damit Befehle in TSM (z. B. Backup und Restore) Proxy-Knoten verwenden können, geben Sie die Option asnodename im Datenbankkonfigurationsparameter *vendoropt* wie folgt an:

  ```
  db2 update db cfg for dbname using vendoropt "'-asnodename=proxy-knoten'"
  ```

  Dabei steht *proxy-knoten* für den Namen des gemeinsam genutzten TSM-Proxy-Knotens.
- Setzen Sie zum Konfigurieren der Protokollarchivierung auf dem TSM-Server den Datenbankkonfigurationsparameter *logarchmeth1* auf 'TSM' und geben Sie den Namen des Proxy-Knotens in der Option 'asnodename' des Datenbankkonfigurationsparameters *logarchopt1* wie folgt an:

  ```
  db2 update db cfg for dbname using logarchmeth1 tsm
  logarchopt1 "'-asnodename=proxy-knoten'"
  ```

  Dabei steht *proxy-knoten* für den Namen des gemeinsam genutzten TSM-Proxy-Knotens.

  In den Datenbankkonfigurationsparametern *logarchmeth2* und *logarchopt2* können ähnliche Änderungen vorgenommen werden.

TSM-Clientkonfiguration mit der Datei dsm.sys

1. Fügen Sie die Proxy-Knoteninformationen in der Datei dsm.sys wie folgt hinzu:

   ```
   asnodename proxy-knoten
   ```

   Dabei steht *proxy-knoten* für den Namen des gemeinsam genutzten TSM-Proxy-Knotens.
2. Stellen Sie wie folgt sicher, dass die angegebene Datei dsm.opt im Pfad **DSMI_CONFIG** den Namen des TSM-Servers enthält:

   ```
   servername servername
   ```

   Dabei ist *servername* der Name des TSM-Servers.

Aspekte der Verwendung von Tivoli Storage Manager

- Bestimmte Funktionen innerhalb von Tivoli Storage Manager (TSM) können Sie möglicherweise nur verwenden, indem Sie den vollständig qualifizierten Pfadnamen des Objekts angeben, das diese Funktion verwendet. (Beachten Sie, dass unter Windows-Betriebssystemen das Zeichen \ statt / eingegeben werden muss.) Für vollständig qualifizierte Pfadnamen gilt Folgendes:
  - Der Pfadname eines Objekts für eine Datenbankgesamtrecovery setzt sich wie folgt zusammen: `/<datenbank>/NODEnnnn/FULL_BACKUP.zeitmarke.folgenr`
  - Der Pfadname eines Objekts für eine inkrementelle Datenbankrecovery setzt sich wie folgt zusammen: `/<datenbank>/NODEnnnn/DB_INCR_BACKUP.zeitmarke.folgenr`
  - Der Pfadname eines Objekts für eine inkrementelle Datenbankdelta-Recovery setzt sich wie folgt zusammen: `/<datenbank>/NODEnnnn/DB_DELTA_BACKUP.zeitmarke.folgenr`
  - Der Pfadname eines Objekts für eine Tabellenbereichsgesamtrecovery setzt sich wie folgt zusammen: `/<datenbank>/NODEnnnn/TSP_BACKUP.zeitmarke.folgenr`
  - Der Pfadname eines Objekts für eine inkrementelle Tabellenbereichs-Recovery setzt sich wie folgt zusammen: `/<datenbank>/NODEnnnn/TSP_INCR_BACKUP.zeitmarke.folgenr`
  - Der Pfadname eines Objekts für eine inkrementelle Tabellenbereichsdelta-Recovery setzt sich wie folgt zusammen: `/<database>/NODEnnnn/TSP_DELTA_BACKUP.zeitmarke.folgenr`

  Dabei steht `<datenbank>` für den Datenbankaliasnamen und `NODEnnnn` für die Knotennummer. Die Namen in Großbuchstaben müssen genau wie dargestellt eingegeben werden.

  Wenn Sie mehrere Backup-Images mit demselben Datenbankaliasnamen haben, dienen die in einem vollständig qualifizierten Namen angegebene Zeitmarke und die Folgenummer als Unterscheidungsmerkmal. Sie müssen TSM abfragen, um die zu verwendende Backup-Version zu ermitteln.
- Wenn Sie eine Online-Backup-Operation ausführen und die Optionen USE TSM und INCLUDE LOGS angeben, kann es zu einem Deadlock kommen, wenn die beiden Prozesse gleichzeitig versuchen, auf dasselbe Bandlaufwerk zu schreiben. Wenn Sie ein Bandlaufwerk als Speichereinheit für Protokolle und Backup-Images verwenden, müssen Sie zwei separate Bandpools für TSM definieren, d. h. einen für das Backup-Image und einen für die archivierten Protokolle.
- Damit Client-Proxy-Knoten verwendet werden können, muss der TSM-Administrator die folgenden Schritte auf dem TSM-Server ausführen:
  1. Wenn die DB2-Clients noch nicht auf dem TSM-Server registriert sind, registrieren Sie jeden Client mit dem TSM-Befehl register node.
  2. Registrieren Sie auf dem TSM-Server einen (virtuellen) allgemeinen TSM-Knotennamen, der von der Clientgruppe verwendet werden solle, mit dem TSM-Befehl register node.
  3. Erteilen Sie allen Computern in der Gruppe die Proxy-Berechtigung mit dem TSM-Befehl grant proxynode.

  Informationen zum Einrichten von Proxy-Knotenclients finden Sie im Abschnitt „Konfigurieren eines Tivoli Storage Manager-Clients" auf Seite 361 oder in der Tivoli-Dokumentation.

Kapitel 15. DB2 Advanced Copy Services (ACS)

DB2 ACS (Advanced Copy Services, erweiterte Kopierservices) ermöglicht Ihnen die Verwendung der leistungsstarken Kopiertechnologie einer Speichereinheit, um im Rahmen von Backup- und Restoreoperationen Daten kopieren zu können.

Bei einer herkömmlichen Backup- oder Restoreoperation kopiert der Datenbankmanager mithilfe von Systemaufrufen Daten auf eine bzw. von einer Platte oder Speichereinheit. Die Möglichkeit, Daten unter Verwendung der Speichereinheit zu kopieren, beschleunigt die Backup- und Restoreoperationen erheblich. Eine Backup-Operation, die DB2 ACS verwendet, wird als Momentaufnahmebackup bezeichnet.

Zur Ausführung von Backup- und Restoreoperationen für Momentaufnahmen benötigen Sie einen DB2 ACS-API-Treiber für Ihre Speichereinheit. In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS Series

Eine Komponente von DB2 Advanced Copy Services (ACS) benötigt Tivoli Storage Manager for Advanced Copy Services. Ausführliche Anweisungen zur Konfiguration und Verwendung von ACS finden Sie in der Tivoli-Dokumentation für Advanced Copy Services unter http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html .

Aktivieren von DB2 ACS

Zur Verwendung von DB2 Advanced Copy Services (ACS) oder zur Ausführung von Momentaufnahmebackup-Operationen muss DB2 ACS installiert, aktiviert und konfiguriert werden.

Vorbereitung

DB2 ACS ist Bestandteil von IBM DB2 High Availability Feature. Zur Verwendung von DB2 ACS benötigen Sie eine Lizenz für DB2 High Availability Feature.

Zur Ausführung von Backup- und Restoreoperationen für Momentaufnahmen benötigen Sie einen DB2 ACS-API-Treiber für Ihre Speichereinheit. In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS Series

Vorgehensweise

1. Installieren Sie DB2 ACS. Siehe hierzu „Installieren von DB2 ACS".
2. Erstellen Sie die Datenbankmanagerinstanz(en), für die Sie DB2 ACS verwenden möchten.

   Bei der Erstellung einer neuen Datenbankmanagerinstanz wird das Verzeichnis `acs` im Verzeichnis `sqllib` der neuen Instanz erstellt. Da jede Datenbankmanagerinstanz über das Verzeichnis `acs` verfügt, können die einzelnen Datenbankmanagerinstanzen unterschiedlich konfiguriert werden.
3. Führen Sie für jede Datenbankmanagerinstanz, mit der Sie DB2 ACS verwenden möchten, die folgenden Schritte aus:
   a. Aktivieren Sie DB2 ACS. Siehe hierzu „Aktivieren von DB2 ACS" auf Seite 367.
   b. Konfigurieren Sie DB2 ACS. Siehe hierzu „Konfigurieren von DB2 Advanced Copy Services (ACS)" auf Seite 368.

Ergebnisse

Nach der Aktivierung von DB2 ACS können Sie Momentaufnahmebackup-Operationen ausführen.

Ausführliche Anweisungen zur Konfiguration und Verwendung von ACS finden Sie in der Tivoli-Dokumentation für Advanced Copy Services unter http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html

Installieren von DB2 ACS

Die erforderlichen Dateien und Bibliotheken für DB2 Advanced Copy Services (ACS) werden vom Installationsprogramm für IBM Data Server installiert.

Einschränkungen

DB2 ACS unterstützt einen Teil der Hardware und Betriebssysteme, die von IBM Data Server unterstützt werden. Eine Liste der von DB2 ACS unterstützten Hardware und Betriebssysteme finden Sie in „DB2 Advanced Copy Services (ACS) - Unterstützte Betriebssysteme und Hardware" auf Seite 415.

Vorbereitung

Vor der Installation von ACS müssen folgende Bibliotheken installiert sein:

Unter AIX:

- ln -s /opt/freeware/lib/powerpc-ibm-aix5.3.0//libgcc_s.a /usr/lib/libgcc_s.a

Unter Red Hat Enterprise Linux:

- ln -s libssl.so.0.9.7xxx libssl.so.0.9.7
- ln -s libcrypto.so.0.9.7xxx libcrypto.so.0.9.7
- ln -s libssl.so.0.9.7xxx libssl.so
- ln -s libssl.so.0.9.7xxx libssl.so.0

Vorgehensweise

1. Installieren Sie IBM Data Server.
2. Fügen Sie in die TCP/IP-Servicedatei einen Port für den DB2 ACS-Agenten ein. Beispiel:

```
db2acs  5400/tcp # DB2 ACS service port
```

Weitere Schritte

Nach dem Installieren von DB2 ACS müssen Sie DB2 ACS aktivieren und DB2 ACS konfigurieren, bevor Sie Momentaufnahmebackup-Operationen ausführen können.

Ausführliche Anweisungen zur Konfiguration und Verwendung von ACS finden Sie in der Tivoli-Dokumentation für Advanced Copy Services unter http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html

Aktivieren von DB2 ACS

Bevor Sie mit DB2 Advanced Copy Services (ACS) ein Momentaufnahmebackup für eine bestimmte Datenbankmanagerinstanz durchführen können, müssen Sie DB2 ACS für diese Instanz aktivieren. DB2 ACS wird durch Ausführung eines Scripts aktiviert.

Vorbereitung

Vor der Aktivierung von DB2 ACS müssen Sie die folgenden Tasks ausführen:

1. Installieren Sie DB2 ACS.
2. Erstellen Sie die Datenbankmanagerinstanz(en), für die Sie DB2 ACS verwenden möchten.

Informationen zu dieser Task

Der Datenbankmanager ruft setup.sh automatisch auf, um DB2 ACS während der Erstellung der Datenbankmanagerinstanz oder bei einem Upgrade von IBM Data Server zu aktivieren.

DB2 ACS können Sie auch manuell aktivieren.

Vorgehensweise

Führen Sie zum manuellen Aktivieren von DB2 ACS das Script `setup.sh` als Benutzer mit Rootberechtigung und mit den für die Aktivierung von DB2 ACS erforderlichen Parametern aus.
Weitere Informationen zu `setup.sh` finden Sie in „DB2 ACS-Konfigurationsscript setup.sh" auf Seite 370.

Ergebnisse

Ein wichtiges Ergebnis bei der Ausführung des Scripts setup.sh ist die Überprüfung des Eigentumsrechts und der Berechtigungen für die ausführbaren Dateien von DB2 ACS im Verzeichnis `sqllib/acs`.

Weitere Schritte

Nach der Aktivierung von DB2 ACS müssen Sie DB2 ACS konfigurieren, bevor Sie Momentaufnahmebackup-Operationen ausführen können.

Ausführliche Anweisungen zur Konfiguration und Verwendung von ACS finden Sie in der Tivoli-Dokumentation für Advanced Copy Services unter http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html

Konfigurieren von DB2 Advanced Copy Services (ACS)

Bevor Sie mit DB2 Advanced Copy Services (ACS) ein Momentaufnahmebackup durchführen können, müssen Sie DB2 ACS konfigurieren. Zum Konfigurieren von DB2 ACS werden Konfigurationsdateien verwendet.

Vorbereitung

Eine Zusammenstellung aller Softwarevoraussetzungen finden Sie in der Dokumentation zu IBM Tivoli Storage Manager im Abschnitt Software Requirements Summary.

Vor dem Konfigurieren von DB2 ACS müssen Sie die folgenden Tasks ausführen:

1. Installieren Sie DB2 ACS.
2. Erstellen Sie die Datenbankmanagerinstanz(en), für die Sie DB2 ACS verwenden möchten.
3. Aktivieren Sie DB2 ACS.

Vorgehensweise

Führen Sie das Script setup.sh im Verzeichnis `sqllib/acs` ohne Angabe von Parametern aus. Daraufhin wird ein interaktiver, textbasierter Assistent aufgerufen, der DB2 ACS konfiguriert. Der Assistent erstellt eine Konfigurationsprofildatei und ändert `/etc/initab` auf der Maschine, um den Start des DB2 ACS-Dämons auszulösen.
Nachfolgend ist ein Beispiel der Ausgabe des Assistenten für setup.sh dargestellt:

```
./setup.sh
Do you have a full TSM license to enable all features of TSM for ACS ?[y/n]

n

****** Profile parameters for section GLOBAL: ******
ACS_DIR [/home/krodger/sqllib/acs ]
ACSD [localhost 57328 ]
TRACE [NO ]

****** Profile parameters for section ACSD: ******
ACS_REPOSITORY *mandatory parameter* /home/krodger/acsrepository

****** Profile parameters for section CLIENT: ******
MAX_VERSIONS [ADAPTIVE ] 2
LVM_FREEZE_THAW [YES ]
DEVICE_CLASS [STANDARD ]

****** Profile parameters for section STANDARD: ******
COPYSERVICES_HARDWARE_TYPE *mandatory parameter*
NAS_NSERIES COPYSERVICES_PRIMARY_SERVERNAME *mandatory parameter* fas960a
COPYSERVICES_USERNAME [superuser ] root

=====================================================================

The profile has beeen successfully created.
```

```
Do you want to continue by specifying passwords for the defined devices? [y/n]

y

Please specify the passwords for the following profile sections:
STANDARD
master

Creating password file at /home/krodger/sqllib/acs/shared/pwd.acsd.
A copy of this file needs to be available to all components that connect to acsd.

BKI1555I: Profile successfully created. Performing additional checks.
Make sure to restart all ACS components to reload the profile.
```

Ergebnisse

Nach dem Konfigurieren von DB2 ACS können Sie Momentaufnahmebackupoperationen ausführen.

Ausführliche Anweisungen zur Konfiguration und Verwendung von ACS finden Sie in der Tivoli-Dokumentation für Advanced Copy Services unter http://publib.boulder.ibm.com/tividd/td/IBMTivoliStorageManagerforAdvancedCopyServices5.3.3.html

Hinweis zur Verwendung:

Wenn beim Konfigurieren von ACS unter AIX Absturzprobleme auftreten, befolgen Sie eine der folgenden Anweisungen, um das Problem zu umgehen:

1. Geben Sie im ACS-Profil `TRACE = YES` ein. Hierdurch wird die ACS-Tracefunktion aktiviert. Gleichzeitig kann diese Einstellung jedoch den Absturz im Assistenten beheben.
2. Wenn der Absturz eintritt, nachdem die Operationen des Assistenten abgeschlossen sind, müssen Sie sicherstellen, dass die ACS-Dämonen installiert wurden. Führen Sie die folgenden Schritte aus, um ACS-Dämonen zu installieren:
   a. `./setup.sh -a query -d $HOME/sqllib`

      Die Ausgabe lautet ähnlich der folgenden: `SUID bit for ../acs/acsnsan is set. Ok.`
   b. Führen Sie den eben angezeigten Agenten mit der Option -f install aus:`./acsnsan -f install`

Die Dämonen sind nun im Verzeichnis `/etc/inittab` installiert und werden momentan ausgeführt.

Nach Durchführung einer dieser Maßnahmen zur Fehlerbehebung werden die übrigen setup.sh-Aktionen normal ausgeführt.

