



DB2 Connect 사용자 안내서



DB2 Connect 사용자 안내서

주:

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 201 페이지의 부록 B 『주의사항』의 일반 정보를 읽으십시오.

개정판 주의사항

이 문서에는 IBM에서 소유하고 있는 정보가 있습니다. 이는 라이선스 계약에 따라 제공한 것이며 저작권의 보호를 받습니다. 이 책의 정보에는 제품 보증이 포함되지 않으며, 이 매뉴얼에서 제공된 어떠한 문장도 이와 같이 해석할 수 없습니다.

온라인으로 IBM 서적을 주문하거나 로컬 IBM 담당자를 통해 서적을 주문할 수 있습니다.

- 온라인으로 서적을 주문하려면 IBM Publications Center(www.ibm.com/shop/publications/order)로 이동하십시오.
- 로컬 IBM 담당자를 찾으려면 IBM Directory of Worldwide Contacts(www.ibm.com/planetwide)로 이동하십시오.

미국 또는 캐나다의 DB2 Marketing and Sales에서 DB2 서적을 주문하려면 1-800-IBM-4YOU(426-4968)로 전화하십시오.

IBM은 귀하가 IBM으로 보낸 정보를 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 사용하거나 배포할 수 있습니다.

목차

이 책에 대한 정보 v

제 1 부 DB2 Connect 개념 1

제 1 장 DB2 Connect 3
DB2 Connect 제품 3
DB2 Connect 버전 8에 전달된 기능 3
호스트 데이터베이스 5
DB2 Connect 및 SQL문 5
DB2 Connect 관리 유틸리티 6
InfoSphere Federation Server 및 DB2 Connect . . . 7

제 2 장 DRDA(Distributed Relational Database Architecture) 9
DRDA 및 데이터 액세스 9
DB2 Connect 및 DRDA 10
리모트 작업 단위(RUOW) 11
분산 요청(DR) 12

제 3 장 DB2 Connect 시나리오 15
호스트 데이터베이스에 직접 액세스 15
DB2 Connect Personal Edition을 사용하여 System z 호스트 또는 IBM i DB2 데이터에 액세스 . . . 17
DB2 Connect 서버 제품을 연결 서버로 사용 . . . 19
DB2 Connect 및 웹 응용프로그램 20
DB2 Connect 및 IBM WebSphere 21
Java 응용프로그램 서버로서의 DB2 Connect . . . 22
웹 서버에서 DB2 Connect 23
DB2 Connect 및 응용프로그램 서버(AS) 24
DB2 Connect 및 트랜잭션 처리 모니터 27

제 2 부 DB2 Connect 참조 31

제 4 장 데이터베이스 디렉토리 갱신 33
시스템 데이터베이스 디렉토리 값 33
노드 디렉토리 값 34
DCS 디렉토리 값 35
디렉토리 사용자 정의 워크시트 40
동일한 데이터베이스에 대해서 여러 항목 정의 . . . 41
BiDi 데이터 처리 41

제 5 장 DB2 Connect 보안 45
DB2 Connect를 통한 트러스트된 연결 45

CLI를 통해 트러스트된 연결 작성 및 종료 . . . 46
CLI를 통해 트러스트된 연결에서 사용자 전환 . . 48
DB2 Connect 인증 고려사항 50
Kerberos 지원 52
z/OS 보안에 대한 힌트 및 추가 정보 52
DB2 Connect에서 지원되는 인증 유형 54

제 6 장 바인딩 응용프로그램 및 유틸리티(DB2 Connect) 57

제 7 장 다중 사이트 갱신 61
제어 센터에서 다중 사이트 갱신 가능 62
제어 센터를 사용한 다중 사이트 갱신 테스트 . . . 62
다중 사이트 갱신 및 동기점 관리 프로그램 63
XA 호환 트랜잭션 관리 프로그램으로 DB2 Connect 구성 64
느슨하게 결합된 트랜잭션에 대한 DB2 Connect 지원 65

제 8 장 DB2 Connect에서 데이터 이동 67

제 9 장 SQLCODE 맵핑 71
SQLCODE 맵핑 해제 71
SQLCODE 맵핑 조정 71

제 10 장 데이터베이스 시스템 모니터링과 DB2 Connect 77
리모트 클라이언트에 대한 연결 모니터링 . . . 77
Windows 성능 모니터를 사용하여 성능 모니터링 . . 77
GET SNAPSHOT 명령 사용 78
DCS 응용프로그램 상태 81
Health Monitor 및 경보 85
z/OS용 DB2 Health Monitor 개요 85
z/OS용 DB2 Health Monitor 시작, 중지 및 새로 고침 86
권장 조치의 보기, 제출 및 저장 88
상태 경보 요약 보기 91
상태 경보 오브젝트 보기 93

제 3 부 고가용성 및 DB2 Connect 95

제 11 장 호스트 데이터베이스 연결을 위한 고가용성 및 로드 밸런스 97

제 12 장 자동 클라이언트 리라우트 설명 및 설정 (DB2 Connect)	99
제 13 장 클라이언트 연결 분배기 기술에 대한 자동 클라이언트 리라우트 구성	101
제 4 부 성능 조정 및 DB2 Connect	103
제 14 장 DB2 Connect 성능 고려사항	105
제 15 장 ODBC 액세스 최적화.	109
제 16 장 응용프로그램 설계	111
제 17 장 연결 관리	115
연결 풀링	115
연결 집중기(connection concentrator)	117
연결 풀링 및 연결 집중기(connection concentrator)	123
WebSphere MQ Transaction Manager 및 DB2에 필요한 연결 집중기(connection concentrator)	124
제 18 장 DB2 Connect 서버 Sysplex 지원	125
System z SYSPLEX 사용에 대한 고려사항.	125
DB2 Sysplex 사용	126
Sysplex의 구성 요구사항	127
제 19 장 클라이언트 Sysplex 지원	129
트랜잭션 레벨 워크로드 밸런싱(클라이언트 측)	129
트랜잭션 레벨 워크로드 밸런싱 구성(클라이언트 측).	131
자동 클라이언트 리라우트(클라이언트 측)	133
자동 클라이언트 리라우트 구성(클라이언트 측)	135
XA 지원(클라이언트 측)	137
XA 지원 사용(클라이언트 측).	138
클라이언트 친화도 구성	139
클라이언트 Sysplex 지원 사용의 한계	144
제 20 장 DB2 Connect 조정	147
호스트 데이터베이스 조정	149
네트워크 조정 고려사항	149
시스템 자원 경합	151
DB2 Connect 성능 문제점 해결.	151
z/OS용 DB2 조정	152
DB2 Connect 데이터 전송률 증가	152
추가 쿼리 블록.	153
RFC-1323 창 스케일	154
호스트 데이터 변환	155
문자 데이터의 데이터 유형.	155

네트워크 하드웨어.	155
제 21 장 CLI/ODBC 응용프로그램 성능 조정	159
제 5 부 문제점 해결	161
제 22 장 DB2 Connect 문제점 해결	163
관련 정보 수집.	163
초기 연결에 실패한 경우	163
초기 연결 후 문제점이 발생한 경우.	164
진단 도구	166
제 23 장 DB2 Connect 내의 DB2 추적.	167
db2trc를 사용하여 DB2 추적 확보	167
DB2 추적 파일 덤프.	168
DB2 추적 파일 포매팅	169
제 24 장 DRDA 추적 파일	171
추적 유틸리티	171
추적 출력	172
추적 출력 파일 분석	173
출력 파일 샘플 추적	174
DRDA 추적을 위한 후속 버퍼 정보.	179
제 6 부 메시지	181
제 25 장 DB2 Connect의 일반적인 문제점	183
제 7 부 부록	187
부록 A. DB2 기술 정보 개요.	189
DB2 기술 라이브러리(하드카피 또는 PDF 형식)	190
인쇄된 DB2 서적 주문	192
명령행 처리기에서 SQL 상태 도움말 표시.	193
DB2 정보 센터의 다른 버전에 액세스	194
DB2 정보 센터에서 원하는 언어로 항목 표시	194
컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신	195
컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신	196
DB2 지습서.	198
DB2 문제점 해결 정보	199
이용약관	199
부록 B. 주의사항	201
색인	205

이 책에 대한 정보

*DB2 Connect 사용자 안내서*에서는 학습 및 DB2[®] Connect[™] 제품 사용에 필요한 모든 정보를 제공합니다. DB2 Connect 개념은 DB2 Connect와 기타 네트워크 환경 파트 간의 관계를 나타내는 일반적인 시나리오와 함께 설명되어 있습니다. 데이터베이스 디렉토리, 시스템 간 보안, 다중 사이트 갱신, 데이터 이동 및 DB2 Connect 모니터링과 관련된 고려사항이 설명되어 있습니다. DB2 Connect에서 사용자 네트워크 환경에 고가용성을 지원하는 방법이 나와 있습니다. DB2 Connect별 및 네트워크 전체의 좋은 성능이 소개되어 있으며, 이는 가능한 문제점 해결에 대한 주제의 일부입니다.