Konfigurieren des DB2 ACS-Verzeichnisses

Bei der Erstellung einer neuen Datenbankmanagerinstanz wird das Verzeichnis `acs` im Verzeichnis `sqllib` der neuen Instanz erstellt. DB2 Advanced Copy Services (ACS) verwendet das Verzeichnis `acs` zum Speichern von Konfigurationsdateien wie beispielsweise der Steuerdatei des Zieldatenträgers oder des gemeinsamen Repositorys für Recoveryobjekte. Die Änderung oder Konfiguration des Verzeichnisses `acs` unterliegt gewissen Einschränkungen.

1. Das Verzeichnis `acs` darf nicht an DB2 ACS- oder Momentaufnahmebackup-Operationen beteiligt sein.

2. Das Verzeichnis acs kann für ein Momentaufnahmebackup unter Verwendung von IBM Tivoli Storage Manager (TSM) auf allen Datenbankpartitionen und auf dem Backup-System in ein NFS exportiert und dort gemeinsam genutzt werden.

DB2 ACS-Konfigurationsscript setup.sh

Das Script setup.sh aktiviert und konfiguriert DB2 ACS (Advanced Copy Services, erweiterte Kopierservices).

Position

Das Script setup.sh befindet sich im Verzeichnis sqllib/acs.

Syntax

Die Syntax für setup.sh lautet wie folgt:

```
Syntax: setup.sh -a aktion
                 -d verzeichnis-der-db2-instanz
                 -u benutzer-id-der-instanz
                 -g primäre-gruppe-der-instanz
```

Dabei kann *aktion* eine der folgenden Aktionen sein:

- start (Starten)
- stop (Stoppen)
- query (Abfragen)
- enable (Aktivieren)
- disable (Inaktivieren)

Verwendung

Der Datenbankmanager ruft setup.sh automatisch auf, um DB2 ACS während der Erstellung der Datenbankmanagerinstanz oder bei einem Upgrade von IBM Data Server zu aktivieren.

Sie können das Script setup.sh auch manuell aufrufen:

Aktivieren von DB2 ACS

Sie können DB2 ACS aktivieren, indem Sie setup.sh als Benutzer mit Rootberechtigung mit den oben angegebenen Parametern ausführen.

Konfigurieren von DB2 ACS

Sie können DB2 ACS konfigurieren, indem Sie setup.sh ohne Parameter ausführen. Wenn Sie setup.sh ohne Parameter ausführen, führt Sie ein Assistent durch die für die Konfiguration von DB2 ACS erforderlichen Schritte.

Ein wichtiges Ergebnis bei der Ausführung des Scripts setup.sh ist die Überprüfung des Eigentumsrechts und der Berechtigungen für die ausführbaren Dateien von DB2 ACS im Verzeichnis sqllib/acs.

DB2 ACS-API

Die DB2 ACS-API (ACS = Advanced Copy Services, erweiterte Kopierservices) definiert eine Reihe von Funktionen, die der Datenbankmanager zur Kommunikation mit der Speicherhardware verwendet, um Momentaufnahmebackup-Operationen auszuführen.

Zur Ausführung von Backup- und Restoreoperationen für Momentaufnahmen benötigen Sie einen DB2 ACS-API-Treiber für Ihre Speichereinheit. In IBM Data Server ist ein DB2 ACS-API-Treiber für die folgende Speicherhardware integriert:

- IBM TotalStorage SAN Volume Controller
- IBM System Storage DS6000
- IBM System Storage DS8000
- IBM System Storage N Series
- NetApp V-series
- NetApp FAS Series

DB2 ACS-API-Funktionen

Der Datenbankmanager übermittelt DB2 ACS-Anforderungen mithilfe der DB2 ACS-API-Funktionen an die Speicherhardware.

db2ACSQueryApiVersion - Aktuelle Version der DB2 ACS-API zurückgeben

Gibt die aktuelle Version der DB2 ACS-API zurück.

API-Include-Datei

```
db2ACSApi.h
```

API- und Datenstruktursyntax

```
db2ACS_Version db2ACSQueryApiVersion();
```

Parameter

Keine

Hinweise zur Verwendung

Mögliche Rückgabewerte:

- `DB2ACS_API_VERSION1`
- `DB2ACS_API_VERSION_UNKNOWN`

db2ACSInitialize - DB2 ACS-Sitzung initialisieren

Initialisiert eine neue DB2 ACS-Sitzung. Dieser Aufruf stellt die Kommunikation zwischen der DB2 ACS-Bibliothek des Datenbankmanagers und dem DB2 ACS-API-Treiber für die Speicherhardware her.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Session Initialization
 * ============================================================================ */
```

```
db2ACS_RC db2ACSInitialize(
              db2ACS_CB              * pControlBlock,
              db2ACS_ReturnCode      * pRC );
```

Parameter

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Vor dem Aufruf von db2ACSInitialize() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->session
pControlBlock->options

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 13. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
`DB2ACS_RC_OK`	Die Operation war erfolgreich.	
`DB2ACS_RC_INIT_FAILED`	Der Datenbankmanager hat versucht, eine DB2 ACS-Sitzung zu initialisieren, aber die Initialisierung ist fehlgeschlagen.	
`DB2ACS_RC_INV_ACTION`	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_INV_DEV_HANDLE`	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Tabelle 13. Rückkehrcodes (Forts.)

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_COMM_ERROR	Es ist ein Kommunikationsfehler mit einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_NO_DEV_AVAIL	Derzeit ist keine Speichereinheit, wie z. B. ein Bandlaufwerk, verfügbar.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Bevor der Datenbankmanager DB2 ACS-API-Aufrufe ausführen kann (Ausnahme: Aufrufe an db2ACSQueryAPIVersion()), muss der Datenbankmanager db2ACSInitialize() aufrufen. Nach der Herstellung einer DB2 ACS-Sitzung durch Aufruf von db2ACSInitialize() kann der Datenbankmanager in DB2 ACS beliebige Kombinationen von Abfrage-, Lese-, Schreib- oder Löschoperationen ausführen. Der Datenbankmanager kann die DB2 ACS-Sitzung durch Aufruf von db2ACSTerminate() beenden.

db2ACSTerminate - DB2 ACS-Sitzung beenden

Beendet eine DB2 ACS-Sitzung.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Session Termination
 * ============================================================================ */
```

```
db2ACS_RC db2ACSTerminate(
              db2ACS_CB              * pControlBlock,
              db2ACS_ReturnCode      * pRC );
```

Parameter

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Der Datenbankmanager ordnet den Speicher für diesen Parameter vor dem Aufruf von db2ACSInitialize() zu. Der Datenbankmanager ist für die Freigabe dieses Speichers nach Ausführung von db2ACSTerminate() verantwortlich.

Vor dem Aufruf von db2ACSTerminate() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->options

Es besteht die Möglichkeit, dass der DB2 ACS-API-Treiber den Speicher in `pControlBlock->vendorInfo.vendorCB` inaktiviert und freigibt.

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 14. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
`DB2ACS_RC_OK`	Die Operation war erfolgreich.	Geben Sie den Speicher frei, der dieser Sitzung zugeordnet ist, und beenden Sie die Operation.
`DB2ACS_INV_ACTION`	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.

- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Der DB2 ACS-API-Treiber muss den gesamten Speicher freigeben, den er für die DB2 ACS-Sitzung in db2ACSTerminate() zugeordnet hat.

Unabhängig davon, ob db2ACSTerminate() fehlerfrei beendet wird, kann der Datenbankmanager in dieser DB2 ACS-Sitzung keine DB2 ACS-Funktionen mehr aufrufen, ohne zuvor db2ACSInitialize() aufzurufen.

db2ACSPrepare - Ausführung einer Momentaufnahmebackup-Operation vorbereiten

Bei der Durchführung eines Momentaufnahmebackups wird die Datenbank vom Datenbankmanager ausgesetzt. db2ACSPrepare() führt alle Schritte zur Vorbereitung einer Momentaufnahmebackup-Operation bis zu dem Punkt aus, an dem der Datenbankmanager die Datenbank aussetzt.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Prepare
 * ============================================================================ */
db2ACS_RC db2ACSPrepare(
               db2ACS_GroupList        * pGroupList,
               db2ACS_CB               * pControlBlock,
               db2ACS_ReturnCode       * pRC );
```

Parameter

pGroupList
Datentyp: `db2ACS_GroupList *`

`db2ACS_GroupList` enthält eine Liste der Gruppen, die in die Momentaufnahmebackup-Operation eingefügt werden sollen.

Wenn **pGroupList** `NULL` ist, werden alle Gruppen (Pfade) in die Momentaufnahmebackup-Operation eingefügt.

Wenn **pGroupList** nicht `NULL` ist:

- **pGroupList** enthält eine Liste der Gruppen (Pfade), die in die Momentaufnahmebackup-Operation eingefügt werden sollen.
- Der Datenbankmanager ist für die Zuordnung und Freigabe des Speichers für **pGroupList** verantwortlich.
- Der Datenbankmanager füllt vor der Übergabe von **pGroupList** an db2ACSPrepare() die folgenden Felder:

  pGroupList->numGroupID
  pGroupList->id

pControlBlock
Datentyp: `db2ACS_CB *`

db2ACS_CB enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSPrepare() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: db2ACS_ReturnCode *

db2ACS_ReturnCode enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters db2ACS_ReturnCode für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 15. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API” auf Seite 413.

Hinweise zur Verwendung

Wenn db2ACSPrepare() erfolgreich ist, setzt der Datenbankmanager die Datenbank aus, bevor db2ACSSnapshot() aufgerufen wird.

db2ACSBeginOperation - DB2 ACS-Operation starten

Startet eine DB2 ACS-Operation.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Operation Begin
 *
 * A valid ACS operation is specified by passing an ObjectType OR'd with one of
 * the following Operations, such as:
 *
 *    (DB2ACS_OP_CREATE | DB2ACS_OBJTYPE_SNAPSHOT)
 * ============================================================================ */
db2ACS_RC db2ACSBeginOperation(
               db2ACS_Operation           operation,
               db2ACS_CB                * pControlBlock,
               db2ACS_ReturnCode        * pRC );
```

Parameter

operation

Datentyp: `db2ACS_Operation`

operation ist eine Bitmaske, die die zu startende DB2 ACS-Operation sowie den beteiligten Objekttyp angibt.

Operationstypen:

DB2ACS_OP_CREATE
DB2ACS_OP_READ
DB2ACS_OP_DELETE

Objekttypen:

DB2ACS_OBJTYPE_BACKUP
DB2ACS_OBJTYPE_LOG
DB2ACS_OBJTYPE_LOADCOPY
DB2ACS_OBJTYPE_SNAPSHOT

Beispiel: `( DB2ACS_OP_CREATE | DB2ACS_OBJTYPE_SNAPSHOT )` oder `( DB2ACS_OP_DELETE | DB2ACS_OBJTYPE_LOADCOPY )`.

Der Datenbankmanager übergibt **operation** an den Funktionsaufruf db2ACSBeginOperation().

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSBeginOperation() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

Wenn **operation** den Wert DB2ACS_OP_CREATE oder DB2ACS_OP_READ aufweist, füllt der Datenbankmanager außerdem das folgende Feld:

pControlBlock->operation

Die in pControlBlock->operation enthaltenen Informationen sind nur im Kontext einer bestimmten DB2 ACS-Operation gültig. pControlBlock->operation wird während der Ausführung von db2ACSBeginOperation() gesetzt und bleibt unverändert, bis die Rückgabe von db2ACSEndOperation() erfolgt. Der Datenbankmanager und der DB2 ACS-API-Treiber sollten außerhalb des Kontexts einer DB2 ACS-Operation nicht auf pControlBlock->operation verweisen.

pRC Datentyp: db2ACS_ReturnCode *

db2ACS_ReturnCode enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters db2ACS_ReturnCode für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 16. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_OPTIONS	Der Datenbankmanager hat ungültige Optionen angegeben.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Keine

db2ACSEndOperation - DB2 ACS-Operation beenden

Beendet eine DB2 ACS-Operation.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Operation End
 * ============================================================================ */
db2ACS_RC db2ACSEndOperation(
               db2ACS_EndAction          endAction,
               db2ACS_CB               * pControlBlock,
               db2ACS_ReturnCode       * pRC );
```

Parameter

endAction

Datentyp: `db2ACS_EndAction`

endAction ist eine Bitmaske, die angibt, wie der DB2 ACS-API-Treiber die DB2 ACS-Operation beenden soll.

Werte:

DB2ACS_END_COMMIT
DB2ACS_END_ABORT

Der Datenbankmanager übergibt **endAction** an den Funktionsaufruf db2ACSEndOperation().

pControlBlock

Datentyp: `db2ACS_CB`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSEndOperation() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 17. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_COMMIT_FAILED	Der DB2 ACS-API-Treiber konnte eine Transaktion nicht festschreiben.	
DB2ACS_RC_ABORT_FAILED	Der Datenbankmanager hat versucht, eine DB2 ACS-Operation abzubrechen, aber der Versuch ist fehlgeschlagen.	

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Wenn der Datenbankmanager DB2ACS_END_ABORT als Parameter **endAction** übergibt, sollten die Momentaufnahmebackup-Objekte gelöscht werden.

db2ACSBeginQuery - Abfrage zu Momentaufnahmebackup-Objekten starten

Startet eine DB2 ACS-Abfrageoperation für Momentaufnahmebackup-Objekte, die für Restoreoperationen verfügbar sind.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
db2ACS_RC db2ACSBeginQuery(
              db2ACS_QueryInput       * pQueryInput,
              db2ACS_CB               * pControlBlock,
              db2ACS_ReturnCode       * pRC );
```

Parameter

pQueryInput
Datentyp: `db2ACS_QueryInput *`

`db2ACS_QueryInput` verfügt über dieselben Felder wie `db2ACS_ObjectInfo`. `db2ACS_ObjectInfo` enthält Informationen zu einem Objekt, das mit der DB2 ACS-API erstellt wurde.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Vor dem Aufruf von db2ACSBeginQuery() füllt der Datenbankmanager die Felder von **pQueryInput**.

Der DB2 ACS-API-Treiber muss die Verwendung der folgenden Platzhalterzeichen in der Abfrage unterstützen:

- `DB2ACS_WILDCARD` in Feldern für Zeichenfolgen
- `DB2ACS_ANY_PARTITIONNUM` in Feldern für Datenbankpartitionen
- `DB2ACS_ANY_UINT32` in Feldern mit 32 Bit großen ganzen Zahl ohne Vorzeichen (Uint32)

pControlBlock
Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSBeginQuery() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 18. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
`DB2ACS_RC_OK`	Die Operation war erfolgreich.	
`DB2ACS_RC_INV_ACTION`	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Tabelle 18. Rückkehrcodes (Forts.)

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

db2ACSBeginQuery() gibt keine Abfragedaten zurück.

db2ACSGetNextObject - Nächstes für Restore verfügbares Momentaufnahmebackup-Objekt auflisten

Gibt das nächste Element in einer Liste mit Momentaufnahmebackup-Objekten zurück, die für Restoreoperationen verwendet werden können.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
db2ACS_RC db2ACSGetNextObject(
              db2ACS_QueryOutput      * pQueryOutput,
              db2ACS_CB               * pControlBlock,
              db2ACS_ReturnCode       * pRC );
```

Parameter

pQueryOutput

Datentyp: db2ACS_QueryOutput *

db2ACS_QueryOutput enthält Abfrageergebnisse für Momentaufnahmebackup-Objekte.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pQueryOutput**.

pControlBlock
Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSGetNextObject() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 19. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
`DB2ACS_RC_OK`	Die Operation war erfolgreich.	
`DB2ACS_RC_INV_ACTION`	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_INV_DEV_HANDLE`	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_DEV_ERROR`	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_IO_ERROR`	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Tabelle 19. Rückkehrcodes (Forts.)

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OBJ_NOT_FOUND	Der DB2 ACS-API-Treiber konnte das vom Datenbankmanager angegebene Momentaufnahmebackup-Objekt nicht finden.	Der Funktionsaufruf ist nicht fehlgeschlagen, aber es sind keine Momentaufnahmebackup-Objekte vorhanden, die mit den an db2ACSBeginQuery() übergebenen Bedingungen übereinstimmen.
DB2ACS_RC_END_OF_DATA	Der DB2 ACS-API-Treiber kann keine weiteren Momentaufnahmebackup-Objekte finden.	Der Funktionsaufruf ist nicht fehlgeschlagen, aber es sind keine weiteren Momentaufnahmebackup-Objekte vorhanden, die mit den an db2ACSBeginQuery() übergebenen Bedingungen übereinstimmen.
DB2ACS_RC_MORE_DATA	Es sind weitere Daten zur Übertragung von der Speicherposition an den Datenbankmanager vorhanden.	Es werden Informationen zu einem Momentaufnahmebackup-Objekt zurückgegeben, die mit den an db2ACSBeginQuery() übergebenen Bedingungen übereinstimmen; darüber hinaus sind weitere Momentaufnahmebackup-Objekte vorhanden, die mit den an db2ACSBeginQuery() übergebenen Bedingungen übereinstimmen.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Der Datenbankmanager muss db2ACSBeginQuery() aufrufen, bevor db2ACSGetNextObject() aufgerufen wird. Der Datenbankmanager gibt die Suchbedingungen im Parameter db2ACS_QueryInput an, der an db2ACSBeginQuery() übergeben wird.

db2ACSGetNextObject() gibt Informationen zu einem Momentaufnahmebackup-Objekt zurück, das mit den an db2ACSBeginQuery() übergebenen Suchbedingungen übereinstimmt. Wenn db2ACSGetNextObject() den Rückkehrcode DB2ACS_RC_MORE_DATA zurückgibt, kann der Datenbankmanager db2ACSGetNextObject() erneut aufrufen, um Informationen zu einem weiteren Momentaufnahmebackup-Objekt zu erhalten, das mit den Suchbedingungen übereinstimmt. Wenn db2ACSGetNextObject() den Rückkehrcode DB2ACS_RC_END_OF_DATA zurückgibt, sind keine weiteren Momentaufnahmebackup-Objekte vorhanden, die mit den Suchbedingungen übereinstimmen.

db2ACSEndQuery - Abfrage zu Momentaufnahmebackup-Objekten beenden

Der Datenbankmanager verwendet die DB2 ACS-API-Funktionen db2ACSBeginQuery() und db2ACSGetNextObject() für Abfragen zu Momentaufnahmebackup-Objekten, die für Restoreoperation verfügbar sind. db2ACSEndQuery() beendet die DB2 ACS-Abfragesitzung.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
db2ACS_RC db2ACSEndQuery(
              db2ACS_CB               * pControlBlock,
              db2ACS_ReturnCode       * pRC );
```

Parameter

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSEndQuery() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 20. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
`DB2ACS_RC_OK`	Die Operation war erfolgreich.	
`DB2ACS_RC_INV_ACTION`	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_INV_DEV_HANDLE`	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Tabelle 20. Rückkehrcodes (Forts.)

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Der Datenbankmanager kann db2ACSGetNextObject() in dieser DB2 ACS-Sitzung erst dann erneut aufrufen, wenn zuvor db2ACSBeginQuery() aufgerufen wurde.

db2ACSSnapshot - DB2 ACS-Operation ausführen

Führt eine DB2 ACS-Operation aus.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
typedef union db2ACS_ReadList
{
   db2ACS_GroupList           group;
} db2ACS_ReadList;


db2ACS_RC db2ACSSnapshot(
               db2ACS_Action             action,
               db2ACS_ObjectID           objectID,
               db2ACS_ReadList         * pReadList,
               db2ACS_CB               * pControlBlock,
               db2ACS_ReturnCode       * pRC );
```

Parameter

action Datentyp: db2ACS_Action

Der Typ der auszuführenden DB2 ACS-Aktion. Werte:

DB2ACS_ACTION_WRITE
DB2ACS_ACTION_READ_BY_OBJECT
DB2ACS_ACTION_READ_BY_GROUP

Der Datenbankmanager übergibt **action** an db2ACSSnapshot().

objectID

Datentyp: `db2ACS_ObjectID`

`db2ACS_ObjectID` gibt eine eindeutige ID für jedes gespeicherte Objekt an, die von einer Abfrage an das Speicherrepository zurückgegeben wird. Der Wert für `db2ACS_ObjectID` ist in jedem Falle eindeutig und nur für den Zeitraum einer DB2 ACS-Sitzung vorhanden.

Wenn der Datenbankmanager im Aufruf an db2ACSBeginOperation() als Operation `DB2ACS_OP_READ` oder `DB2ACS_OP_DELETE` angegeben hat, übergibt er den Wert für die Objekt-ID (**objectID**) an db2ACSSnapshot().

pReadList

Datentyp: `db2ACS_ReadList *`

`db2ACS_ReadList` enthält eine Liste mit Gruppen.

pReadList wird nur dann verwendet, wenn **action** den Wert `DB2ACS_ACTION_READ_BY_GROUP` aufweist.

Wenn **action** den Wert `DB2ACS_ACTION_READ_BY_GROUP` aufweist, ist der Datenbankmanager vor dem Aufruf von db2ACSSnapshot() für die Zuordnung des Speichers und das Füllen der Felder für **pReadLIst** und danach für die Freigabe des Speichers für **pReadList** verantwortlich.

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSSnapshot() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 21. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Der Datenbankmanager ruft db2ACSBeginOperation() auf, bevor db2ACSPartition(), db2ACSPrepare() und db2ACSSnapshot() aufgerufen werden. Der Datenbankmanager gibt den Typ der DB2 ACS-Operation, die der DB2 ACS-API-Treiber ausführen soll, im Parameter **operation** des Aufrufs an db2ACSBeginOperation() an.

db2ACSPartition - Zieldaten für eine Datenbankpartition gruppieren

Ordnet jedem Pfad, den der Datenbankmanager als einer Datenbankpartition zugehörig auflistet, eine Gruppen-ID zu.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Partition
 * ============================================================================ */
db2ACS_RC db2ACSPartition(
               db2ACS_PathList         * pPathList,
               db2ACS_CreateObjectInfo * pCreateObjInfo,
               db2ACS_CB               * PControlBlock,
               db2ACS_ReturnCode       * pRC );
```

Parameter

pPathList

Datentyp: `db2ACS_PathList`

`db2ACS_PathList` enthält eine Liste mit Datenbankpfaden sowie einige zusätzliche Informationen zu den einzelnen Pfaden, die sich auf DB2 ACS-Operationen beziehen.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Das Feld **entry** der Struktur `db2ACS_PathList` ist ein Array von Elementen des Typs `db2ACS_PathEntry`. `db2ACS_PathEntry` enthält Informationen zu einem Datenbankpfad.

Vor dem Aufruf von db2ACSPartition füllt der Datenbankmanager die folgenden Felder für jeden Eintrag `db2ACS_PathEntry` in **pPathList**:

- **path**
- **type**
- **toBeExcluded**

Jeder Pfad, der vom Datenbankmanager als zur Datenbankpartition gehörig erkannt wird, erhält vom DB2 ACS-API-Treiber eine Gruppen-ID. Der DB2 ACS-API-Treiber füllt vor der Rückgabe das Feld **groupID** für jeden Eintrag `db2ACS_PathEntry` in **pPathList** aus.

pCreateObjInfo

Datentyp: `db2ACS_CreateObjectInfo`

`db2ACS_CreateObjectInfo` enthält Informationen zur Erstellung des DB2 ACS-Backup-Objekts.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der Datenbankmanager füllt vor dem Aufruf von db2ACSPartition die Felder von **pCreateObjInfo**.

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSPartition() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 22. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
`DB2ACS_RC_OK`	Die Operation war erfolgreich.	
`DB2ACS_RC_INIT_FAILED`	Der Datenbankmanager hat versucht, eine DB2 ACS-Sitzung zu initialisieren, aber die Initialisierung ist fehlgeschlagen.	
`DB2ACS_RC_INV_ACTION`	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_INV_DEV_HANDLE`	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_DEV_ERROR`	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_IO_ERROR`	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
`DB2ACS_RC_OBJ_OUT_OF_SCOPE`	Der Datenbankmanager hat versucht, eine DB2 ACS-Operation für ein Recoveryobjekt auszuführen, das nicht vom DB2 ACS-API-Treiber verwaltet wird.	

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Die Verarbeitung der Daten auf einer Datenbankpartition durch DB2 Advanced Copy Services erfolgt atomar. Das heißt: Die Daten für eine Datenbankpartition werden unabhängig von anderen Datenbankpartitionen zusammen gesichert oder wiederhergestellt, auch wenn die Aktion Teil einer Operation ist, an der mehrere Datenbankpartitionen beteiligt sind. db2ACSPartition gruppiert die Informationen zu den Datenbankpfaden einer einzelnen Datenbankpartition.

Der Datenbankmanager ruft db2ACSPartition auf, bevor db2ACSSnapshot aufgerufen wird. Der Datenbankmanager listet alle Pfade, die dieser Datenbankpartition zugeordnet sind, im Parameter **pPathList** auf. Der Datenbankmanager kann eine DB2 ACS-Operation für eine Untergruppe der in **pPathList** aufgelisteten Pfade ausführen; dazu muss diese Untergruppe im Parameter **pReadList** angegeben werden, der an db2ACSSnapshot übergeben wird.

db2ACSVerify - Überprüfen, ob eine DB2 ACS-Operation erfolgreich ausgführt wurde

Überprüft, ob eine DB2 ACS-Operation erfolgreich war.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Verify
 * ============================================================================ */
db2ACS_RC db2ACSVerify(
               db2ACS_PostObjectInfo   * pPostObjInfo,
               db2ACS_CB               * pControlBlock,
               db2ACS_ReturnCode       * pRC );
```

Parameter

pPostObjInfo

Datentyp: `db2ACS_PostObjectInfo`

`db2ACS_DB2ID` besteht aus einer Datengruppe, die zum Zeitpunkt der Erstellung eines Momentaufnahmebackup-Objekts nicht bekannt sein kann, aber im Objektrepository verwaltet werden muss.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der Datenbankmanager füllt vor dem Aufruf von db2ACSVerify die Felder von **pCreateObjInfo**. **pPostObjInfo** enthält Informationen, die nach der DB2 ACS-Operation relevant sind. Beispiel: Nach einem erfolgreichen Momentaufnahmebackup enthält **pPostObjInfo** möglicherweise die erste aktive Protokolldatei. Sind nach der DB2 ACS-Operation keine relevanten Daten vorhanden, wird **pPostObjInfo** vom Datenbankmanager auf `NULL` gesetzt.

pControlBlock

Datentyp: `db2ACS_CB *`

db2ACS_CB enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSVerify() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: db2ACS_ReturnCode *

db2ACS_ReturnCode enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters db2ACS_ReturnCode für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 23. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Wenn db2ACSVerify zurückgibt, dass eine Momentaufnahmebackup-Operation erfolgreich war, sind die vom Momentaufnahmebackup generierten Recoveryobjekte für Restoreoperationen verfügbar.

db2ACSDelete - Recoveryobjekte löschen, die mit DB2 ACS erstellt wurden

Löscht Recoveryobjekte, die mithilfe von DB2 Advanced Copy Services (ACS) erstellt wurden.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
/* ============================================================================
 * Delete
 * ============================================================================ */
db2ACS_RC db2ACSDelete(
               db2ACS_ObjectID          objectID,
               db2ACS_CB              * pControlBlock,
               db2ACS_ReturnCode      * pRC );
```

Parameter

objectID

Datentyp: `db2ACS_ObjectID`

`db2ACS_ObjectID` gibt eine eindeutige ID für jedes gespeicherte Objekt an, die von einer Abfrage an das Speicherrepository zurückgegeben wird. Der Wert für `db2ACS_ObjectID` ist in jedem Falle eindeutig und nur für den Zeitraum einer DB2 ACS-Sitzung vorhanden.

Der Datenbankmanager kann mit db2ACSQuery() eine gültige Objekt-ID (**objectID**) abrufen und an db2ACSDelete() übergeben.

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSDelete() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 24. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	Das angegebene Objekt wird gelöscht. Es können keine weiteren DB2 ACS-Operationen für dieses Objekt ausgeführt werden.
DB2ACS_RC_DELETE_FAILED	Der DB2 ACS-API-Treiber konnte die vom Datenbankmanager angegebenen Momentaufnahmebackup-Objekte nicht löschen.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_OBJ_NOT_FOUND	Der DB2 ACS-API-Treiber konnte das vom Datenbankmanager angegebene Momentaufnahmebackup-Objekt nicht finden.	

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Wenn der Datenbankmanager db2ACSDelete aufruft, löscht der DB2 ACS-API-Treiber das über die Objekt-ID identifizierte Recoveryobjekt.

Der Datenbankmanager ruft db2ACSDelete auf, wenn ein Benutzer db2acsutil mit dem Parameter DELETE aufruft.

db2ACSStoreMetaData - Metadaten für ein mit DB2 ACS generiertes Recoveryobjekt speichern

Speichert Metadaten zu einem Recoveryobjekt, das mit DB2 Advanced Copy Services (ACS) erstellt wurde.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
db2ACS_RC db2ACSStoreMetaData(
              db2ACS_MetaData         * pMetaData,
              db2ACS_CB               * pControlBlock,
              db2ACS_ReturnCode       * pRC );
```

Parameter

pMetaData

Datentyp: `db2ACS_MetaData`

`db2ACS_MetaData` speichert Metadaten zu Momentaufnahmebackups.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Die im Feld **data** von **pMetaData** gespeicherten Metadaten sind interne Daten des Datenbankmanagers, die sich im Laufe der Zeit ändern können; deshalb behandelt der DB2 ACS-API-Treiber diese Daten als binären Datenstrom.

pControlBlock

Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSStoreMetaData() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 25. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Eine Momentaufnahmebackup-Operation besteht aus mehreren DB2 ACS-API-Funktionsaufrufen wie db2ACSInitialize, db2ACSBeginOperation, db2ACSPrepare und db2ACSSnapshot. db2ACSStoreMetaData ist ebenfalls Teil der Gesamtoperation. Alle API-Aufrufe (auch db2ACSStoreMetaData) müssen erfolgreich sein, damit die Momentaufnahmebackup-Operation erfolgreich ausgeführt werden kann. Wenn db2ACSStoreMetaData fehlschlägt, kann das von der DB2 ACS-Backup-Operation generierte Recoveryobjekt nicht verwendet werden.

db2ACSRetrieveMetaData - Metadaten zu einem mit DB2 ACS generierten Recoveryobjekt abrufen

Ruft Metadaten zu einem Recoveryobjekt ab, das mit DB2 Advanced Copy Services (ACS) erstellt wurde.

Include-Datei

```
db2ACSApi.h
```

Syntax und Datenstrukturen

```
db2ACS_RC db2ACSRetrieveMetaData(
              db2ACS_MetaData         * pMetaData,
              db2ACS_ObjectID           objectID,
              db2ACS_CB               * pControlBlock,
              db2ACS_ReturnCode       * pRC );
```

Parameter

pMetaData
Datentyp: `db2ACS_MetaData`

`db2ACS_MetaData` speichert Metadaten zu Momentaufnahmebackups.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Die im Feld **data** von **pMetaData** gespeicherten Metadaten sind interne Daten des Datenbankmanagers, die sich im Laufe der Zeit ändern können; deshalb behandelt der DB2 ACS-API-Treiber diese Daten als binären Datenstrom.

objectID
Datentyp: `db2ACS_ObjectID`

`db2ACS_ObjectID` gibt eine eindeutige ID für jedes gespeicherte Objekt an, die von einer Abfrage an das Speicherrepository zurückgegeben wird. Der Wert für `db2ACS_ObjectID` ist in jedem Falle eindeutig und nur für den Zeitraum einer DB2 ACS-Sitzung vorhanden.

Der Datenbankmanager kann mit db2ACSQuery() eine gültige Objekt-ID (**objectID**) abrufen und an db2ACSRetrieveMetaData() übergeben.

pControlBlock
Datentyp: `db2ACS_CB *`

`db2ACS_CB` enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

Vor dem Aufruf von db2ACSRetrieveMetaData() füllt der Datenbankmanager die folgenden Felder:

pControlBlock->handle
pControlBlock->vendorInfo
pControlBlock->options

pRC Datentyp: `db2ACS_ReturnCode *`

`db2ACS_ReturnCode` enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters `db2ACS_ReturnCode` für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

Der Datenbankmanager ordnet den Speicher für diesen Parameter zu und übergibt einen Verweis auf dieses instanziierte Objekt an die Funktion. Der Datenbankmanager ist für die Freigabe des Speichers verantwortlich.

Der DB2 ACS-API-Treiber füllt vor der Rückgabe die Felder von **pRC**.

Rückkehrcodes

Tabelle 26. Rückkehrcodes

Rückkehrcode	Beschreibung	Anmerkungen
DB2ACS_RC_OK	Die Operation war erfolgreich.	
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.
DB2ACS_RC_OBJ_NOT_FOUND	Der DB2 ACS-API-Treiber konnte das vom Datenbankmanager angegebene Momentaufnahmebackup-Objekt nicht finden.	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt. Der Datenbankmanager kann die DB2 ACS-API-Sitzung nicht verwenden.

Wenn der DB2 ACS-API-Treiber einen Fehler feststellt, bricht der Treiber möglicherweise eine DB2 ACS-Operation ab. Die DB2 ACS-Sitzung kann nur für die folgenden Aktionen verwendet werden:

- Wenn ein Aufruf an db2ACSBeginQuery() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndQuery() aufrufen.
- Wenn ein Aufruf an db2ACSBeginOperation() zuvor erfolgreich war, kann der Datenbankmanager db2ACSEndOperation() aufrufen.
- Wenn ein Aufruf an db2ACSInitialize() zuvor erfolgreich war, kann der Datenbankmanager db2ACSTerminate() aufrufen.

Weitere Informationen zu DB2 ACS-API-Rückkehrcodes finden Sie in „Rückkehrcodes der DB2 ACS-API" auf Seite 413.

Hinweise zur Verwendung

Keine

Datenstrukturen der DB2 ACS-API

Wenn Sie die Funktionen der DB2 ACS-API aufrufen möchten, müssen Sie die Datenstrukturen der DB2 ACS-API verwenden.

db2ACS_BackupDetails - Datenstruktur der DB2 ACS-API

db2ACS_BackupDetails enthält Informationen zu einer Momentaufnahmebackup-Operation.