이 책의 사용자

시스템 관리자, 데이터베이스 관리자 및 시스템 통신 전문가가 이 책의 일부 또는 전체를 사용합니다.

제 1 부 DB2 Connect 개념

제 1 장 DB2 Connect

DB2 Connect는 Linux®, UNIX® 및 Windows® 운영 체제에서 실행되는 e-business 및 기타 응용프로그램을 위해 IBM® 메인프레임 데이터베이스에 빠르고 강력한 연결을 제공합니다.

DB2 Connect Personal Edition은 System z® 및 IBM Power Systems® 서버에 대한 직접 연결을 제공하지만, DB2 Connect 서버 제품은 클라이언트가 DB2 Connect 게이트웨이를 통해 System z 및 IBM Power Systems 서버에 액세스할 수 있도록 하는 간접 연결을 제공합니다. 여러 DB2 Connect 서버 제품은 환경에 적절한 제품을 선택할 수 있도록 고유 패키징 및 라이선스 부여 솔루션을 제공합니다.

DB2 Connect 제품

DB2 Connect에는 DB2 Connect Personal Edition 및 다수의 DB2 Connect 서버 제품을 포함하는 몇 가지 연결 솔루션이 있습니다.

- DB2 Connect Enterprise Edition
- DB2 Connect Application Server Edition
- System z용 DB2 Connect Unlimited Edition
- System i®용 DB2 Connect Unlimited Edition

DB2 Connect 제품에 대한 자세한 정보는 www.ibm.com/software/data/db2/db2connect/를 참조하십시오.

DB2 Connect 버전 8에 전달된 기능

이 절에는 DB2 Connect 버전 8에 소개된 개선된 기능의 요약이 있습니다. DB2 Connect 기능에 영향을 미치는 DB2® 버전 9에 소개된 변경사항의 목록을 찾으려면 다음 주제를 참조하십시오.

- DB2 Connect 버전 9.5 FixPack 요약
- DB2 Connect 버전 9.1 FixPack 요약

DB2 Connect 버전 8 릴리스 2에 제공된 기능

DB2 Connect 버전 8.2에서는 다음 기능이 향상되었습니다.

- 자동 클라이언트 리라우트

서버 또는 DB2 Connect Server에 대한 TCP/IP 연결이 끊어지는 경우, 대체 서버가 존재하면 클라이언트가 자동으로 연결을 재설정하려고 시도합니다. 대체 서버는 서버 인스턴스에 지정되어 있고 해당 위치는 연결 중에 클라이언트에 전송됩니다.

- 데이터 암호화

클라이언트/서버 통신은 네트워크에서 사용자 데이터 암호화를 제공합니다.

DB2 Connect 버전 8 릴리스 1에 제공된 기능(모든 FixPack 및 수정 레벨 포함)

DB2 Connect 버전 8.1에서는 다음 기능이 향상되었습니다.

- 긴 SQL문 지원(최대 2MB)

최대 2MB의 SQL문이 CLI와 JDBC 응용프로그램을 통해 흐를 수 있습니다. 그러나 임베디드(embedded) 인터페이스에서는 64K가 한계입니다.

- SQL문의 원점을 식별하는 진단 정보

z/OS®용 DB2 동적 SQL문 캐시로 특정 명령문을 실행한 응용프로그램을 판별하는 기능을 제공합니다.

- 컬럼 방식 입력 배열

응용프로그램이 하나의 SQL문에 여러 매개변수 세트를 제공할 수 있습니다.

- 네트워크 시간 모니터링

새 모니터 요소를 사용하여 데이터베이스 또는 응용프로그램 레벨에서 보다는 데이터베이스 활동 및 네트워크 트래픽 정보를 얻을 수 있습니다.

- DB2 CLI 동적 화면 이동 커서 지원

이제, z/OS용 DB2 Universal Database(UDB) 버전 8.1 이상인 서버에 액세스할 때 DB2 CLI에서 동적 화면 이동 커서가 지원됩니다.

- eWLM 지원

병목 현상을 판별하기 위해 미들웨어 그룹을 통해 끝에서 끝까지 작업 단위를 모니터링할 수 있는 기능을 제공합니다.

- DB2 ping 명령 개선사항

DB2 ping 명령은 이제 요청 및 응답 패킷 크기의 지정을 지원합니다.

주: DB2 Connect는 버전 9 게이트웨이를 통해 버전 7 클라이언트에서 호스트로 발행될 때 PING 명령을 지원하지 않습니다.

호스트 데이터베이스

데이터베이스라는 용어는 이 책 전체에서 관계형 데이터베이스 관리 시스템(RDBMS)을 설명하는 데 사용됩니다. DB2 Connect가 통신하는 다른 시스템에서는 데이터베이스라는 용어를 사용하여 약간 다른 개념을 설명할 수 있습니다. DB2 Connect 용어 데이터베이스는 또한 다음을 참조할 수 있습니다.

System z

z/OS용 DB2. LOCATION NAME으로 식별되는 z/OS용 DB2 서브시스템. LOCATION NAME은 TSO에 로그인해서, 사용 가능한 쿼리 도구 중 하나를 사용하여 다음 SQL 쿼리를 발행함으로써 결정될 수 있습니다.

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

LOCATION NAME은 부트 스트랩 데이터 세트(BSDS)에 정의되며 분산 데이터 기능(DDF)이 시작될 때 작성되는 DSNL004I 메시지(LOCATION=location)에도 정의됩니다. LOCATION NAME은 최대 8자의 별명 위치 이름을 지원하며, 응용프로그램이 다른 db 별명 이름을 사용하여 버전 8 z/OS 서버에 액세스할 수 있도록 합니다. z/OS -display ddf 명령을 사용하여 DB2 서버 위치 이름, 도메인 이름, IP 주소 및 포트를 알아내십시오.

VSE DBNAME으로 식별되는 데이터베이스 파티션에서 실행되는 VSE용 DB2

VM 해당 DBNAME으로 식별되는 CMS 가상 머신에서 실행하는 VM용 DB2

IBM Power Systems 서버

IBM i 운영 체제의 필수 부분인 IBM i용 DB2. 시스템이 독립적인 보조 스토리지 풀을 사용하도록 구성되어 있지 않으면 IBM Power Systems 서버에 하나의 데이터베이스만 존재할 수 있습니다.

DB2 Connect 및 SQL문

DB2 Connect는 응용프로그램이 제출한 SQL문을 IBM 메인프레임 데이터베이스 서버로 전달합니다.

DB2 Connect는 거의 유효한 모든 SQL문 및 지원되는 DB2 API(Application Programming Interface)를 전달할 수 있습니다.

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

- PHP
- pureQuery
- Python
- Ruby
- DB2 CLI
- Embedded SQL

Embedded SQL 지원

Embedded SQL 처리에는 정적 SQL과 동적 SQL의 두 유형이 있습니다. 정적 SQL은 SQL문을 먼저 처리하여 실행시 필요한 시간을 최소화합니다. 동적 SQL은 SQL문이 IBM 메인프레임 데이터베이스 서버로 제출될 때 처리됩니다. 동적 SQL이 보다 편리하기는 하나 속도가 느려질 수 있습니다. 정적 SQL이나 동적 SQL의 사용 여부는 응용프로그래머가 결정합니다. 두 유형 모두 DB2 Connect에 의해 지원됩니다.

서로 다른 IBM 메인프레임 데이터베이스 서버는 SQL을 다르게 구현합니다. DB2 Connect는 z/OS용 DB2, DB2 서버(이전의 SQL/DS™) 및 IBM i용 DB2 구현뿐만 아니라, 공통 IBM SQL을 전적으로 지원합니다. IBM SQL은 데이터베이스를 독립적으로 유지보수하기 위해 강력하게 권장됩니다.

DB2 Connect 관리 유틸리티

중요사항: 제어 센터 및 연관된 구성요소는 버전 9.7에서 사용되지 않으며 이후 릴리스에서 제거될 수 있습니다. 자세한 정보는 *버전 9.7의 새로운 내용* 책에 있는 『제어 센터 도구 및 DB2 Administration Server(DAS)는 사용되지 않음』 주제를 참조하십시오.

다음 유틸리티는 DB2 Connect 관리자에게 도움이 될 것입니다.

- 명령행 처리기(CLP)를 사용하여 IBM 메인프레임 데이터베이스 서버 데이터베이스에 대한 SQL문을 발행할 수 있습니다. 이 처리기는 SQL문을 지정한 데이터베이스로 보냅니다.
- DB2 명령 센터는 명령행 처리기(CLP)에 그래픽 인터페이스를 제공합니다.
- импорт 및 익스포트 유틸리티를 사용하여 워크스테이션의 파일 및 IBM 메인프레임 데이터베이스 서버 데이터베이스 간에 서로 데이터를 로드, импорт 및 익스포트할 수 있습니다. 그리고 이 파일을 사용하여 워크스테이션에서 실행 중인 데이터베이스, 스프레드시트, 기타 응용프로그램으로 데이터를 импорт할 수 있습니다.
- DB2 Connect 서버 제품을 실행하는 경우, 이벤트 보기 프로그램 및 성능 모니터를 사용할 수 있습니다. 이벤트 보기 프로그램을 사용하여 DB2 Connect가 기록한 예외 이벤트를 볼 수 있습니다. 성능 모니터를 사용하여 로컬로 또는 리모트로 DB2 Connect 서버의 성능을 모니터 및 관리할 수 있습니다.

- DB2 제어 센터를 사용하여 DB2 Connect 서버의 모든 면을 관리하고 모니터링할 수 있습니다. 또한, 관리자는 z/OS용 DB2 데이터베이스 오브젝트(예: 테이블, 뷰, 버퍼 풀 및 스프레드)로 작업할 수 있습니다.
- 시스템 관리자는 데이터베이스 시스템 모니터 유틸리티를 사용하여 시스템 연결을 모니터링할 수 있습니다. 이 기능은 DB2 Connect가 서버로 작동하는 경우에만 사용할 수 있습니다. 또한 이 유틸리티는 시스템 관리자가 오류의 원인을 알아내는 데 도움을 줍니다. 시스템 관리자는 클라이언트 응용프로그램을 IBM 메인프레임 데이터베이스 서버에서 실행 중인 해당 작업과 상관시킬 수 있습니다.

주: 이전 릴리스에서는 제어 센터와 같은 DB2 그래픽 관리 도구가 모든 플랫폼에서 지원되었습니다. 버전 9에서 현재 DB2 그래픽 관리 도구는 Windows x86, Windows x64 (AMD64/EM64T), x86의 Linux 및 AMD64/EM64T의 Linux에서만 지원됩니다. 모든 플랫폼에서 관리 목적으로 DB2 명령행 처리기(CLP)를 사용할 수 있습니다.

InfoSphere Federation Server 및 DB2 Connect

DB2 Connect는 기존 호스트 및 중간 범위의 서버에 위치한 많은 볼륨의 데이터를 가공할 수 있도록 하는 반면에 InfoSphere™ Federation Server는 여러 벤더의 데이터 소스 데이터에 액세스하고 통합할 수 있도록 하는 별도의 제품입니다.

InfoSphere Federation Server는 데이터 소스의 컬렉션을 단일 소스인 것처럼 보고 조작할 수 있도록 함으로써 정보의 통합을 돕습니다. 이는 호출하는 응용프로그램에 데이터 소스를 완전히 투명하게 볼 수 있도록 합니다. InfoSphere Federation Server는 DB2 Connect 서버 제품과 함께 작동합니다. InfoSphere Federation Server는 DB2 계열 제품, Informix®, Oracle, Sybase, Teradata 및 Microsoft® SQL Server 데이터베이스에 대한 읽기 및 쓰기 액세스를 제공합니다. 또한 InfoSphere Federation Server는 비관계형 데이터 소스 및 Life Sciences 데이터 소스(예: Documentum, IBM Lotus® Extended Search, 테이블 구조 파일 및 XML)에 대한 읽기 액세스를 제공합니다. 이를 사용하여 페더레이티드 시스템 데이터에 대해 쿼리를 공식화할 수 있습니다.

제 2 장 DRDA(Distributed Relational Database Architecture)

DRDA[®](Distributed Relational Database Architecture)는 응용프로그램 뿐만 아니라 IBM 및 비IBM의 다중 데이터베이스 시스템이 함께 작동할 수 있도록 하는 프로토콜 세트입니다. DRDA를 사용하는 관계형 데이터베이스 관리 제품의 모든 조합을 연결하여 분산 관계형 데이터베이스 관리 시스템을 형성할 수 있습니다. DRDA는 교환해야 하는 정보와 교환 방법을 정의하여 시스템 간의 통신을 조정합니다.

작업 단위

작업 단위(UOW)란 하나의 논리적인 트랜잭션입니다. 이 단위는 모든 조치가 성공적으로 실행되거나 아니면 시퀀스 전체로서 실패한 것으로 생각되는 SQL 문의 시퀀스로 구성됩니다.

분산 작업 단위

다중 사이트 갱신이라고도 하는 분산 작업 단위(DUOW)는 하나의 작업 단위(UOW) 내에 둘 이상의 데이터베이스 서버를 포함합니다. DUOW는 다음 특성을 갖습니다.

- 작업 단위(UOW) 당 둘 이상의 데이터베이스 관리 서버가 갱신됩니다.
- 응용프로그램이 작업의 분산을 지시하고 커미트를 시작합니다.
- 작업 단위(UOW) 당 여러 개의 요청이 있을 수 있습니다.
- 요청 당 하나의 데이터베이스 관리 서버가 있습니다.
- 여러 데이터베이스 서버를 통해 커미트를 조정합니다.

DRDA 및 데이터 액세스

DRDA가 데이터베이스 통신 프로토콜을 정의하지만, 응용프로그램 프로그래머가 사용해야 하는 프로그래밍 인터페이스 또는 API를 정의하지는 않습니다. 일반적으로, 응용 프로그램은 DRDA를 사용하여 목표 DRDA 서버가 실행할 수 있는 요청을 전달할 수 있습니다. 오늘날 사용 가능한 모든 DRDA 서버는 응용프로그램이 DB2 Connect를 통해 전달한 SQL 요청을 실행할 수 있습니다.

IBM은 응용프로그램 프로그래머에게 Windows, UNIX 및 Linux 운영 체제용 SQL 요청을 생성하는 도구를 제공합니다. 이러한 도구는 DB2 클라이언트의 일부입니다. DB2 데이터베이스 관리 프로그래밍은 ADO.NET, JDBC, SQLJ, PHP, Perl DBI, Embedded SQL, DB2 콜 레벨 인터페이스(CLI)(DB2 콜 레벨 인터페이스(CLI)) 및 OLE DB와 같은 여러 프로그래밍 인터페이스를 지원합니다. 이 API는 프로그래머가 다양한 프로그래밍 언어로 응용프로그램을 빌드하는 데 사용할 수 있습니다.

DB2 Connect 및 DRDA

DB2 Connect는 DRDA 아키텍처를 구현하여 IBM i용 DB2, IBM Power Systems용 DB2, z/OS용 DB2, DB2 서버 및 기타 DRDA 호환 데이터베이스 서버에 저장된 데이터에 액세스하는 데 따르는 비용 및 복잡성을 줄여줍니다. DB2 Connect는 DRDA 아키텍처를 최대한 활용하여 고객이 요구하는 시스템 관리 특성으로 제대로 수행되는 저비용의 솔루션을 제공합니다.