```
/* ------------------------------------------------------------------------ */
typedef struct db2ACS_BackupDetails
{
   /* A traditional DB2 backup can consist of multiple objects (logical tapes),
```

```
    * where each object is uniquely numbered with a non-zero natural number.
    * ------------------------------------------------------------------------ */
   db2Uint32                  sequenceNum;

   char                       imageTimestamp[SQLU_TIME_STAMP_LEN + 1];
} db2ACS_BackupDetails;
```

sequenceNum
Datentyp: db2Uint32

Identifiziert ein Backup-Objekt durch die zugehörige eindeutige Zahl.

imageTimestamp
Datentyp: char[]

Eine Zeichenfolge mit der Länge SQLU_TIME_STAMP_LEN + 1.

db2ACS_CB - Datenstruktur der DB2 ACS-API

db2ACS_CB enthält grundlegende Informationen, die zum Initialisieren und Beenden einer DB2 ACS-Sitzung erforderlich sind.

```
/* ============================================================================
 * DB2 Backup Adapter Control Block
 * ============================================================================ */
typedef struct db2ACS_CB
{
   /* Output:  Handle value for this session.
    * ------------------------------------------------------------------------ */
   db2Uint32                  handle;
   db2ACS_VendorInfo          vendorInfo;

   /* Input fields and parameters.
    * ------------------------------------------------------------------------ */
   db2ACS_SessionInfo         session;
   db2ACS_Options             options;

   /* Operation info is optional, possibly NULL, and is only ever valid
    * within the context of an operation (from call to BeginOperation() until
    * the EndOperation() call returns).
    *
    * The operation info will be present during creation or read operations
    * of snapshot and backup objects.
    * ------------------------------------------------------------------------ */
   db2ACS_OperationInfo     * operation;
} db2ACS_CB;
```

handle
Datentyp: db2Uint32

Eine Kennung, die auf die DB2 ACS-Sitzung verweist.

vendorInfo
Datentyp: db2ACS_VendorInfo

db2ACS_VendorInfo enthält Informationen zum DB2 ACS-API-Treiber.

session
Datentyp: db2ACS_SessionInfo

db2ACS_SessionInfo enthält alle Informationen zur DB2 ACS-Sitzung.

options
Datentyp: db2ACS_Options

db2ACS_Options gibt Optionen an, die für eine DB2 ACS-Operation verwendet werden. Der Inhalt dieser Zeichenfolge bezieht sich auf den DB2 ACS-API-Treiber.

operation
Datentyp: db2ACS_OperationInfo *

db2ACS_OperationInfo enthält Informationen zu einer Momentaufnahmebackup-Operation.

db2ACS_CreateObjectInfo - Datenstruktur der DB2 ACS-API

db2ACS_CreateObjectInfo enthält Informationen zur Erstellung des DB2 ACS-Backup-Objekts.

```
/* ============================================================================
 * Object Creation Parameters.
 * ============================================================================ */
typedef struct db2ACS_CreateObjectInfo
{
   db2ACS_ObjectInfo          object;
   db2ACS_DB2ID               db2ID;

   /* ------------------------------------------------------------------------
    * The following fields are optional information for the database manager
    * to use as it sees fit.
    * ------------------------------------------------------------------------ */

   /* Historically both the size estimate and management
    * class parameters have been used by the TSM client API for traditional
    * backup objects, log archives, and load copies, but not for snapshot
    * backups.
    * ------------------------------------------------------------------------ */
     db2Uint64 sizeEstimate;
   char                       mgmtClass[DB2ACS_MAX_MGMTCLASS_SZ + 1];

   /* The appOptions is a copy of the iOptions field of flags passed to DB2's
    * db2Backup() API when this execution was initiated.  This field will
    * only contain valid data when creating a backup or snapshot object.
    * ------------------------------------------------------------------------ */
   db2Uint32                  appOptions;
} db2ACS_CreateObjectInfo;
```

Objekt
Datentyp: db2ACS_ObjectInfo

db2ACS_ObjectInfo enthält Informationen zu einem Objekt, das mit der DB2 ACS-API erstellt wurde.

db2ID Datentyp: db2ACS_DB2ID

db2ACS_DB2ID identifiziert IBM Data Server.

sizeEstimate
Datentyp: db2Uint64

Eine Schätzung der Größe der Backup-Objekte, die erstellt werden. Diese Schätzung gilt nicht für Protokollarchive, Ladekopien oder Momentaufnahmebackup-Objekte.

mgmtClass
Datentyp: db2ACS_MgmtClass

Eine Zeichenfolge mit der Länge db2ACS_MAX_MGMTCLASS_SZ + 1.

Gilt nicht für Momentaufnahmebackup-Objekte.

appOptions
Datentyp: db2Uint32.

Eine Kopie der Backup-Optionen wird an den Befehl BACKUP übergeben, der das Momentaufnahmebackup initialisiert hat.

db2ACS_DB2ID - Datenstruktur der DB2 ACS-API

db2ACS_DB2ID identifiziert IBM Data Server.

```
/* ============================================================================
 * DB2 Data Server Identifier
 * ============================================================================ */
typedef struct db2ACS_DB2ID
{
   db2Uint32                  version;
   db2Uint32                  release;
   db2Uint32                  level;
   char                       signature[DB2ACS_SIGNATURE_SZ + 1];
} db2ACS_DB2ID;
```

version

Datentyp: db2Uint32

Version von IBM Data Server. Beispiel: 9

release

Datentyp: db2Uint32

Release-Level von IBM Data Server. Beispiel: 5

level Datentyp: db2Uint32

Aktualitäts-ID für IBM Data Server. Beispiel: 0

signature

Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_SIGNATURE_SZ + 1. Beispiel: SQL09050

db2ACS_GroupList - Datenstruktur der DB2 ACS-API

db2ACS_GroupList enthält eine Liste der Gruppen, die in die Momentaufnahme-backup-Operation eingefügt werden sollen.

```
/* ============================================================================
 * Snapshot Group List
 *
 * This is an array of size 'numGroupIDs', indicating the set of groups that
 * are to be included in the snapshot operation.
 * ============================================================================ */
typedef struct db2ACS_GroupList
{
   db2Uint32                  numGroupIDs;
   db2Uint32                * id;
} db2ACS_GroupList;
```

numGroupIDs

Datentyp: db2Uint32

Die Anzahl der Gruppen im Array **id**.

id Datentyp: db2Uint32 *

Ein Array von Gruppen-IDs. Die identifizierten Gruppen sind die Gruppen (oder Pfadlisten), die in die Momentaufnahmebackup-Operation eingefügt werden sollen.

db2ACS_LoadcopyDetails - Datenstruktur der DB2 ACS-API

db2ACS_LoadcopyDetails enthält Informationen zu einer Ladekopie-Operation.

```
/* ---------------------------------------------------------------------------- */
typedef struct db2ACS_LoadcopyDetails
{
   /* Just like the BackupDetails, a DB2 load copy can consist of multiple
```

```
    * objects (logical tapes), where each object is uniquely numbered with a
    * non-zero natural number.
    * ------------------------------------------------------------------------ */
   db2Uint32                  sequenceNum;

   char                       imageTimestamp[SQLU_TIME_STAMP_LEN + 1];
} db2ACS_LoadcopyDetails;
```

sequenceNum
Datentyp: db2Uint32

Identifiziert ein Backup-Objekt durch die zugehörige eindeutige Zahl.

imageTimestamp
Datentyp: char[]

Eine Zeichenfolge mit der Länge SQLU_TIME_STAMP_LEN + 1.

db2ACS_LogDetails - Datenstruktur der DB2 ACS-API

db2ACS_LogDetails enthält Informationen, die eine bestimmte Datenbankprotokolldatei identifizieren.

```
/* ------------------------------------------------------------------------ */
typedef struct db2ACS_LogDetails
{
   db2Uint32                  fileID;
   db2Uint32                  chainID;
} db2ACS_LogDetails;
```

fileID Datentyp: db2Uint32

Eine Zahl, die den Dateinamen der Datenbankprotokolldatei darstellt.

chainID
Datentyp: db2Uint32

Eine Zahl, die die Datenbankprotokolldateikette identifiziert, zu der die Datenbankprotokolldatei **fileID** gehört.

db2ACS_ObjectInfo - Datenstruktur der DB2 ACS-API

db2ACS_ObjectInfo enthält Informationen zu einem Objekt, das mit der DB2 ACS-API erstellt wurde.

```
/* ========================================================================
 * Object Description and Associated Information.
 *
 * This structure is used for both input and output, and its contents define
 * the minimum information that must be recorded about any object created
 * through this interface.
 * ======================================================================== */
typedef struct db2ACS_ObjectInfo
{
   db2ACS_ObjectType          type;
      SQL_PDB_NODE_TYPE dbPartitionNum;

   char                       db[SQL_DBNAME_SZ + 1];
   char                       instance[DB2ACS_MAX_OWNER_SZ + 1];
   char                       host[SQL_HOSTNAME_SZ + 1];
   char                       owner[DB2ACS_MAX_OWNER_SZ + 1];

   Union-Verknüpfung   {
      db2ACS_BackupDetails    backup;
      db2ACS_LogDetails       log;
      db2ACS_LoadcopyDetails  loadcopy;
      db2ACS_SnapshotDetails  snapshot;
   } details;
} db2ACS_ObjectInfo;
```

type Datentyp: db2ACS_ObjectType

Gibt den Typ der Momentaufnahmebackup-Objekte an. Werte:

DB2ACS_OBJTYPE_ALL
DB2ACS_OBJTYPE_BACKUP
DB2ACS_OBJTYPE_LOG
DB2ACS_OBJTYPE_LOADCOPY
DB2ACS_OBJTYPE_SNAPSHOT

DB2ACS_OBJTYPE_ALL kann nur als Filter für Abfragen verwendet werden. Es gibt keine Objekte des Typs 0.

dbPartitionNum
Datentyp: SQL_PDB_NODE_TYPE

Eine Kennung für diese Datenbankpartition.

db Datentyp: char[]

Eine Zeichenfolge mit der Länge SQL_DBNAME_SZ + 1.

instance
Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_OWNER_SZ + 1.

host Datentyp: char[]

Eine Zeichenfolge mit der Länge SQL_HOSTNAME_SZ + 1.

owner Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_OWNER_SZ + 1.

details

backup
Datentyp: db2ACS_BackupDetails

db2ACS_BackupDetails enthält Informationen zu einer Momentaufnahmebackup-Operation.

log Datentyp: db2ACS_LogDetails

db2ACS_LogDetails enthält Informationen, die eine bestimmte Datenbankprotokolldatei identifizieren.

loadcopy
Datentyp: db2ACS_LoadcopyDetails

db2ACS_LoadcopyDetails enthält Informationen zu einer Ladekopie-Operation.

snapshot
Datentyp: db2ACS_SnapshotDetails

db2ACS_SnapshotDetails enthält Informationen zu einer Momentaufnahmebackup-Operation.

db2ACS_ObjectStatus - Datenstruktur der DB2 ACS-API

db2ACS_ObjectStatus enthält Informationen zum Status oder Fortschritt einer Momentaufnahmebackup-Operation bzw. zum Status oder zur Benutzerfreundlichkeit eines Momentaufnahmebackup-Objekts.

```
typedef struct db2ACS_ObjectStatus
{
   /* The total and completed bytes refer only to the ACS snapshot backup
    * itself, not to the progress of any offloaded tape backup.
    *
    * A bytesTotal of 0 indicates that the progress could not be determined.
    * ------------------------------------------------------------------------ */
   db2Uint64                bytesCompleted;
   db2Uint64                bytesTotal;
   db2ACS_ProgressState     progressState;
   db2ACS_UsabilityState    usabilityState;
} db2ACS_ObjectStatus;
```

bytesCompleted
Datentyp: db2Uint64

Der Teil des Momentaufnahmebackups, der bereits abgeschlossen ist (in Byte).

bytesTotal
Datentyp: db2Uint64

Die Größe des abgeschlossenen Momentaufnahmebackups (in Byte).

progressState
Datentyp: db2ACS_ProgressState

Der Status der Momentaufnahmebackup-Operation. Werte:

DB2ACS_PSTATE_UNKNOWN
DB2ACS_PSTATE_IN_PROGRESS
DB2ACS_PSTATE_SUCCESSFUL
DB2ACS_PSTATE_FAILED

usabilityState
Datentyp: db2ACS_UsabilityState

Der Status des Momentaufnahmebackup-Objekts bzw. wie das Momentaufnahmebackup-Objekt verwendet werden kann. Werte:

DB2ACS_USTATE_UNKNOWN
DB2ACS_USTATE_LOCALLY_MOUNTABLE
DB2ACS_USTATE_REMOTELY_MOUNTABLE
DB2ACS_USTATE_REPETITIVELY_RESTORABLE
DB2ACS_USTATE_DESTRUCTIVELY_RESTORABLE
DB2ACS_USTATE_SWAP_RESTORABLE
DB2ACS_USTATE_PHYSICAL_PROTECTION
DB2ACS_USTATE_FULL_COPY
DB2ACS_USTATE_DELETED
DB2ACS_USTATE_FORCED_MOUNT
DB2ACS_USTATE_BACKGROUND_MONITOR_PENDING
DB2ACS_USTATE_TAPE_BACKUP_PENDING
DB2ACS_USTATE_TAPE_BACKUP_IN_PROGRESS
DB2ACS_USTATE_TAPE_BACKUP_COMPLETE

db2ACS_OperationInfo - Datenstruktur der DB2 ACS-API

db2ACS_OperationInfo enthält Informationen zu einer Momentaufnahmebackup-Operation.

```
/* ============================================================================
 * Operation Info
 *
 * The information contained within this structure is only valid within the
```

```
 * context of a particular operation.  It will be valid at the time
 * BeginOperation() is called, and will remain unchanged until EndOperation()
 * returns, but must not be referenced outside the scope of an operation.
 * ============================================================================ */
typedef struct db2ACS_OperationInfo
{
   db2ACS_SyncMode            syncMode;

   /* List of database and backup operation partitions.
    *
    * For details, refer to the db2ACS_PartitionList definition.
    * ------------------------------------------------------------------------- */
   db2ACS_PartitionList     * dbPartitionList;
} db2ACS_OperationInfo;
```

syncMode

Datentyp: `db2ACS_SyncMode`

Die Synchronisationsebene zwischen den Backup-Operationen auf unterschiedlichen Datenbankpartitionen.

Werte:

DB2ACS_SYNC_NONE
Für zusammengehörige Operationen auf mehreren Datenbankpartitionen findet keine Synchronisation statt. Wird bei Operationen verwendet, die auf mehreren Datenbankpartitionen ausgeführt werden und keine Synchronisation verwenden.

DB2ACS_SYNC_SERIAL
Wird bei der Ausführung gleichzeitiger Momentaufnahmebackup-Operationen auf mehreren Datenbankpartitionen verwendet. Auf jeder Datenbankpartition wird die Ein-/Ausgabe (E/A) bei Ausgabe der Momentaufnahmebackup-Operation ausgesetzt; anschließend wird die E/A auf diesen Datenbankpartitionen seriell und nicht gleichzeitig fortgesetzt.

SYNC_PARALLEL
Eine Momentaufnahmenoperation wird auf mehreren Partitionen gleichzeitig ausgeführt. Nachdem alle an der Momentaufnahmebackup-Operation beteiligten Datenbankpartitionen die Vorbereitungen für die Momentaufnahmebackup-Operation abgeschlossen haben, wird die E/A auf allen Datenbankpartitionen ausgesetzt. Die verbleibenden Schritte für das Momentaufnahmebackup werden auf allen beteiligten Datenbankpartitionen gleichzeitig ausgeführt.

dbPartitionList

Datentyp: `db2ACS_PartitionList *`

`db2ACS_PartitionList` enthält Informationen zu den Datenbankpartitionen in der Datenbank, die an einer DB2 ACS-Operation beteiligt sind.

db2ACS_Options - Datenstruktur der DB2 ACS-API

`db2ACS_Options` gibt Optionen an, die für eine DB2 ACS-Operation verwendet werden. Der Inhalt dieser Zeichenfolge bezieht sich auf den DB2 ACS-API-Treiber.

```
/* ============================================================================
 * DB2 Backup Adapter User Options
 * ============================================================================ */
typedef struct db2ACS_Options
```

```
{
   db2Uint32                 size;
   void                    * data;
} db2ACS_Options;
```

size Datentyp: db2Uint32

Die Größe des Parameters **data** (in Byte).

data Datentyp: void *

Ein Verweis auf einen Speicherblock, der die Optionen enthält.

db2ACS_PartitionEntry - Datenstruktur der DB2 ACS-API

db2ACS_PartitionEntry ist ein Element von db2ACS_PartitionList.

```
typedef struct db2ACS_PartitionEntry
{
   SQL_PDB_NODE_TYPE          num;
   char                       host[SQL_HOSTNAME_SZ + 1];
} db2ACS_PartitionEntry;
```

num Datentyp: SQL_PDB_NODE_TYPE

Eine Kennung für diesen Datenbankpartitionseintrag.

host Datentyp: char[]

Eine Zeichenfolge mit der Länge SQL_HOSTNAME_SZ + 1.

db2ACS_PartitionList - Datenstruktur der DB2 ACS-API

db2ACS_PartitionList enthält Informationen zu den Datenbankpartitionen in der Datenbank, die an einer DB2 ACS-Operation beteiligt sind.

```
typedef struct db2ACS_PartitionList
{
   db2Uint64                  numPartsInDB;
   db2Uint64                  numPartsInOperation;
   db2ACS_PartitionEntry    * partition;
} db2ACS_PartitionList;
```

numPartsInDB

Datentyp: db2Uint64

Die Anzahl der Datenbankpartitionen in der Datenbank.

numPartsInOperation

Datentyp: db2Uint64

Die Anzahl der Datenbankpartitionen, die an der DB2 ACS-Operation beteiligt sind.

partition

Datentyp: db2ACS_PartitionEntry *

db2ACS_PartitionEntry ist ein Element von db2ACS_PartitionList.

db2ACS_PathEntry - Datenstruktur der DB2 ACS-API

db2ACS_PathEntry enthält Informationen zu einem Datenbankpfad.

```
typedef struct db2ACS_PathEntry
{
   /* INPUT: The path and type will be provided by the database server, as well
    *        as a flag indicating if the path is to be excluded from the backup.
    * ------------------------------------------------------------------------ */
   char                       path[DB2ACS_MAX_PATH_SZ + 1];
   db2ACS_PathType            type;
   db2Uint32                  toBeExcluded;
```

```
   /* OUTPUT: The group ID is to be provided by the backup adapter for use by
    *         the DB2 server.  The group ID will be used during with snapshot
    *         operations as an indication of which paths are dependent and must
    *         be included together in any snapshot operation.  Unique group IDs
    *         indicate that the paths in those groups are independent for the
    *         purposes of snapshot operations.
    * ------------------------------------------------------------------------ */
   db2Uint32                 groupID;
} db2ACS_PathEntry;
```

path Datentyp: `char[]`

Eine Zeichenfolge mit der Länge `DB2ACS_MAX_PATH_SZ + 1`.

type Datentyp: `db2ACS_PathType`

Der Pfadtyp. Werte:

DB2ACS_PATH_TYPE_UNKNOWN
DB2ACS_PATH_TYPE_LOCAL_DB_DIRECTORY
DB2ACS_PATH_TYPE_DBPATH
DB2ACS_PATH_TYPE_DB_STORAGE_PATH
DB2ACS_PATH_TYPE_TBSP_CONTAINER
DB2ACS_PATH_TYPE_TBSP_DIRECTORY
DB2ACS_PATH_TYPE_TBSP_DEVICE
DB2ACS_PATH_TYPE_LOGPATH
DB2ACS_PATH_TYPE_MIRRORLOGPATH

toBeExcluded
Datentyp: `db2Uint32`

Eine Markierung, die angibt, ob der angegebene Pfad in das Momentaufnahmebackup eingefügt werden soll. Werte:

- 0 - Pfad in Momentaufnahmebackup einfügen
- 1 - Pfad nicht in Momentaufnahmebackup einfügen

groupID
Datentyp: `db2Uint32`

Eine Gruppen-ID.

db2ACS_PathList - Datenstruktur der DB2 ACS-API

`db2ACS_PathList` enthält eine Liste mit Datenbankpfaden sowie einige zusätzliche Informationen zu den einzelnen Pfaden, die sich auf DB2 ACS-Operationen beziehen.

```
/* ============================================================================
 * Snapshot File List
 *
 * This is an array of 'numEntries' db2ACS_PathEntry's, where each path entry is
 * a path to some storage on the DB2 server which is in use by the current
 * database.
 * ============================================================================ */

typedef struct db2ACS_PathList
{
   db2Uint32                 numEntries;
   db2ACS_PathEntry        * entry;
} db2ACS_PathList;
```

numEntries
Datentyp: `db2Uint32`

Die Anzahl der Pfadeinträge im Array **entry**.

entry Datentyp: db2ACS_PathEntry

db2ACS_PathEntry enthält Informationen zu einem Datenbankpfad.

db2ACS_PostObjectInfo - Datenstruktur der DB2 ACS-API

db2ACS_DB2ID besteht aus einer Datengruppe, die zum Zeitpunkt der Erstellung eines Momentaufnahmebackup-Objekts nicht bekannt sein kann, aber im Objektrepository verwaltet werden muss.

```
/* ============================================================================
 * The PostObjectInfo is a set of data that can not be known at object
 * creation time, but which must be maintained in the object repository.  This
 * is an optional field on the Verify() call, which may be NULL if there are
 * no post-operation updates to be made.
 * ========================================================================== */
typedef struct db2ACS_PostObjectInfo
{
   /* The first active log will only be valid when creating a backup or
    * snapshot object.  It will indicate the file number and chain id of the
    * first log required for recovery using this object.
    * ------------------------------------------------------------------------ */
   db2ACS_LogDetails          firstActiveLog;
} db2ACS_PostObjectInfo;
```

firstActiveLog

Datentyp: db2ACS_LogDetails

db2ACS_LogDetails enthält Informationen, die eine bestimmte Datenbankprotokolldatei identifizieren.

db2ACS_QueryInput und db2ACS_QueryOutput - Datenstrukturen der DB2 ACS-API

db2ACS_QueryInput enthält Informationen, die ein Objekt identifizieren, das von Ihnen abgefragt wird. db2ACS_QueryOutput enthält Abfrageergebnisse für Momentaufnahmebackup-Objekte.

```
/* ============================================================================
 * Unique Querying.
 *
 * When using this structure as query input, to indicate the
 * intention to supply a 'wildcard' search criteria, DB2 will supply:
 *
 *    -- character strings as "*".
 *    -- numeric values as (-1), cast as the appropriate signed or unsigned
 *       type.
 * ========================================================================== */

typedef struct db2ACS_ObjectInfo db2ACS_QueryInput;

typedef struct db2ACS_QueryOutput
{
   db2ACS_ObjectID            objectID;
   db2ACS_ObjectInfo          object;
   db2ACS_PostObjectInfo      postInfo;
   db2ACS_DB2ID               db2ID;
   db2ACS_ObjectStatus        status;

   /* Size of the object in bytes.
    * ------------------------------------------------------------------------  */
   db2Uint64                  objectSize;

   /* Size of the metadata associated with the object, if any, in bytes.
    * ------------------------------------------------------------------------  */
   db2Uint64                  metaDataSize;

   /* The creation time of the object is a 64bit value with a definition
```

```
    * equivalent to an ANSI C time_t value (seconds since the epoch, GMT).
    *
    * This field is equivalent to the file creation or modification time in
    * a traditional filesystem.  This should be created and stored
    * automatically by the BA subsystem, and a valid time value should be
    * returned with object query results, for all object types.
    * ------------------------------------------------------------------  */
   db2Uint64                createTime;
} db2ACS_QueryOutput;
```

objectID
Datentyp: db2ACS_ObjectID

db2ACS_ObjectID gibt eine eindeutige ID für jedes gespeicherte Objekt an, die von einer Abfrage an das Speicherrepository zurückgegeben wird. Der Wert für db2ACS_ObjectID ist in jedem Falle eindeutig und nur für den Zeitraum einer DB2 ACS-Sitzung vorhanden.

object Datentyp: db2ACS_ObjectInfo

db2ACS_ObjectInfo enthält Informationen zu einem Objekt, das mit der DB2 ACS-API erstellt wurde.

postInfo
Datentyp: db2ACS_PostObjectInfo

db2ACS_DB2ID besteht aus einer Datengruppe, die zum Zeitpunkt der Erstellung eines Momentaufnahmebackup-Objekts nicht bekannt sein kann, aber im Objektrepository verwaltet werden muss.

db2ID Datentyp: db2ACS_DB2ID

db2ACS_DB2ID identifiziert IBM Data Server.

status Datentyp: db2ACS_ObjectStatus

db2ACS_ObjectStatus enthält Informationen zum Status oder Fortschritt einer Momentaufnahmebackup-Operation bzw. zum Status oder zur Benutzerfreundlichkeit eines Momentaufnahmebackup-Objekts.

objectSize
Datentyp: db2Uint64

Die Größe des Objekts (in Byte).

metaDataSize
Datentyp: db2Uint64

Die Größe der Metadaten, die gegebenenfalls dem Objekt zugeordnet sind (in Byte).

createTime
Datentyp: db2Uint64

Der Zeitpunkt der Erstellung eines Objekts. Der Wert von **createTime** entspricht dem ANSI-C-Wert time_t.

db2ACS_ReadList - Datenstruktur der DB2 ACS-API

db2ACS_ReadList enthält eine Liste mit Gruppen.

```
/* The ReadList will only be used for snapshots where the action is READ, and
 * where one of the granularity modifiers other than BY_OBJ has been specified.
 * In the typical usage scenario of ( READ | BY_OBJ ) the ReadList parameter
 * should be ignored.
 *
 * When the action is DB2ACS_ACTION_BY_GROUP the union is to be interpreted
 * as a group list.
```

```
 * ------------------------------------------------------------------------ */
typedef union db2ACS_ReadList
{
   db2ACS_GroupList           group;
} db2ACS_ReadList;
```

group Datentyp: db2ACS_GroupList

db2ACS_GroupList enthält eine Liste der Gruppen, die in die Momentaufnahmebackup-Operation eingefügt werden sollen.

db2ACS_ReturnCode - Datenstruktur der DB2 ACS-API

db2ACS_ReturnCode enthält Diagnoseinformationen, einschließlich Nachrichtentext und Fehlercodes, die sich auf die Speicherhardware beziehen. Der Inhalt des Parameters db2ACS_ReturnCode für einen DB2 ACS-API-Funktionsaufruf wird in den Diagnoseprotokollen des Datenbankmanagers erfasst.

```
/* ========================================================================
 * Storage Adapter Return Code and Diagnostic Data.
 *
 * These will be recorded in the DB2 diagnostic logs, but are intended to be
 * internal return and reason codes from the storage layers which can be used
 * in conjunction with the DB2ACS_RC to provide more detailed diagnostic info.
 * ======================================================================== */
typedef struct db2ACS_ReturnCode
{
   int                        returnCode;
   int                        reasonCode;
   char                       description[DB2ACS_MAX_COMMENT_SZ + 1];
} db2ACS_ReturnCode;
```

returnCode
Datentyp: int

Rückkehrcode, der sich auf die Speicherhardware bezieht.

reasonCode
Datentyp: int

Ursachencode, der sich auf die Speicherhardware bezieht.

description
Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_COMMENT_SZ + 1.

db2ACS_SessionInfo - Datenstruktur der DB2 ACS-API

db2ACS_SessionInfo enthält alle Informationen zur DB2 ACS-Sitzung.

```
/* ========================================================================
 * Session Info
 * ======================================================================== */
typedef struct db2ACS_SessionInfo
{
   db2ACS_DB2ID               db2ID;

   /* Fields identifying the backup session originator.
    * ------------------------------------------------------------------- */
      SQL_PDB_NODE_TYPE dbPartitionNum;
   char                       db[SQL_DBNAME_SZ + 1];
   char                       instance[DB2ACS_MAX_OWNER_SZ + 1];
   char                       host[SQL_HOSTNAME_SZ + 1];
   char                       user[DB2ACS_MAX_OWNER_SZ + 1];
   char                       password[DB2ACS_MAX_PASSWORD_SZ + 1];
```

```
   /* The fully qualified ACS vendor library name to be used.
    * ------------------------------------------------------------------------ */
   char                       libraryName[DB2ACS_MAX_PATH_SZ + 1];
} db2ACS_SessionInfo;
```

db2ID Datentyp: db2ACS_DB2ID

db2ACS_DB2ID identifiziert IBM Data Server.

dbPartitionNum
Datentyp: SQL_PDB_NODE_TYPE

Die eindeutige, numerische Kennung für eine Datenbankpartition.

db Datentyp: char[]

Eine Zeichenfolge mit der Länge SQL_DBNAME_SZ + 1.

instance
Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_OWNER_SZ + 1.

host Datentyp: char[]

Eine Zeichenfolge mit der Länge SQL_HOSTNAME_SZ + 1.

user Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_OWNER_SZ + 1.

password
Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_PASSWORD_SZ + 1.

libraryName
Datentyp: char[]

Eine Zeichenfolge mit der Länge DB2ACS_MAX_PATH_SZ + 1.

db2ACS_SnapshotDetails - Datenstruktur der DB2 ACS-API

db2ACS_SnapshotDetails enthält Informationen zu einer Momentaufnahmebackup-Operation.

```
typedef struct db2ACS_SnapshotDetails
{
   char                       imageTimestamp[SQLU_TIME_STAMP_LEN + 1];
} db2ACS_SnapshotDetails;
```

imageTimestamp
Datentyp: char[]

Eine Zeichenfolge mit der Länge SQLU_TIME_STAMP_LEN + 1.

db2ACS_MetaData - Datenstruktur der DB2 ACS-API

db2ACS_MetaData speichert Metadaten zu Momentaufnahmebackups.

```
/* ============================================================================
 * The metadata structure itself is internal to DB2 and is to be treated by
 * the storage interface as an unstructured block of data of the given size.
 * ============================================================================ */
typedef struct db2ACS_MetaData
{
   db2Uint64                  size;
   void                     * data;
} db2ACS_MetaData;
```

size Datentyp: db2Uint32

Die Größe des Parameters **data** (in Byte).

data Datentyp: `void *`

Ein Verweis auf einen Speicherblock, den der Datenbankmanager zum Speichern von Momentaufnahmebackup-Metadaten verwendet.

db2ACS_VendorInfo - Datenstruktur der DB2 ACS-API

`db2ACS_VendorInfo` enthält Informationen zum DB2 ACS-API-Treiber.