DRDA 용어에서 응용프로그램 리퀘스터(AR)는 분산 연결의 응용프로그램 쪽을 처리하는 코드입니다. AR은 데이터를 요청하는 응용프로그램입니다. DB2 Connect는 DB2 Connect 워크스테이션의 로컬이 될 수 있는 응용프로그램 대신 또는 DB2 Connect에 대해 리모트로 설치된 개별 클라이언트에서 응용프로그램 리퀘스터로 작동합니다.

응용프로그램 서버(AS)는 연결에 있어서 데이터베이스 쪽을 처리하는 코드입니다.

또한, DRDA는 응용프로그램 리퀘스터와 서버 간에 멀티 티어 연결도 지원합니다. 이 토폴로지에서 응용프로그램 리퀘스터가 연결하는 서버는 응용프로그램 서버이지만, 다운스트림 그 이후의 다른 서버는 응용프로그램 리퀘스터와 직접 상호작용하지 않으므로 데이터베이스 서버(DS)라고 합니다. 또한 데이터베이스 요청을 발행한 시스템도 아니고 요청의 데이터베이스 기능을 수행하는 시스템도 아닌 해당 역할을 강조하기 위해, 응용프로그램 리퀘스터와 마지막 데이터베이스 서버 사이의 각 응용프로그램 서버 또는 데이터베이스 서버를 중간 서버라고도 합니다. 데이터베이스 서버 및 중간 서버의 사용은 DB2 Connect에 의해 지원됩니다.

그림 1은 로컬 클라이언트만 있는 경우, DB2 Connect 워크스테이션과 IBM 메인프레임 서버 간의 데이터 플로우를 나타냅니다.

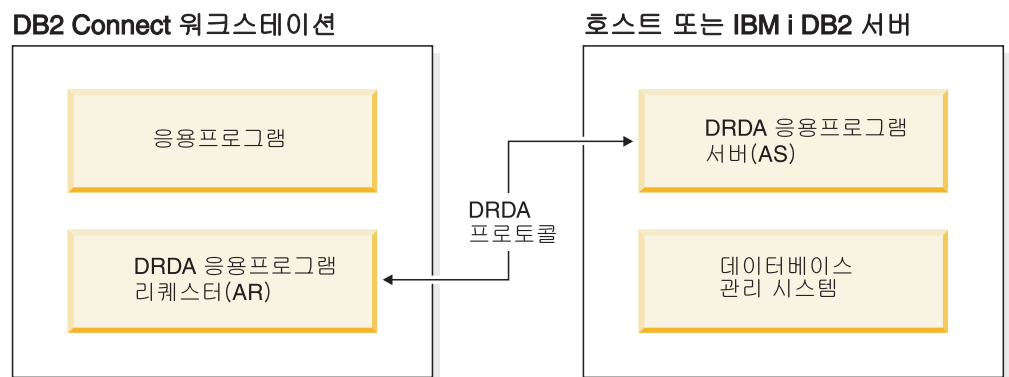


그림 1. DB2 Connect 서버와 IBM 메인프레임 서버 간의 데이터 플로우

DRDA는 다음 아키텍처를 사용하여 DRDA 서버 데이터베이스 관리 시스템과 IBM Data Server Client 간의 연결을 구현합니다.

- Character Data Representation Architecture(CDRA)

- Distributed Data Management Architecture(DDM)
- Formatted Data Object Content Architecture(FD:OCA)
- Transmission Control Protocol/Internet Protocol(TCP/IP).

이러한 아키텍처는 빌딩 블록으로 사용됩니다. 네트워크를 통해 이동되는 데이터 스트림은 분산 관계형 데이터베이스 액세스를 지원하는 데이터 스트림 프로토콜을 기술하는 DRDA 아키텍처가 지정합니다.

요청은 다양한 유형의 통신 정보 및 액세스할 DRDA 서버 데이터베이스의 이름이 있는 디렉토리를 사용하여 올바른 목적지로 라우트됩니다.

리모트 작업 단위(RUOW)

사용자나 응용프로그램에서는 리모트 작업 단위(RUOW)를 사용하여 각 작업 단위(UOW)마다 한 위치에서 데이터를 읽거나 갱신할 수 있습니다. 이 단위는 작업 단위 안에서 하나의 데이터베이스로의 액세스를 지원합니다. 응용프로그램이 여러 리모트 데이터베이스를 갱신하는 동안에는 작업 단위 안에서 하나의 데이터베이스만 액세스할 수 있습니다.

리모트 작업 단위(RUOW)는 다음의 특성을 갖습니다.

- 작업 단위마다 다중 요청(SQL문)이 지원됩니다.
- 작업 단위마다 다중 커서가 지원됩니다.
- 각 작업 단위는 하나의 데이터베이스만 갱신할 수 있습니다.
- 응용프로그램은 작업 단위를 커미트하거나 롤백합니다. 특정 오류 상황에서는 데이터베이스 서버 또는 DB2 Connect가 작업 단위(UOW)를 롤백할 수 있습니다.

예를 들어, 12 페이지의 그림 2는 거래 수수료 스케줄뿐만 아니라, 당좌 및 저축 예금 테이블을 포함하는 데이터베이스에 액세스하는 자금 전송 응용프로그램을 실행 중인 데이터베이스 클라이언트를 나타냅니다. 응용프로그램에서는 다음을 수행하십시오.

- 사용자 인터페이스로부터 전송할 금액을 접수합니다.
- 저축 예금에서 그 금액을 빼서 새로운 차액 잔고를 결정합니다.
- 수수료 스케줄을 읽어서 주어진 차액 잔고를 가진 저축 예금에 대한 트랜잭션을 결정합니다.
- 저축 예금에서 트랜잭션 수수료를 뺍니다.
- 당좌 예금에 전송 금액을 더합니다.
- 예금(작업 단위)을 커미트합니다.

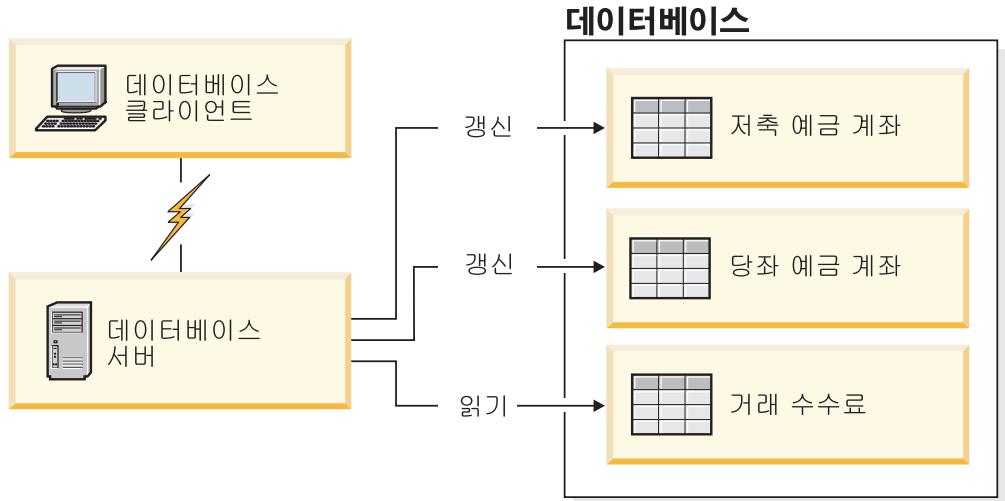


그림 2. 트랜잭션에서 단일 데이터베이스 사용

이러한 응용프로그램을 설정하려면 다음을 수행하십시오.

1. 동일한 데이터베이스에 저축 예금, 당좌 예금 및 거래 수수료 스케줄에 대한 테이블을 작성하십시오.
2. 물리적으로 원격인 경우, 적절한 통신 프로토콜을 사용할 수 있도록 데이터베이스 서버를 설정하십시오.
3. 물리적으로 원격인 경우, 노드와 데이터베이스를 카탈로그하여 데이터베이스 서버에 있는 데이터베이스를 식별하십시오.
4. 응용프로그램을 프리컴파일하여 유형 1 연결을 지정하십시오. 즉, PREP 명령에 CONNECT(1)을 지정하십시오.