```
/* ************************************************************************
 * Storage Vendor Identifier
 * ************************************************************************ */
typedef struct db2ACS_VendorInfo
{
   void                     * vendorCB;              /* Vendor control block  */
   db2Uint32                  version;               /* Current version       */
   db2Uint32                  release;               /* Current release       */
   db2Uint32                  level;                 /* Current level         */
   char                       signature[DB2ACS_MAX_VENDORID_SZ + 1];
} db2ACS_VendorInfo;
```

vendorCB

Datentyp: `void *`

Verweis auf einen Steuerblock, der sich auf den DB2 ACS-API-Treiber bezieht.

version

Datentyp: `db2Uint32`

Version des DB2 ACS-API-Treibers.

release

Datentyp: `db2Uint32`

Release-Level des DB2 ACS-API-Treibers.

level Datentyp: `db2Uint32`

Aktualitäts-ID des DB2 ACS-API-Treibers.

signature

Datentyp: `db2ACS_VendorSignature`

Eine Zeichenfolge mit der Länge `DB2ACS_MAX_VENDORID_SZ + 1`.

DB2 ACS - Best Practices

Beachten Sie bei Installation und Konfiguration von DB2 ACS (Advanced Copy Services, erweiterte Kopierservices) die folgenden Best Practices.

Angabe einer dedizierten Datenträgergruppe für Protokollpfade

Es empfiehlt sich, Protokollpfade zu verwenden, die auf dem zugehörigen Momentaufnahmedatenträger enthalten und somit unabhängig vom Datenbankverzeichnis und Datenbankcontainern sind.

Angabe einer Datenträgergruppe für jede einzelne Datenbankpartition

In einer DPF-Umgebung (Database Partitioning Feature, Datenbankpartitionierungsfunktion) muss sich jede einzelne Datenbankpartition auf einer Gruppe von Momentaufnahmedatenträgern befinden, die unabhängig von den übrigen Datenbankpartitionen ist.

DB2 ACS - Einschränkungen

Bei der Installation von DB2 ACS (Advanced Copy Services, erweiterte Kopierservices) sind verschiedene Einschränkungen zu beachten.

Eine Datenträgerfreigabe wird nicht unterstützt. Wenn sich eine Datenbankpartition auf demselben Speicherdatenträger befindet wie eine andere Datenbankpartition, sind Momentaufnahmeoperationen nicht zulässig.

Rückkehrcodes der DB2 ACS-API

Die Funktionen der DB2 ACS-API geben eine definierte Gruppe von möglichen Rückkehrcodes zurück.

Tabelle 27. Rückkehrcodes der DB2 ACS-API

Rückkehrcode	Beschreibung
DB2ACS_RC_OK	Die Operation war erfolgreich.
DB2ACS_RC_LINK_EXIST	Die Sitzung wurde zuvor aktiviert.
DB2ACS_RC_COMM_ERROR	Es ist ein Kommunikationsfehler mit einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.
DB2ACS_RC_INV_VERSION	Die Version der DB2 ACS-Bibliothek des Datenbankmanagers und die Version des DB2 ACS-API-Treibers sind nicht kompatibel.
DB2ACS_RC_INV_ACTION	Der Datenbankmanager hat eine ungültige Aktion vom DB2 ACS-API-Treiber angefordert.
DB2ACS_RC_NO_DEV_AVAIL	Derzeit ist keine Speichereinheit, wie z. B. ein Bandlaufwerk, verfügbar.
DB2ACS_RC_OBJ_NOT_FOUND	Der DB2 ACS-API-Treiber konnte das vom Datenbankmanager angegebene Momentaufnahmebackup-Objekt nicht finden.
DB2ACS_RC_OBJS_FOUND	Der DB2 ACS-API-Treiber hat mehrere Momentaufnahmebackup-Objekte gefunden, die mit der Spezifikation des Datenbankmanagers übereinstimmen.
DB2ACS_RC_INV_USERID	Der Datenbankmanager hat eine ungültige Benutzer-ID an den DB2 ACS-API-Treiber übergeben.
DB2ACS_RC_INV_PASSWORD	Der Datenbankmanager hat ein ungültiges Kennwort an den DB2 ACS-API-Treiber übergeben.
DB2ACS_RC_INV_OPTIONS	Der Datenbankmanager hat ungültige Optionen angegeben.
DB2ACS_RC_INIT_FAILED	Der Datenbankmanager hat versucht, eine DB2 ACS-Sitzung zu initialisieren, aber die Initialisierung ist fehlgeschlagen.
DB2ACS_RC_INV_DEV_HANDLE	Der Datenbankmanager hat eine ungültige Kennung für eine Speichereinheit übergeben.
DB2ACS_RC_BUFF_SIZE	Der Datenbankmanager hat eine ungültige Puffergröße angegeben.
DB2ACS_RC_END_OF_DATA	Der DB2 ACS-API-Treiber kann keine weiteren Momentaufnahmebackup-Objekte finden.
DB2ACS_RC_END_OF_TAPE	Die Speichereinheit hat unerwarteterweise das Ende des Backup-Bandes erreicht.
DB2ACS_RC_DATA_RESEND	Eine Speichereinheit, wie z. B. ein Bandlaufwerk, hat den Datenbankmanager aufgefordert, den neuesten Datenpuffer erneut zu senden.
DB2ACS_RC_COMMIT_FAILED	Der DB2 ACS-API-Treiber konnte eine Transaktion nicht festschreiben.

Tabelle 27. Rückkehrcodes der DB2 ACS-API (Forts.)

Rückkehrcode	Beschreibung
DB2ACS_RC_DEV_ERROR	Es ist ein Fehler in einer Speichereinheit, wie z. B. einem Bandlaufwerk, aufgetreten.
DB2ACS_RC_WARNING	Die Speicherhardware hat eine Warnung zurückgegeben. Weitere Informationen hierzu finden Sie in den Diagnoseprotokollen des Datenbankmanagers.
DB2ACS_RC_LINK_NOT_EXIST	Die Sitzung wurde zuvor nicht aktiviert.
DB2ACS_RC_MORE_DATA	Es sind weitere Daten zur Übertragung von der Speicherposition an den Datenbankmanager vorhanden.
DB2ACS_RC_ENDOFMEDIA_NO_DATA	Die Speichereinheit hat das Ende des Speicherdatenträgers erreicht und keine Daten gefunden.
DB2ACS_RC_ENDOFMEDIA	Die Speichereinheit hat das Ende des Speicherdatenträgers erreicht.
DB2ACS_RC_MAX_LINK_GRANT	Die maximale Anzahl an Verbindungen wurde hergestellt. Der Datenbankmanager kann keine weiteren Verbindungen herstellen.
DB2ACS_RC_IO_ERROR	Der DB2 ACS-API-Treiber hat einen Fehler festgestellt, der auf Eingabe- oder Ausgabeoperationen zurückzuführen ist.
DB2ACS_RC_DELETE_FAILED	Der DB2 ACS-API-Treiber konnte die vom Datenbankmanager angegebenen Momentaufnahmebackup-Objekte nicht löschen.
DB2ACS_RC_INV_BKUP_FNAME	Der Datenbankmanager hat einen ungültigen Dateinamen für das Momentaufnahmebackup-Objekt angegeben.
DB2ACS_RC_NOT_ENOUGH_SPACE	Der DB2 ACS-API-Treiber geht davon aus, dass nicht genügend Speicherplatz vorhanden ist, um ein Momentaufnahmebackup der vom Datenbankmanager angegebenen Datenbank auszuführen.
DB2ACS_RC_ABORT_FAILED	Der Datenbankmanager hat versucht, eine DB2 ACS-Operation abzubrechen, aber der Versuch ist fehlgeschlagen.
DB2ACS_RC_UNEXPECTED_ERROR	Der DB2 ACS-API-Treiber hat einen schwer wiegenden, unbekannten Fehler festgestellt.
DB2ACS_RC_NO_DATA	Der DB2 ACS-API-Treiber hat keine Daten an den Datenbankmanager zurückgegeben.
DB2ACS_RC_OBJ_OUT_OF_SCOPE	Der Datenbankmanager hat versucht, eine DB2 ACS-Operation für ein Recoveryobjekt auszuführen, das nicht vom DB2 ACS-API-Treiber verwaltet wird.
DB2ACS_RC_INV_CALL_SEQUENCE	Der Datenbankmanager hat DB2 ACS-API-Funktionen in einer ungültigen Sequenz aufgerufen. Der Datenbankmanager muss beispielsweise db2ACSInitialize aufrufen, bevor andere DB2 ACS-API-Funktionen aufgerufen werden können (Ausnahme: db2ACSQueryAPIVersion).
DB2ACS_RC_SHARED_STORAGE_GROUP	Der Datenbankmanager hat versucht, eine Momentaufnahmenoperation für ein Speicherobjekt auszuführen, das von einer anderen Datenbank oder Anwendung verwendet wird.

DB2 Advanced Copy Services (ACS) - Unterstützte Betriebssysteme und Hardware

In IBM Data Server ist ein DB2 ACS-API-Treiber integriert, der einen Teil der von IBM Data Server unterstützten Betriebssysteme und Hardware unterstützt.

Tabelle 28. DB2 ACS-API - Unterstützte Betriebssysteme und Hardware

Hardware	Betriebssystem AIX mit SAN-Speicher [2]	Betriebssystem AIX mit NFS-Speicher[2]	Betriebssystem Linux mit NFS-Speicher[1]
IBM TotalStorage SAN Volume Controller	Vollständige Unterstützung	Keine Unterstützung	Keine Unterstützung
IBM System Storage DS6000	Vollständige Unterstützung mit folgender Ausnahme: • Die Erstellung von Teilkopien wird nicht unterstützt.	Keine Unterstützung	Keine Unterstützung
IBM System Storage DS8000	Vollständige Unterstützung mit folgender Ausnahme: • Die Erstellung von Teilkopien wird nicht unterstützt.	Keine Unterstützung	Keine Unterstützung
IBM System Storage N Series	Vollständige Unterstützung	Vollständige Unterstützung	Vollständige Unterstützung
NetApp V-series	Vollständige Unterstützung	Vollständige Unterstützung	Vollständige Unterstützung
NetApp FAS Series	Vollständige Unterstützung.	Vollständige Unterstützung.	Vollständige Unterstützung.

[1]Mit DB2 ACS und Linux werden nur die folgenden Systeme unterstützt:

- 64-Bit nur mit x86-Prozessoren (Intel Pentium, Intel Xeon und AMD)
- POWER (IBM eServer OpenPower-, System i- oder pSeries-Systeme, die Linux unterstützen)

[2] Nur AIX 5.3 wird mit DB2 ACS auf AIX unterstützt. Bei AIX 6.1-Betriebssystemen können Sie einen manuellen Kopiervorgang ausführen. Sie können z. B. die Schreibein-/ausgabe aussetzen und die Funktionalität der geteilten Spiegeldatenbank wie im Abschnitt„Verwenden einer geteilten Spiegeldatenbank als Backup-Image” auf Seite 255 beschrieben nutzen. Ferner haben Sie die Möglichkeit, die Vollversion von Tivoli Storage Manager for Advanced Copy Services V6.1 für AIX 6.1 zu erwerben und zu installieren.

Teil 3. 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.

Wenn Sie IBM dabei unterstützen möchten, die Benutzerfreundlichkeit der IBM Information Management-Produkte weiter zu verbessern, können Sie uns Ihr Feedback mithilfe der Umfrage zur Verbraucherfreundlichkeit von Software geben: http://www.ibm.com/software/data/info/consumability-survey/ (landessprachliche Version unter https://www-950.ibm.com/survey/oid/wsb.dll/studies/consumabilitywebform.htm?renderlang=de).

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/e-business/linkweb/publications/servlet/pbi.wss zur Verfügung steht. Englische und übersetzte Versionen der DB2 Version 9.5-Handbücher im PDF-Format können über die folgende Adresse heruntergeladen werden: www.ibm.com/support/docview.wss?uid=swg27009727 bzw. www.ibm.com/support/docview.wss?uid=swg27009728.

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 29. Technische Informationen zu DB2

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
Administrative API Reference	SC23-5842-03	Ja	Dezember 2010
Administrative Routines and Views	SC23-5843-03	Nein	Dezember 2010
Call Level Interface Guide and Reference, Volume 1	SC23-5844-03	Ja	Dezember 2010
Call Level Interface Guide and Reference, Volume 2	SC23-5845-03	Ja	Dezember 2010
Command Reference	SC23-5846-03	Ja	Dezember 2010
Dienstprogramme für das Versetzen von Daten - Handbuch und Referenz	SC12-3917-03	Ja	Dezember 2010
Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz	SC12-3919-03	Ja	Dezember 2010
Datenserver, Datenbanken und Datenbankobjekte	SC12-3912-03	Ja	Dezember 2010
Datenbanksicherheit	SC12-3914-03	Ja	Dezember 2010
Developing ADO.NET and OLE DB Applications	SC23-5851-02	Ja	April 2009
Developing Embedded SQL Applications	SC23-5852-02	Ja	April 2009
Developing Java Applications	SC23-5853-03	Ja	Dezember 2010

Tabelle 29. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
Developing Perl and PHP Applications	SC23-5854-02	Nein	April 2009
Developing User-defined Routines (SQL and External)	SC23-5855-03	Ja	Dezember 2010
Getting Started with Database Application Development	GC23-5856-03	Ja	Dezember 2010
Installation und Verwaltung von DB2 unter Linux und Windows - Erste Schritte	GC12-3922-03	Ja	Dezember 2010
Internationalisierung	SC12-3916-02	Ja	April 2009
Fehlernachrichten, Band 1	GI11-3098-00	Nein	
Fehlernachrichten, Band 2	GI11-3099-00	Nein	
Migration	GC12-3921-03	Ja	Dezember 2010
Net Search Extender - Verwaltung und Benutzerhandbuch	SC12-3979-02	Ja	April 2009
Partitionierung und Clustering	SC12-3915-03	Ja	Dezember 2010
Query Patroller - Verwaltung und Benutzerhandbuch	SC12-3977-01	Ja	April 2009
IBM Data Server-Clients - Einstieg	GC12-3924-03	Nein	Dezember 2010
DB2-Server - Einstieg	GC12-3923-03	Ja	Dezember 2010
Spatial Extender und Geodetic Data Management Feature - Benutzer- und Referenzhandbuch	SC12-3978-02	Ja	April 2009
SQL Reference, Volume 1	SC23-5861-03	Ja	Dezember 2010
SQL Reference, Volume 2	SC23-5862-03	Ja	Dezember 2010
Systemmonitor - Handbuch und Referenz	SC12-3918-03	Ja	Dezember 2010
Text Search	SC12-3920-02	Ja	Dezember 2010
Fehlerbehebung	GI11-3097-03	Nein	Dezember 2010
Optimieren der Datenbankleistung	SC12-3913-03	Ja	Dezember 2010
Lernprogramm für Visual Explain	SC12-3932-00	Nein	
Neuerungen	SC12-3928-03	Ja	Dezember 2010
Workload-Manager - Handbuch und Referenz	SC12-3929-03	Ja	Dezember 2010

Tabelle 29. Technische Informationen zu DB2 (Forts.)

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
pureXML - Handbuch	SC12-3930-03	Ja	Dezember 2010
XQuery - Referenz	SC12-3931-02	Nein	April 2009

Tabelle 30. Technische Informationen zu DB2 Connect

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
DB2 Connect Personal Edition - Einstieg	GC12-3926-03	Ja	Dezember 2010
DB2 Connect-Server - Einstieg	GC12-3927-03	Ja	Dezember 2010
DB2 Connect - Benutzerhandbuch	SC12-3925-03	Ja	Dezember 2010

Tabelle 31. Technische Informationen zu Information Integration

Name	IBM Form	In gedruckter Form verfügbar	Letzte Aktualisierung
Information Integration: Föderierte Systeme - Verwaltung	SC12-3759-01	Ja	
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-02	Ja	
Information Integration: Konfiguration föderierter Datenquellen	SC12-3777-01	Nein	
Information Integration: SQL Replication - Handbuch und Referenz	SC12-3782-01	Ja	
Information Integration: Replikation und Event-Publishing - Einführung	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 420.

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:

```
? sqlstatus oder ? klassencode
```

Hierbei steht *sqlstatus* 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 9.8 lautet die URL der *DB2-Informationszentrale* http://publib.boulder.ibm.com/infocenter/db2luw/v9r8/.

Für Themen aus DB2 Version 9.7 lautet die URL der *DB2-Informationszentrale* http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/.

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 9.1 lautet die URL der *DB2-Informationszentrale* http://publib.boulder.ibm.com/infocenter/db2luw/v9/.

Für Themen aus DB2 Version 8 lautet die URL der *DB2-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 ein 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 die Verknüpfung an 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* die Position des Programmdateiverzeichnisses ist.
     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
     ```

Anmerkung: Die Stapeldatei `help_end` enthält die Befehle, die erforderlich sind, um die Prozesse, die mit der Stapeldatei `help_start` gestartet wurden, ordnungsgemäß zu beenden. Verwenden Sie nicht die Tastenkombination `Strg+C` oder eine andere Methode, um `help_start.bat` zu beenden.

- Unter Linux: Navigieren Sie in das Verzeichnis `doc/bin` des Installationsverzeichnisses und führen Sie das Script `help_end` aus:

  ```
  help_end
  ```

  Anmerkung: Das Script `help_end` enthält die Befehle, die erforderlich sind, um die Prozesse, die mit dem Script `help_start` gestartet wurden, ordnungsgemäß zu beenden. Verwenden Sie keine andere Methode, um das Script `help_start` zu beenden.

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-Datenbankprodukten zu unterstützen.

DB2-Dokumentation

Informationen zur Fehlerbehebung stehen im Handbuch DB2-Fehlerbehebung oder im Abschnitt mit grundlegenden Informationen zu Datenbanken in 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-Datenbankprodukte 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/support/db2_9/ 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
U59/3600
3600 Steeles Avenue East
Markham, Ontario L3R 9Z7
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 Gewährleistung, 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 davon 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 veröffentlichen 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 kann Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes enthalten. Sie sollen nur die Funktionen des Lizenzprogramms 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 kann Musteranwendungsprogramme enthalten, die in Quellensprache geschrieben sind und Programmiertechniken in verschiedenen Betriebsumgebungen veranschaulichen. 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

IBM, das IBM Logo und ibm.com sind Marken oder eingetragene Marken der IBM Corporation in den USA und/oder anderen Ländern. Weitere Produkt- oder Servicenamen können Marken von IBM oder anderen Herstellern sein. Eine aktuelle Liste der IBM Marken finden Sie auf der Webseite Copyright and trademark information unter www.ibm.com/legal/copytrade.shtml.

Die folgenden 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 auf Java basierenden Marken und Logos 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 anderen Ländern.
- Intel, das Intel-Logo, Intel Inside, das Intel Inside-Logo, Intel Centrino, das Intel Centrino-Logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium und Pentium sind Marken oder eingetragene Marken der Intel Corporation oder deren Tochtergesellschaften in den USA oder anderen Ländern.
- Microsoft, Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.

Weitere Unternehmens-, Produkt- oder Servicenamen können Marken anderer Hersteller sein.

Index

A

AIX
  Unterstützung von Backup und Restore 220
Aktive Protokolle 5
Aktualisierende Recovery
  Aspekte der Protokollverwaltung 163
  Datenbank 295
  Konfigurationsdateiparameter 59
  Protokollfolge 163
  Tabellenbereich 295, 350
Aktualisierungen
  DB2-Informationszentrale 425
Alternative Server
  Beispiele 202
  identifizieren 21
Anstehende Aktionen, Statusangabe
  Beschreibungen 190, 248
Archivierte Protokolle
  offline 7
  online 7
Archivierung von Protokolldateien
  auf Band 165
Archivprotokollierung 7, 70, 172
Assistent für automatische Verwaltung konfigurieren 261
ASYNC
  Synchronisationsmodus 46
Ausgefallener Datenbankpartitionsserver 287
Ausgesetzte E/A und Plattenspiegelung 187
Ausgesetzte E/A zur Unterstützung der fortlaufenden Verfügbarkeit 15
Automatische Clientweiterleitung
  Beispiele 202
  Beschreibung 18
  einrichten 18
  Einschränkungen 21
  HADR (High Availability Disaster Recovery) 29
  High Availability Disaster Recovery (HADR) 194
  Literaturübersicht 9
  Verbindungsfehler 20
Automatische Funktionsübernahme
  AIX 140
  Solaris-Betriebssystem 151
  Sun Cluster 3.0 154
  Übersicht 151
Automatische Reorganisation
  Konfiguration
    Beispiel 57
Automatische Statistikerfassung
  Konfiguration
    Beispiel 57
Automatische Verwaltung 261
  AUTOMAINT_GET_POLICY 56
  AUTOMAINT_GET_POLICYFILE 56
  AUTOMAINT_SET_POLICY 56
  AUTOMAINT_SET_POLICYFILE 56
  Backup 213, 261
  Konfiguration
    abrufen 56
  konfigurieren 56
  Richtlinienspezifikation
    Beispiel 57
Automatischer inkrementeller Restore
  Einschränkungen 302
Automatischer Neustart 281
Automatisches Backup
  aktivieren 261
  Beispielkonfiguration 57

B

Backup
  automatisch 261
  automatisiert 213
  Band 256
  benannte Pipes 258
  Benutzerexitprogramm 218
  CLP-Beispiele 266
  Containernamen 249
  Datenbanken
    automatisch 261
  Einschränkungen von Betriebssystemen 220
  Häufigkeit 216
  Images 249
    Protokolldateien einschließen in 174
  Informationen anzeigen 249
  Informationen zum Speicher 218
  inkrementell 298
  offline 216
  online 216
  partitionierte Datenbanken 259
Backup auf Band 256
BACKUP DATABASE, Befehl 252
Backup-Dienstprogramm
  Beispiele 266
  Einschränkungen 252
  erforderliche Berechtigungen und Zugriffsrechte 264
  Fehlerbehebung 249
  Fortschritt überwachen 247
  Informationen anzeigen 249
  Leistung 262
  Übersicht 249
Backupkomprimierung 219
Bandlaufwerke
  Protokolldateien speichern auf 70, 165
Bedarfsgesteuerte Protokollarchivierung 165
Bedingungen
  Verwendung der Veröffentlichungen 429
Befehle
  db2adutl
    knotenübergreifende Recovery, Beispiel 268
Begrenzungen
  Protokollfolgenummern 236
    Zustand eines erreichten Grenzwerts beheben 241
Beispiele
  alternativer Server 202
  automatische Clientweiterleitung 202
  automatische Verwaltung
    Konfiguration 57
Bemerkungen 431
Benannte Pipes
  Backup 258
Benutzerdefinierte Ereignisse 140

Benutzerexitprogramme
Archivierung von Protokolldateien 70
Aufrufformat 169
Backup 218
Beispielprogramme
UNIX 168
Windows 168
Datenbankrecovery 167
Fehlerbehandlung 170
Protokolldateien abrufen 70
Protokolle 218
Bereitschaftsdatenbank
Statusangaben 190
Bereitschaftsmoduskonfiguration
Beschreibung 140
Bestellen von DB2-Büchern 422
Beziehungen
zwischen Tabellen 219
blk_log_dsk_ful, Konfigurationsparameter
Details 59
Bücher
gedruckt
bestellen 422

C

Client-Kommunikationsfehler 18
Clientweiterleitung
automatisch 18
Beispiele 202
Einschränkungen 21
HADR (High Availability Disaster Recovery) 29
CLP (Befehlszeilenprozessor)
Beispiele
aktualisierende Sitzungen 356
Backup 266
Restore von Sitzungen durchführen 333
Sitzungen für umgeleiteten Restore 333
Sitzungen zum erneuten Erstellen von Datenbanken 335
Cluster
HACMP 140
Cluster-Management-Software
DB2 High Availability Instance Configuration Utility (db2haicu) 99
Ressource 100
Ressourcengruppe 100
unterstützt 140
Clusterdomäne
Netzschnittstellenkarten (NICs) 101
Clusterdomänen 98
Datenbankpartitionen 103
IP-Netzadressen 101
Mountpunkte 103
Netzersatzsysteme 101
Netzprotokolle 101
Pfade 103
Teilnetzmasken 101
Clustering 147
IP-Adressübernahme 5
Überwachungssignalfunktion 5
Clusterverwaltung
HADR (High Availability Disaster Recovery) 43
Container
Namen 249

D

Datei des Recoveryprotokolls
abgelaufen
Eintragsstatus 224
aktiv
Eintragsstatus 224
bereinigen 244
automatisiert 231
Befehl PRUNE HISTORY 230
db2Prune, API 230
do_not_delete (Nicht löschen)
Eintragsstatus 224, 233
Einträge
bereinigen 230
schützen 233
Eintragsstatus 224
inaktiv
Eintragsstatus 224
Daten
Paritätsstriping nach Sektoren (RAID-Stufe 5) 284
Recovery
Übersicht 211
Datenbanken
aktualisierende Recovery
Übersicht 295
Backup
automatisiert 261
Strategie 213
erneut erstellen
Beispiele 335
Einschränkungen 332
Images des inkrementellen Backups 330
partitionierte 330
Tabellenbereichscontainer 324
Übersicht 318
Zielimageauswahl 325
in HADR-Umgebung aktivieren 178
in HADR-Umgebung inaktivieren 178
Konfiguration
HADR 38
mehrere Verbindungen 38
nicht wiederherstellbar 213
Protokollierung
Konfigurationsparameter 59
Übersicht 5
Umlaufprotokollierung 6
Recovery
Strategie 213
temporäre Tabellenbereiche 324
Datenbankkonfigurationsparameter
autorestart 281
Datenbanknummer
Restore
neue Datenbanken 311
vorhandene Datenbanken 310
Datenbankobjekte
Datei des Recoveryprotokolls 213
Protokolldatei der Tabellenbereichsänderungen 213
Protokolldatei für die Recovery 213
Datenbankpartitionen
Uhren synchronisieren 161
Datenbankpartitionsserver
Recovery nach Fehler 291
Datenbankrollen tauschen
HADR (High Availability Disaster Recovery) 209
Datenbankserver
alternativ 21

Datenträgerfehler
  Aspekte von Katalogpartitionen 284
  Auswirkungen begrenzen 284
  Protokolle 218
DB_HISTORY, Verwaltungssicht
  Einträge in der Datei des Recoveryprotokolls anzeigen 228
DB2 ACS-API
  Datenstrukturen
    db2ACS_QueryInput 408
DB2 Advanced Copy Services (ACS)
  aktivieren 365, 367
  Best Practices 412
  Einschränkungen 413
  installieren 366
  Konfigurationsscript 'setup.sh' 370
  konfigurieren 368
  Übersicht 365
  Verzeichnis 369
DB2 Advanced Copy Services (ACS), API
  Betriebssysteme 415
  Datenstrukturen
    db2ACS_BackupDetails 398
    db2ACS_CB 399
    db2ACS_CreateObjectInfo 400
    db2ACS_DB2ID 401, 408
    db2ACS_GroupList 401
    db2ACS_LoadcopyDetails 401
    db2ACS_LogDetails 402
    db2ACS_MetaData 411
    db2ACS_ObjectInfo 402
    db2ACS_ObjectStatus 404
    db2ACS_OperationInfo 404
    db2ACS_Options 405
    db2ACS_PartitionEntry 406
    db2ACS_PartitionList 406
    db2ACS_PathEntry 406
    db2ACS_PathList 407
    db2ACS_QueryOutput 408
    db2ACS_ReadList 409
    db2ACS_ReturnCode 410
    db2ACS_SessionInfo 410
    db2ACS_SnapshotDetails 411
    db2ACS_VendorInfo 412
    Übersicht 398
  Funktionen
    db2ACSBeginOperation 377
    db2ACSBeginQuery 380
    db2ACSDelete 393
    db2ACSEndOperation 379
    db2ACSEndQuery 385
    db2ACSGetNextObject 382
    db2ACSInitialize 371
    db2ACSPartition 388
    db2ACSPrepare 375
    db2ACSQueryApiVersion 371
    db2ACSRetrieveMetaData 396
    db2ACSSnapshot 386
    db2ACSStoreMetaData 395
    db2ACSTerminate 373
    db2ACSVerify 391
    Übersicht 371
  Hardware 415
  Rückkehrcodes 413
  Übersicht 371
DB2 High Availability Feature
  Cluster-Manager
    API 140
    Integration 75
  Clusterkonfiguration 96
  IBM Tivoli SA MP Base Component 76
DB2 High Availability Feature (HA)
  Übersicht 12
DB2 High Availability Instance Configuration Utility (db2haicu)
  ausführen
    interaktiver Modus 107
    XML-Eingabedatei 107, 126
  Beispieleingabedatei
    db2ha_sample_DPF_mutual.xml 127
    db2ha_sample_DPF_NPlusM.xml 130
    db2ha_sample_HADR.xml 133
    db2ha_sample_sharedstorage_mutual.xml 126
  Beschreibung 104
  Clusterdomänen 98
    erstellen 136
    verwalten 136
  Clusterumgebung 97
  Einschränkungen 138
  erkannte Datenbankpfade 136
  Fehlerbehebung 138
  Quorumeinheiten 101
  Startmodus 105
  Wartungsmodus 106
  XML-Schema für Eingabedateien 108
    ClusterDomainType 110
    ClusterNodeType 116
    CustomPolicyType 122
    DB2ClusterType 108
    DB2PartitionSetType 118
    DB2PartitionType 119
    FailoverPolicyType 116
    HADBDefn 125
    HADBType 124
    HADRDBDefn 124
    HADRDBType 122
    InterfaceType 114
    IPAddressType 115
    MountType 120
    MutualPolicyType 121
    NPlusMPolicyType 121
    PhysicalNetworkType 113
    QuorumType 111
DB2-Informationszentrale
  Aktualisierung 425
  in verschiedenen Sprachen anzeigen 424
  Sprachen 424
  Versionen 424
DB2_LOGGER_NON_BUFFERED_IO
  nicht gepufferte E/A 73
db2adutl, Befehl
  knotenübergreifende Recovery, Beispiel 268
db2Backup, API
  Backup von Daten 252
db2diag.log, Datei
  Protokoll mit Benachrichtigungen für die Systemverwaltung, Nachrichten 200
db2fm, Befehl
  Übersicht zum Fehlermonitor 10
db2inidb, Befehl
  geteilte Spiegeldatenbank erstellen 255
  Übersicht 15

db2pd
Protokollfolgenummern ermitteln 238
db2Recover, API
Recovery von Daten 267
db2Restore, API
Recovery für Datenbank oder Tabellenbereich 308
db2Rollforward, API
Transaktionen auf wiederhergestelltes Backup-Image anwenden 349
db2tapemgr, Befehl
Protokolldateien auf Band archivieren 165
db2uext2, Programm
Aufrufformat 169
Beschreibung 167
Dienstprogramm EXPORT
Online-Backup, Kompatibilität 264
Direkte E/A (Direct I/O, DIO)
unterstützte Konfiguration 73
Dokumentation
gedruckt 420
Nutzungsbedingungen 429
PDF 420
Übersicht 419
Doppelte Protokollierung 14
Duplizierung
RAID-Stufe 1 284

E

Einschränkungen
HADR (High Availability Disaster Recovery) 54
Ereignis node_down 140
Ereignis node_up 140
Ereignismonitore
High Availability Cluster Multi-Processing (HACMP) für AIX 140
Erfassung statistischer Daten
automatisch 261
erneut erstellen
ausgewählte Tabellenbereiche 318
Erneut erstellen
ausgewählte Tabellenbereiche 328
Datenbanken 318
mit Images des inkrementellen Backups 330
Tabellenbereichscontainer 324
Erstellen
Klondatenbank 184

F

Fehler
Protokoll voll 59
Fehlerbehebung
Lernprogramme 428
Onlineinformationen 428
Fehlerbestimmung
Lernprogramme 428
verfügbare Informationen 428
Fehlermonitor 200
Konfiguration 24
db2fm 25
db2fmc 26
Fehlermonitorfunktion (Fault Monitor Facility) 10
Fehlertoleranz 151
Fernes Catch-up, Status 190
Fernes Catch-up anstehend, Status 190
Fortlaufende Verfügbarkeit 151
Funktionsübernahme 4, 199, 202
Bereitschaftsmodus (Idle Standby) 4
gegenseitige Übernahme (Mutual Takeover) 4
HADR (High Availability Disaster Recovery) 204
Richtlinien für Funktionsübernahme
benutzerdefiniert 102
gegenseitige Übernahme 102
HADR (High Availability Disaster Recovery) 102
lokaler Neustart 102
M+N 102
Umlauf 102

G

Gelöschte Tabelle, Recovery
Beschreibung 279
GET SNAPSHOT, Befehl
Status der HADR-Bereitschaftsdatenbank 193
Geteilte Spiegeldatenbank
als Backup-Image 255
als Bereitschaftsdatenbank 45
als Klondatenbank 184
Handhabung 15

H

HACMP (High Availability Cluster Multi-Processing)
Beschreibung 140
HADR, Datenbankrollen tauschen 207
HADR (High Availability Disaster Recovery)
Anforderungen 52
Initialisierung 50
automatische Clientweiterleitung 29
Befehle 194
Bereitschaftsdatenbank
initialisieren 44
Status 193
synchronisieren 187
Cluster-Manager 43
Datenbanken
aktivieren 178
inaktivieren 178
initialisieren 27
Datenbankrollen tauschen 207
Einschränkungen 54
Funktionsübernahme 204
Konfiguration 31
konfigurieren 27
Ladeoperationen 31
Leistung 41
nicht replizierte Operationen 189
primäre Reintegration 209
Protokollarchivierung 39
replizierte Operationen 188
schrittweisen Upgrade ausführen 179
schrittweises Upgrade 179
automatisierte High Availability Disaster Recovery (HADR) 180
schrittweises Upgrade ausführen 180
stoppen 177
Synchronisationsmodi 46
Systemanforderungen 50
Übersicht 11
Überwachung 201