분산 요청(DR)

분산 요청(DR)이란 응용프로그램 및 사용자가 단일 명령문으로 둘 이상의 DBMS 또는 데이터베이스를 참조하는 SQL문을 제출할 수 있도록 하는 분산 데이터베이스 기능입니다. 예를 들어, 서로 다른 두 z/OS용 DB2 서브시스템에 있는 테이블 간의 조인이 있습니다.

DB2 Connect는 데이터베이스 및 DBMS 간의 분산 요청을 지원합니다. 예를 들어, DB2 테이블과 Oracle 뷰 간에 UNION 조작을 수행할 수 있습니다. 지원되는 DBMS에는 DB2 계열(예: Linux, UNIX 및 Windows용 DB2 데이터베이스, z/OS용 DB2 및 i 용 DB2) 및 Oracle 구성원이 포함됩니다. 다중 벤더는 InfoSphere Federation Server와 함께 DB2 Connect를 사용할 때 지원됩니다.

분산 요청(DR)에서는 데이터베이스 오브젝트에 대한 위치 투명성을 제공합니다. 테이블 및 뷰에 있는 정보가 이동되면 정보를 요청하는 응용프로그램을 변경시키지 않고도 해당 정보에 대한 참조(별칭이라고 함)를 갱신할 수 있습니다. 또한, 분산 요청은 모든

DB2 SQL 통용어 또는 특정 최적화 기능을 지원하지 않는 DBMS에 대해 보상을 제공합니다. 이런 DBMS에서 수행할 수 없는 조작(예: 재귀 SQL)은 DB2 Connect에서 실행됩니다.

분산 요청은 반자율적 방식으로 기능합니다. 예를 들어, Oracle 오브젝트를 참조하는 DB2 쿼리는 Oracle 응용프로그램이 동일한 서버에 액세스하는 중에 제출될 수 있습니다. 분산 요청은 무결성 및 잠금 제한조건 이외에는 Oracle 또는 다른 DBMS 오브젝트에 대한 액세스를 독점하거나 제한하지 않습니다.

분산 요청 기능을 구현하기 위해서는 DB2 Connect 인스턴스, 페더레이티드 데이터베이스로 사용할 데이터베이스 및 하나 이상의 리모트 데이터 소스가 필요합니다. 페더레이티드 데이터베이스에는 데이터 소스 및 해당 특성을 식별하는 카탈로그 항목이 들어 있습니다. 데이터 소스는 DBMS 및 데이터로 구성됩니다. 응용프로그램은 다른 DB2 데이터베이스처럼 페더레이티드 데이터베이스에 연결합니다. DB2 Connect 페더레이티드 데이터베이스는 사용자 데이터를 관리할 수 있도록 사용이 허가되지 않았었습니다. 이 데이터베이스의 유일한 목적은 데이터 소스 관련 정보를 가지고 있는 것입니다.

페더레이티드 시스템이 설정되고 나면 데이터 소스 내의 정보를 하나의 큰 데이터베이스에 있는 정보로 간주하고 액세스할 수 있습니다. 사용자 및 응용프로그램은 하나의 페더레이티드 데이터베이스로 쿼리를 전송한 후, 필요에 따라 DB2 계열 및 Oracle 시스템에서 데이터를 검색합니다. 사용자 및 응용프로그램은 쿼리에 별칭을 지정합니다. 이 별칭은 데이터 소스에 위치가 지정된 테이블 및 뷰에 대한 참조를 제공합니다. 일반 사용자 관점에서 본다면 별칭(Nickname)은 별명(Alias)과 유사합니다.

많은 요소가 분산 요청의 성능에 영향을 미칠 수 있습니다. 가장 중요한 요소는 데이터 소스와 해당 오브젝트에 대해서 정확한 최신 정보가 페더레이티드 데이터베이스 전역 카탈로그에 저장되는지 확인하는 것입니다. 이 정보는 DB2 옵티마이저에서 사용되며, 데이터 소스에서의 평가를 위해 조작을 푸시다운하는 결정에 영향을 미칠 수 있습니다.

제 3 장 DB2 Connect 시나리오

DB2 Connect는 IBM 메인프레임 데이터베이스 액세스 요구사항을 위한 다양한 솔루션을 제공할 수 있습니다. 이 주제에서는 특정 요구사항이나 환경에 적용되는 여러 시나리오를 대략적으로 설명합니다.

호스트 데이터베이스에 직접 액세스

DB2 Connect의 기본 기능은 워크스테이션에서 실행되는 데스크탑 응용프로그램에서 호스트 데이터베이스로 직접 연결을 제공하는 것입니다. DB2 Connect 라이선스가 적용된 IBM Data Server Driver Package는 이 솔루션을 제공하는 가장 간단한 방법입니다.

DB2 Connect Personal Edition이 설치된 각 워크스테이션에서 z/OS용 DB2, IBM i용 DB2 및 Linux, UNIX 및 Windows용 DB2 데이터베이스 서버에 직접 TCP/IP 연결을 설정할 수 있습니다. 그리고, 응용프로그램이 2단계 커밋 프로토콜에 의해 제공되는 완전한 데이터 무결성을 유지하면서 동일한 트랜잭션에서 여러 개의 DB2 계열 데이터베이스에 연결하거나 갱신할 수 있습니다.

16 페이지의 그림 3은 DB2 Connect Personal Edition이 설치된 워크스테이션에서 IBM 메인프레임 데이터베이스 서버에 직접 연결하는 상황을 나타냅니다.

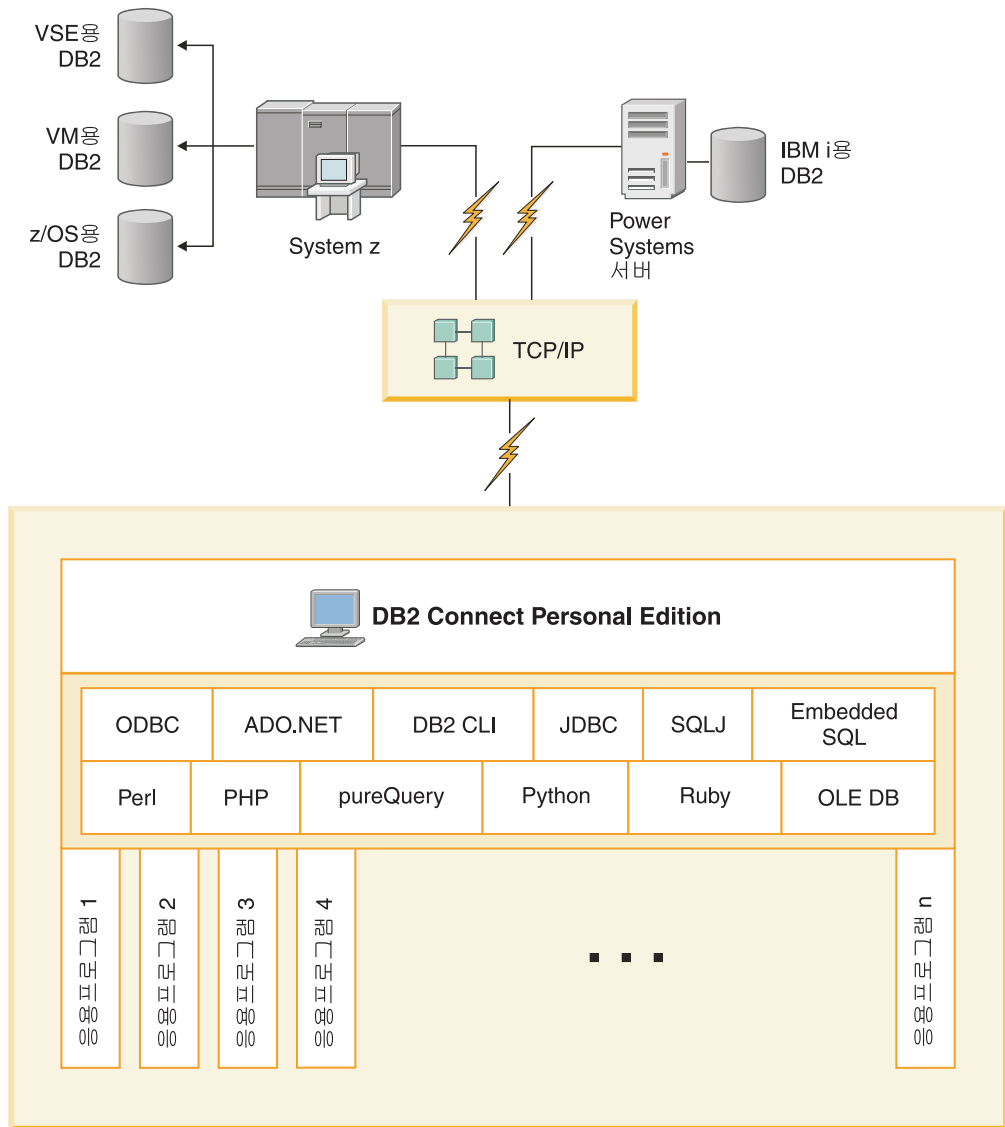


그림 3. DB2 Connect와 IBM 메인프레임 데이터베이스 서버 간의 직접 연결

주:

1. DB2 Connect Personal Edition 워크스테이션에 DB2를 설치할 필요가 없습니다. DB2 Connect Personal Edition 워크스테이션에서 완전한 관계형 데이터베이스 관리 시스템을 사용하려면 DB2를 구입하십시오.
2. 모든 IBM Data Server Client 기능을 DB2 Connect Personal Edition에 사용할 수 있습니다.
3. Sysplex가 사용 가능한 DB2 데이터베이스 서버에 대한 연결이 끊긴 경우, 클라이언트가 자동으로 연결을 재설정하려고 시도합니다.

DB2 Connect Personal Edition을 사용하여 System z 호스트 또는 IBM i DB2 데이터에 액세스

중간 서버가 없는 직접 연결은 매우 편리하고 유용한 구성입니다. 특히 IBM 메인프레임 데이터베이스 서버가 TCP/IP 연결성을 지원하는 상황에 적용됩니다. 이러한 상황에서 각 DB2 Connect 워크스테이션은 IBM 메인프레임 데이터베이스 서버와의 직접 연결을 설정합니다.

TCP/IP 연결에는 IBM 메인프레임 데이터베이스가 TCP/IP를 지원해야 합니다. 다음 버전은 원시 TCP/IP 연결을 지원합니다.

- DB2 버전 7.1 이상
- IBM i용 DB2 버전 5 릴리스 1 이상
- DB2 서버 버전 7 이상

IBM 메인프레임 데이터베이스 서버에 연결하려면 IBM Data Server Client에 추가할 수 있는 DB2 Connect 라이선스가 필요합니다.

18 페이지의 그림 4는 DB2 Connect Personal Edition이 설치된 IBM 메인프레임 데이터베이스 서버에 직접 연결된 워크스테이션을 표시합니다.

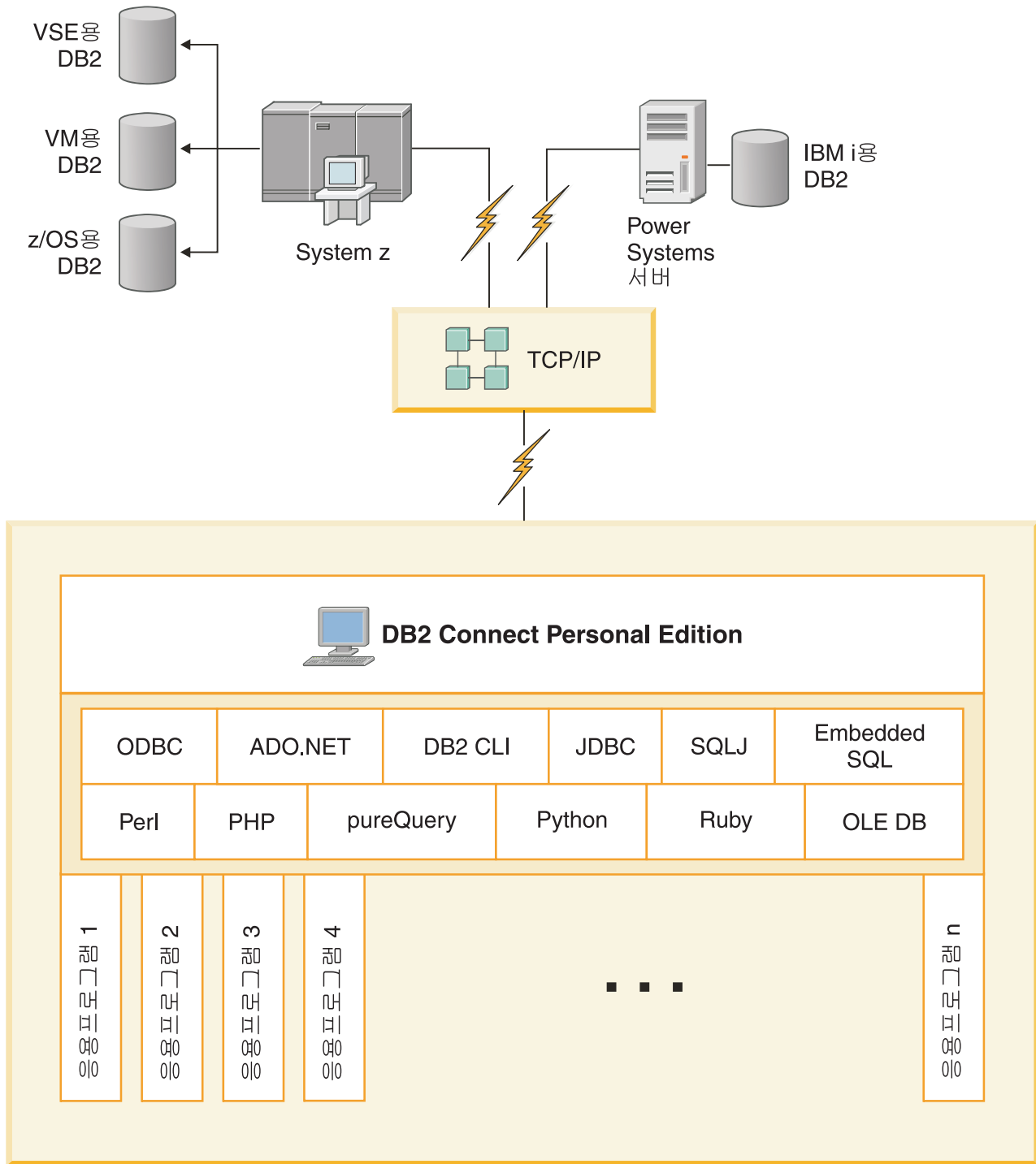


그림 4. DB2 Connect와 IBM 메인프레임 데이터베이스 서버 간의 직접적인 연결

DB2 Connect 서버 제품을 연결 서버로 사용

DB2 Connect 서버를 통해 여러 클라이언트가 IBM 메인프레임 데이터에 연결할 수 있으며, 엔터프라이즈 데이터에 대한 액세스를 설정하고 유지보수하는 데 필요한 노력을 크게 줄일 수 있습니다. 그림 5는 DB2 클라이언트가 DB2 Connect 서버 제품(예: DB2 Connect Enterprise Edition)을 통해 IBM 메인프레임 데이터베이스 서버에 간접적으로 연결하는 환경에 대한 IBM 솔루션을 나타냅니다.

주: 간접 연결은 Linux, UNIX 또는 Windows에서 실행 중인 DB2 클라이언트 또는 JCC 클라이언트에만 지원됩니다. 다른 클라이언트를 사용하여 DB2 Connect 서버 제품을 통해 IBM 메인프레임 데이터베이스 서버에 연결하면 SQL1334 오류가 발생합니다.

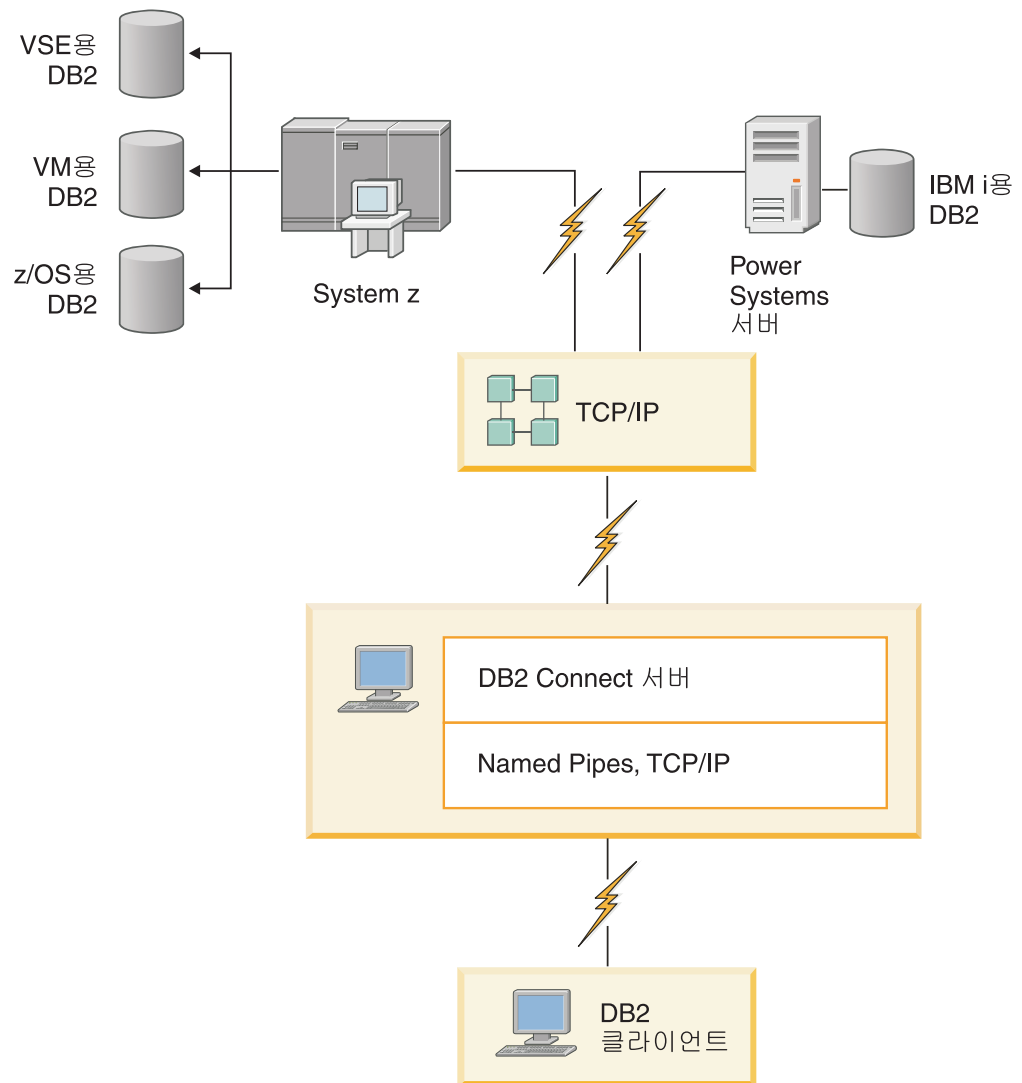


그림 5. DB2 Connect Enterprise Edition

DB2 Connect 서버에 대한 TCP/IP 연결이 끊어지면, 클라이언트가 자동으로 연결을 다시 설정하려고 시도합니다. 클라이언트는 먼저 원래 서버에 대한 연결을 다시 설정하려고 시도합니다. 연결이 재설정되지 않으면, 클라이언트가 대체 DB2 Connect 서버로 장애 복구를 수행합니다. (대체 서버는 서버 인스턴스에 지정되어 있고 해당 위치는 연결 중에 클라이언트에 리턴됩니다.) 대체 서버에 대한 연결이 다시 설정되지 않으면 클라이언트는 원래 서버에 대한 연결을 다시 설정하려고 시도합니다. 클라이언트는 연결이 설정되거나 시도 횟수가 시간종료될 때까지 교대로 원래 서버와 대체 서버에 대한 연결을 다시 설정하려고 계속 시도합니다.

DB2 Connect 및 웹 응용프로그램

웹 브라우저는 빠른 속도로, 온라인 카탈로그에서 인트라넷 응용프로그램에 이르기까지 모든 상황에서 표준 인터페이스가 되고 있습니다. 단순한 웹 응용프로그램의 경우에는 웹 서버 하나로 충분할 수 있습니다. 데이터베이스 액세스 및 트랜잭션 처리를 필요로 하는 많은 볼륨의 응용프로그램의 경우에는 IBM은 DB2 Connect를 사용하여 매우 많은 볼륨의 동시 트랜잭션을 웹을 통해 관리하는 솔루션을 제공합니다.

기존의 CGI 프로그램의 장점 및 한계

WWW(World Wide Web)에서의 e-business 응용프로그램은 일반적으로 CGI(Common Gateway Interface)를 사용하여 사용자가 백엔드 데이터베이스를 쿼리할 수 있도록 합니다. 또한 여러 회사에서 웹 응용프로그램을 내부적으로 사용하며 이들 또한 데이터베이스를 백그라운드로 가지고 있습니다.

사용자가 웹 페이지의 양식을 채우고 이 양식은 CGI를 통해 웹 서버상의 응용프로그램 또는 스크립트에 전달됩니다. 스크립트는 제공된 데이터베이스 API를 차례로 사용하여 호스트 데이터베이스에 SQL 쿼리를 제출합니다. 그런 다음, 동일한 스크립트에서 쿼리 결과가 포함된 웹(HTML) 페이지를 빌드하여 사용자의 웹 브라우저에서 이를 표시하도록 다시 전송합니다. 한 예로, 사용자가 특정 상품이나 서비스의 사용 가능성 또는 현재 가격을 쿼리할 수 있는 온라인 카탈로그를 들 수 있습니다.

CGI 응용프로그램은 디자인과 유지보수가 간단합니다. CGI 표준은 운영 체제 및 언어 둘다에 독립적이므로 거의 모든 컴퓨팅 플랫폼에서 사용 가능합니다. CGI 프로그램은 C++ 또는 스크립트 언어(예: Perl 또는 PHP)로 작성할 수 있습니다.

CGI가 웹 기반 응용프로그램의 이상적인 솔루션으로 보일 수도 있지만 중요한 결점이 있습니다. CGI에 대한 프로그래밍 환경은 다른 API처럼 복잡하지 않지만 확장성은 대규모 전자 상거래 조작에 문제가 될 수 있습니다. CGI 응용프로그램이 호출될 때마다 웹 서버에서 새 프로세스가 작성됩니다. 각 프로세스는 데이터베이스에 직접 연결하고 고유의 쿼리를 제출해야 합니다. 대용량의 트랜잭션 환경에서는 이러한 한계로 인해 심각한 성능 문제가 발생할 수 있습니다.

웹 서버와 함께 DB2 Connect를 사용하여 강력한 대용량의 e-commerce 응용프로그램을 작성할 수 있습니다. DB2 Connect에서는 웹 기본 응용프로그램 성능을 개선하는 여러 가지 솔루션을 제공합니다. 스토어드 프로시저를 사용하여 DB2 Connect 사용자는 데이터베이스로 전송되는 쿼리 수를 줄일 수 있습니다.

연결 풀링으로 데이터베이스에 대한 연결 및 연결 끊기의 빈도를 감소시킬 수 있습니다.

웹 서버 모듈 또는 플러그인으로 PHP 사용

CGI 프로그래밍에 PHP를 사용할 수 있다라도 일반적으로 웹 서버 모듈 또는 플러그인으로 사용됩니다. Apache와 같은 멀티프로세스 웹 서버에서 확장성 문제를 완화시키기 위해 PHP용 IBM DB2 드라이버를 사용할 수 있습니다. 멀티프로세스 웹 서버에서 프로세스의 풀은 서비스 웹 서버 요청에 다시 사용됩니다. 모든 웹 요청에 데이터베이스 연결 빌드 요구를 제거하려면 지속적 연결을 작성할 수 있습니다. 이 환경에서 지속적 연결은 단일 PHP 스크립트 범위를 벗어나 있을 수 있습니다. 후속 웹 요청에서 동일한 연결이 필요한 경우, 연결이 다시 사용됩니다.