Unterstützung 50

HADR (High Availability Disaster Recovery) *(Forts.)*
verwalten 194
Zurücksetzung 209
hadr_peer_window, Datenbankkonfigurationsparameter
Parameter konfigurieren 38
hadr_timeout, Konfigurationsparameter
Parameter konfigurieren 38
Hardware
Platteneinheiten 284
High Availability Disaster Recovery (HADR)
siehe HADR (High Availability Disaster Recovery) 11
Hilfe
Konfiguration der Sprache 424
SQL-Anweisungen 423
Hintereinanderschaltende Zuordnung 140
Hohe Verfügbarkeit
Entwurf 1, 50
Fehlermonitor
konfigurieren (Befehl 'db2fm') 25
konfigurieren (db2fmc und Systembefehle) 26
Registrierungsdatei 24
Übersicht 200
IBM Data Server, Funktionen 9
konfigurieren
AUTO_DEL_REC_OBJ, Parameter 244
Clusterumgebung 74
Übersicht 17
Microsoft Failover Clustering 147
Protokollübertragung 13
Solaris-Betriebssystem 151
Strategien
Clustering 5, 140
Funktionsübernahme 4
Redundanz 3, 187
Übersicht 3
Sun Cluster 3.0 154
Systemausfälle
feststellen 199, 200
Handhabung 199, 202
Übersicht 1
Tivoli System Automation for Multiplatforms 76, 144
Überwachungssignalfunktion 200
Verwaltung 163
Auswirkungen minimieren 176
automatisch 176
manuell 176
Übersicht 163
HP unter IPF
Backup 220
Restore 220
HP-UX
Backup 220
Restore 220

I

IBM Tivoli SA MP Base Component
Deinstallationsprotokoll 93
deinstallieren 76, 88
mit dem DB2-Installationsprogramm 89
mit uninstallSAM 90
hohe Verfügbarkeit 144
Installation 76
Installationsprotokoll 93
installieren 77
mit dem DB2-Installationsprogramm 78
mit installSAM 80
IBM Tivoli SA MP Base Component *(Forts.)*
Installieren neuerer Versionen 81, 87
Lizenzbedingungen 94
Systemanforderungen 94
Übersicht 76
Upgrade durchführen 76, 82
mit dem DB2-Installationsprogramm 83
mit installSAM 86
IBM Tivoli SA MP Base Component, HADR-Scripts
deinstallieren 91
manuell 92
mit dem DB2-Installationsprogramm 91
installieren 91
manuell 92
mit dem DB2-Installationsprogramm 91
Upgrade durchführen 91
manuell 92
mit dem DB2-Installationsprogramm 91
IBM Tivoli Storage Manager (TSM)
mit Befehl BACKUP DATABASE
mindestens erforderliche Version 361
mit Befehl RESTORE DATABASE
mindestens erforderliche Version 361
verwenden 361
Images
Backup 249
Images des inkrementellen Backups
bei der erneuten Erstellung einer Datenbank verwenden 330
Indizes
Protokollierung für High Availability Disaster Recovery (HADR) 30
Inkrementeller Restore 300, 312
Inkrementelles Backup und inkrementelle Recovery 298

K

Keepalive-Pakete
Beschreibung 140
Klondatenbank
erstellen 184
Knoten
Synchronisation 161
Knotenübergreifende Datenbankrecovery, Beispiel 268
Komprimierung
Backup 219
Konfiguration
Datenbanken
HADR 38
Fehlermonitor 24
db2fm 25
db2fmc 26
HADR (High Availability Disaster Recovery) 31
Konfiguration zur gegenseitigen Übernahme 140
Konfigurationsparameter
auto_del_rec_obj 244
Datenbankprotokollierung 58, 59
hadr_peer_window
Optimierung 38
hadr_timeout
Optimierung 38
logarchopt1
knotenübergreifende Recovery, Beispiel 268
TCP_KEEPALIVE 20
vendoropt
knotenübergreifende Recovery, Beispiel 268

Konfigurieren
für hohe Verfügbarkeit 17
Konsistenzzustand
Datenbank 281

L

Leistung
High Availability Disaster Recovery (HADR) 41
Recovery 303
Lernprogramme
Fehlerbehebung 428
Fehlerbestimmung 428
Visual Explain 427
Linux
Unterstützung von Backup und Restore
AMD64 und Intel EM64T 220
IA-64 220
Power PC 220
zSeries 220
LIST HISTORY (Befehl) 223
Literaturübersichten
automatische Clientweiterleitung 9
logarchmeth1, Konfigurationsparameter
und HADR (High Availability Disaster Recovery) 39
logarchmeth2, Konfigurationsparameter
und HADR (High Availability Disaster Recovery) 39
logarchopt1, Konfigurationsparameter
knotenübergreifende Recovery, Beispiel 268
logbufsz, Datenbankkonfigurationsparameter
Übersicht 59
logfilsiz, Datenbankkonfigurationsparameter
Übersicht 59
logfilsiz, Konfigurationsparameter
und HADR (High Availability Disaster Recovery) 31
logprimary, Datenbankkonfigurationsparameter
Übersicht 59
logretain, Datenbankkonfigurationsparameter
Übersicht 59
logsecond, Konfigurationsparameter
Details 59
Lokales Catch-up, Status 190
LSN
siehe Protokollfolgenummern (LSNs)

M

maxRetriesForClientReroute 18
Mehrere Instanzen
Tivoli Storage Manager (TSM) 364
Microsoft Failover Clustering, Server (MSCS) 147
mincommit, Datenbankkonfigurationsparameter
Übersicht 59
mirrorlogpath, Datenbankkonfigurationsparameter
Übersicht 59
mirrorlogpath, Konfigurationsparameter 14
Momentaufnahmebackup 254, 367
Momentaufnahmebackup-Objekte verwalten 245
Restore durchführen 309

N

NEARSYNC, Synchronisationsmodus 46
newlogpath, Datenbankkonfigurationsparameter
Übersicht 59
Nicht gepufferte E/A 73
Nicht wiederherstellbare Datenbanken
Backup und Recovery 213

O

Offline-Backup
Kompatibilität mit Online-Backup 264
OFFLINE LOAD
Kompatibilität mit Online-Backup 264
Offlinearchivprotokolldateien 7
Online
archivierte Protokolle 7
Online-Backup
Kompatibilität mit anderen Dienstprogrammen 264
ONLINE CREATE INDEX
Kompatibilität mit Online-Backup 264
ONLINE INDEX REORG
Kompatibilität mit Online-Backup 264
ONLINE INSPECT
Kompatibilität mit Online-Backup 264
ONLINE LOAD
Kompatibilität mit Online-Backup 264
ONLINE TABLE REORG
Kompatibilität mit Online-Backup 264
Optimieren
Restoreleistung 332
Optimierung
Backup-Leistung 262
overflowlogpath, Datenbankkonfigurationsparameter 59

P

Parallele Recovery 303
Partitionierte Datenbanken, Umgebungen mit
Backup
Übersicht 259
Neuerstellung von Datenbanken 330
Partitionierte Tabellen
Backup 260
Peerstatus
Beschreibungen 190
Platten
Redundant Array of Independent Disks (RAID) 284
Störungen beheben 284
Striping 284
Platteneinheiten
Übersicht 284
Plattenspiegelung
RAID-Stufe 1 284
Primärdatenbank, Verbindung
Verbindung trennen 38
Primäre Reintegration
HADR (High Availability Disaster Recovery) 209
Protokoll mit Benachrichtigungen für die Systemverwaltung 200, 281
Protokollarchivierung
Konfiguration 39
Protokolldatei
Zugriff 228
Protokolldatei, Verwaltung
ACTIVATE DATABASE, Befehl 163
Protokolldateien
Archivieren 70
in Backup-Image einschließen 174
Protokollsteuerdateien 8
Verwaltung 200

Protokolldateispeicher
Begrenzungen 236
Protokolle
aktiv 5
Archivprotokollierung 7, 172
bedarfsgesteuert archivieren 165
Benutzerexitprogramm 218
Datenbank 5
entfernen 172
erforderlicher Speicher 218
Indizes
HADR (High Availability Disaster Recovery) 30
Konfiguration für nicht gepufferte E/A
DB2_LOGGER_NON_BUFFERED_IO 73
Konfigurationsparameter 58
offline archiviert 7
online archiviert 7
spiegeln 14
Umlaufprotokollierung 6, 172
Verlust verhindern 175
verwalten 163
Verzeichnis, voll 70
Zuordnung 172
Protokolle bedarfsgesteuert archivieren 165
Protokollfolgenummern
Begrenzungen
zugehörige Fehlercodes 241
Protokollfolgenummern (LSNs)
aktuelle Werte
feststellen 238
Begrenzungen
beheben 241
Übersicht 236
Übersicht 235
Zunahme überwachen 238
Zunahmeraten
berechnen 240
Zeit bis zur Erreichung des Grenzwerts berechnen 240
Protokollspiegelung 187
Protokollsteuerdateien
Übersicht 8
Protokollübertragung 187
hohe Verfügbarkeit 13
Proxy-Knoten
Tivoli Storage Manager (TSM)
Konfiguration 361

Q

Quorumeinheiten 101

R

RAID-Einheiten (Redundant Array of Independent Disks)
Stufe 1 (Plattenspiegelung oder -duplizierung) 284
Stufe 5 (Daten- und Paritätsstriping nach Sektoren) 284
REBALANCE
Kompatibilität mit Online-Backup 264
RECOVER DATABASE, Befehl 267
erforderliche Berechtigungen und Zugriffsrechte 305
Recovery
aktualisierende Recovery 295
benötigte Zeit 216
beschädigte Tabellenbereiche 282, 283
bis Ende der Protokolle 295
Datenbank erneut erstellen 318
Recovery *(Forts.)*
Datenbanken
Übersicht 267
Einschränkungen von Betriebssystemen 220
gelöschte Tabellen 279
Informationen zum Speicher 218
inkrementell 298
knotenübergreifend, Beispiel 268
Leistung 303
nach dem Ausfall eines Datenbankpartitionsservers 291
parallel 303
Protokolldatei 223
Protokollierung reduzieren 68
Systemabsturz 281
Übersicht über die Strategie 213
Version 294
zeitpunktgesteuert 295
Recovery durchführen
Protokoll des zweiphasigen Commits 287
Recovery nach einem Katastrophenfall
Übersicht 293
Recovery nach Katastrophenfall
HADR (High Availability Disaster Recovery)
Anforderungen 52
Übersicht 11
Recovery nach Systemabsturz 281
Recoveryobjekte
Verwaltung 235, 242, 243, 244
automatisiert 243
db2Prune 242
PRUNE HISTORY 242
Redundant Array of Independent Disks (RAID)
Auswirkungen von Datenträgerausfällen begrenzen 284
Redundanz 3
Reduzieren
Auswirkungen von Datenträgerfehlern 284
Auswirkungen von Transaktionsfehlern 287
Protokollierung
mit deklarierten temporären Tabellen 68
mit dem Parameter NOT LOGGED INITIALLY 68
Regeln
Datei 140
Registrierdatenbankvariable
DB2_HADR_PEER_WAIT_LIMIT 41
Registrierdatenbankvariablen
db2_connretries_interval 18
db2_max_client_connretries 18
REORG TABLE
Kompatibilität mit Online-Backup 264
Reorganisation
automatisch 261
replizierte Operationen
HADR (High Availability Disaster Recovery) 188
Replizierte Operationen
HADR (High Availability Disaster Recovery) 189
Ressource 100
Ressourcengruppe 100
RESTART DATABASE, Befehl 281
Restore
automatisch, inkrementell
Einschränkungen 302
Datenbanken
aktualisierende Recovery 295
inkrementell 298
inkrementell 300, 312
von Daten, in einer neuen Datenbank 311
von Daten, in einer vorhandenen Datenbank 310

RESTORE DATABASE, Befehl 308
Restore durchführen
von einem Momentaufnahmebackup 309
Restoredienstprogramm
Beispiele 333
Einschränkungen 308
erforderliche Berechtigungen und Zugriffsrechte 333
Fortschritt überwachen 247
Kompatibilität mit Online-Backup 264
Leistung 307, 332
Restore in einer neuen Datenbank 311
Restore in einer vorhandenen Datenbank 310
Tabellenbereichscontainer erneut definieren 314
Übersicht 307
retryIntervalForClientReroute 18
ROLLFORWARD, Dienstprogramm
Beispiele 356
Einschränkungen 349
erforderliche Berechtigungen und Zugriffsrechte 355
Fortschritt überwachen 247
Kompatibilität mit Online-Backup 264
Mindestrecoveryzeit 259, 350
Recovery einer gelöschten Tabelle 279
Übersicht 347
ROLLFORWARD DATABASE, Befehl 349
Rotierende Zuordnung 140
Rückkehrcodes
Benutzerexitprogramme 170
RUNSTATS, Dienstprogramm
Kompatibilität mit Online-Backup 264

S

Schrittweises Upgrade
ausführen 179, 180
Server
alternativ 18, 21
Server, Funktionsübernahmeunterstützung
Windows 147
Server-Cluster 147
SET WRITE
Kompatibilität mit Online-Backup 264
Skalierbarkeit
Datenbanken in mehreren Clustern 140
Softwareplatteneinheiten 284
Solaris-Betriebssystem
Unterstützung von Backup und Restore 220
Sommerzeit 185
SP-Rahmen 140
Speicher
Anforderungen
Backup und Recovery 218
Datenträgerfehler 218
spiegeln
Protokolle 14
SQL-Anweisungen
Hilfe anzeigen 423
SQL1849C, Fehlercode
durch Erreichen des Grenzwerts für Protokollfolgenummern 241
SQL1850C, Fehlercode
durch Erreichen des Grenzwerts für Protokollfolgenummern 241
Standortausfall
HADR (High Availability Disaster Recovery) 11
START HADR, Befehl 194
Statistikprofilerstellung
automatisch 261
Statusangaben
Bereitschaftsdatenbank 190
für anstehende Aktionen 248
STOP HADR, Befehl 194
Stoppen
HADR (High Availability Disaster Recovery) 177
Sun Cluster 3.0
hohe Verfügbarkeit 154
SYNC
Synchronisationsmodus 46
Synchronisation
Aspekte der Recovery 161
Datenbankpartition 161
Knoten 161
Synchronisationsmodi
HADR (High Availability Disaster Recovery) 46
Synchronisationspunktmanager (SPM)
Recovery unbestätigter Transaktionen 291
Systemanforderungen
HADR (High Availability Disaster Recovery) 50
Systemuhr 185

T

Tabellen
Beziehungen 219
Tabellenbereiche
aktualisierende Recovery 295, 350
Container
Neuerstellung von Datenbanken 324
erneut erstellen 318, 328
Recovery 282, 283
Restore 295
Tabellenbereichscontainer erneut definieren
Restoredienstprogramm 314
mit einem Script 315
TAKEOVER HADR, Befehl 194
Datenbankrollen tauschen 207
Funktionsübernahme durchführen 204
TCP_KEEPALIVE
Konfigurationsparameter des Betriebssystems 20
Temporäre Tabellenbereiche
und Datenbank erneut erstellen 324
Tivoli Storage Manager (TSM)
Clientkonfiguration 361
dsmapipw (Befehl) 361
partitionierte Tabellen 260
Serverkonfiguration 364
Transaktion nach Fehler 281
Transaktionen
blockieren bei vollem Protokollverzeichnis 70
Recovery nach Fehler
Abstürze 287
auf dem aktiven Datenbankpartitionsserver 287
auf dem ausgefallenen Datenbankpartitionsserver 287
Auswirkungen von Fehlern begrenzen 281

U

Überwachung
Fortschritt
aktualisierend wiederherstellen 247
Backup 247
Recovery nach Systemabsturz 247

Überwachung *(Forts.)*
Fortschritt *(Forts.)*
Restore 247
HADR (High Availability Disaster Recovery) 201
Überwachungssignale
High Availability Cluster Multi-Processing (HACMP) für AIX 140
Solaris 151
Überwachung 199, 200
Umgebungen mit partitionierten Datenbanken
Transaktionen
Recovery nach Fehler 287
Umgeleiteter Restore 314
mit einem Script 315
mit generiertem Script 317
Umlaufprotokollierung 6, 172
Unbestätigte Transaktionen
Recovery
auf dem Host 291
mit dem DB2-Synchronisationspunktmanager 291
ohne den DB2-Synchronisationspunktmanager 292
Ungeplante Betriebsunterbrechung
feststellen 200
userexit, Datenbankkonfigurationsparameter
Details 59

V

vendoropt, Konfigurationsparameter
knotenübergreifende Recovery, Beispiel 268
Verbindungen
Fehler 38
Verbindungsfehler
automatische Clientweiterleitung 20
VERITAS Cluster Server 157
hohe Verfügbarkeit 157
Versionen
Versionsrecovery der Datenbank 294
Verwalten
HADR (High Availability Disaster Recovery) 194
Verwaltung
Terminierung 55
Verwaltungssichten
DB_HISTORY 228
Visual Explain
Lernprogramm 427
Vorbedingungen 97, 107, 135

W

Wiederherstellbare Datenbanken
Beschreibung 213
Windows-Betriebssysteme
Funktionsübernahme 147

Z

Zeit
für Datenbankrecovery benötigte Zeit 216
Zeitänderung 185
Zeitmarken
konvertieren
Client/Server-Umgebung 162
Zielimage
für die erneute Erstellung 325
Zu diesem Handbuch
Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz vii
Zugriffsrechte
Backup-Dienstprogramm 264
Restoredienstprogramm 333
ROLLFORWARD, Dienstprogramm 355
Zurücksetzung
Operationen 209
Zweiphasiges Commit
Protokoll 287

SC12-3919-03

Spine information:

DB2 Version 9.5 für Linux, UNIX und Windows **Version 9 Release 5**

Datenrecovery und hohe Verfügbarkeit - Handbuch und Referenz