DB2 Connect 및 IBM WebSphere

IBM WebSphere®는 PHP와 같은 기존의 스크립팅 도구에서 가능했던 것보다 더 완벽한 e-business 솔루션을 제공합니다. WebSphere Application Server는 PHP의 스크립트 가능성을 수행하고 Servlet, Active Server Pages 및 엔터프라이즈 JavaBeans™를 사용하여 웹을 통해 복잡한 최상의 서비스를 제공할 수 있도록 하며 Java™, TCP/IP, HTTP, HTTPS, HTML, DHTML, XML, MIME, SMTP, IIOP 및 X.509 등과 같은 웹 기반 기술을 지원합니다. WebSphere로 다음을 수행할 수 있습니다.

- 업계 표준을 준수함으로써 개발 속도를 높이고 상호 조작 가능성을 최대화합니다.
- 씨드 파티 도구 기술 및 응용프로그램 프레임워크를 플러그인 합니다.
- 웹 사이트 콘텐츠 성능 및 사용을 분석합니다.
- 보다 많은 사용자를 수용하고 처리량을 유지보수할 수 있도록 사이트를 쉽게 확장할 수 있습니다.
- 여러 주요 운영 환경(AIX®, HP-UX, Linux, Novell NetWare, z/OS, IBM i, Solaris 운영 체제, Microsoft Windows)에 전개합니다.
- Apache, IBM, Netscape 및 Microsoft의 웹 서버를 포함하여 기존의 웹 서버를 사용합니다.

WebSphere는 하나의 제품이 아니라 서로 다른 세 종류의 시장을 목표로 하는 세 가지 제품의 제품군입니다. WebSphere 솔루션의 핵심은 WebSphere Application Server입니다.

WebSphere Application Server는 세 가지 유형의 오브젝트에 대한 환경을 제공합니다. 첫 번째는 Java 서버 페이지로 Active Server Pages와 유사합니다. 두 번째 구성 요소는 Java Servlet으로 구성되고 세 번째는 엔터프라이즈 JavaBeans입니다. 엔터프라이즈 JavaBeans는 규모가 매우 크고 강력한 엔터프라이즈 클래스 응용프로그램을 전개하기 위한 새로운 표준입니다.

WebSphere 응용프로그램은 웹 서버 및 DB2와 동일한 플랫폼에 전개할 수 있습니다. z/OS용 DB2, DB2 서버, IBM i용 DB2의 경우에는 WebSphere를 DB2 Connect 서버 제품과 동일한 플랫폼에 전개할 수 있습니다.

Rational® Application Developer(RAD)뿐만 아니라 여러 개의 WebSphere 솔루션이 있습니다. 자세한 내용은 <http://www.ibm.com/software/webservers/appserv/was/>를 참조하십시오.

Java 응용프로그램 서버로서의 DB2 Connect

스크립트 언어와 관련된 많은 결점은 Java를 대신 사용함으로써 해결할 수 있습니다. IBM은 웹 트랜잭션의 모든 단계에서 Java를 사용할 수 있도록 하는 애플릿 및 응용 프로그램을 모두 제공합니다. IBM이 제공하는 솔루션은 기술을 합쳐서 사용할 수 있도록 하는데, 이는 DB2에서 Perl DBI 또는 Microsoft Active Server Pages와 같은 스크립트 솔루션을 사용하거나, IBM WebSphere와 같은 Java 응용프로그램 서버가 제공하는 더 강력한 구현을 사용하는 것을 의미합니다.

Java 프로그래머를 위한 두 가지 API(Application Programming Interfaces)가 있습니다. 첫 번째 JDBC는 Java Servlets, Java Server Pages(JSP) 및 Enterprise Java Beans(EJB)뿐만 아니라 Java를 사용하여 데이터 인식 Java 애플릿과 Java 응용프로그램을 개발하도록 지원됩니다. JDBC는 콜 레벨 또는 메소드 호출 API입니다. 다른 Java API는 SQLJ입니다. SQLJ는 Java 프로그램 내에서 SQL 인라인을 지정할 수 있는 기능을 제공합니다. DB2는 웹 트랜잭션의 클라이언트 또는 서버에서 두 API를 모두 사용할 수 있습니다.

클라이언트측에서는 애플릿, 데이터 인식 애플릿 및 응용프로그램이 지원됩니다. 데이터베이스 측에서는 Java 지원이 사용자 정의 함수(UDF) 및 스토어드 프로시저 같은 데이터베이스 오브젝트로 구성됩니다.

z/OS용 DB2, DB2 서버 및 IBM i용 DB2의 경우, Java 응용프로그램을 전개하는 서로 다른 두 가지 방법이 있습니다. TCP/IP의 DB2 Connect Personal Edition에서 제공하는 직접 연결을 사용하거나, IBM 메인프레임 데이터 서버에 연결 기능을 제공하는 DB2 Connect 서버 제품 전체를 이동하도록 선택할 수 있습니다.

두 가지 경우에 웹 사용자에게는 데이터베이스에 액세스하기 위한 별도의 소프트웨어는 필요하지 않고 표준 웹 브라우저만 필요합니다. DB2 Connect 서버 제품 및 산업

표준 웹 서버를 설치하기만 하면 됩니다. 웹 서버와 DB2 Connect가 실제 같은 머신에 없으면 웹 서버에 IBM Data Server Client를 설치해야 합니다.

z/OS용 DB2의 경우, 주요 구성요소는 중간 계층 서버에서 실행되는 DB2 Connect 서버 제품입니다. 이 구성요소는 z/OS용 DB2, DB2 서버 및 i용 DB2 서버에 연결하는 것 이외에 JDBC 서버 지원을 제공합니다. 즉, 클라이언트의 웹 브라우저 경우에는 어떠한 특수 소프트웨어도 필요하지 않습니다.

IBM은 Java 응용프로그램 및 애플릿을 개발하기 위한 광범위한 지원 및 도구를 제공합니다. DB2 Database Enterprise Developer Edition은 데이터베이스 응용프로그램 개발을 위해 테스트에 사용되는 DB2 제품 및 DB2 Connect뿐만 아니라 Rational Web Developer, IBM Data Studio, DB2 WebSphere Application Server를 제공합니다. NetBeans, Borland JBuilder 또는 Symantec Visual Cafe와 같은 써드 파티 도구도 IBM의 데이터베이스 솔루션과 함께 실행됩니다.

웹 서버에서 DB2 Connect

IBM은 모든 DB2 Connect 제품에 HTTP(웹) 서버를 제공합니다. DB2 Connect Enterprise Edition과 같은 DB2 Connect 서버 제품은 Apache 또는 Lotus Domino® Go Web Server에 컴퓨터 외부 지원을 제공하며, Microsoft Internet Information Server 또는 Netscape Enterprise Server와 같은 다른 웹 서버와도 작업할 수 있습니다.

System z, IBM Power Systems, VM 및 VSE 시스템에서 실행 중인 DB2 계열 데이터베이스로 작업하는 경우, 웹 서버에 DB2 Connect 서버 제품이 필요합니다. DB2 Connect 서버 제품은 웹 서버가 이러한 IBM 메인프레임 플랫폼에 액세스할 수 있도록 라이브러리 및 통신 인터페이스를 제공합니다. TCP/IP를 사용하여 System z, IBM Power Systems, VM 또는 VSE에서 실행 중인 웹 서버와 데이터베이스 간에 통신할 수 있습니다.

주: IBM 웹 솔루션은 동일한 CGI(Common Gateway Interface) 스크립트(예: PHP) 내에서 또는 CGI 스크립트의 동일한 트랜잭션 내에서 여러 데이터베이스로 작업할 수 있는 기능을 제공합니다.

스토어드 프로시저

클라이언트/서버 환경에서와 같이, 웹 응용프로그램에 대한 중요한 고려사항은 HTTP 서버와 백엔드 데이터베이스 사이에 발생하는 트래픽을 최소화하는 것입니다. 이러한 고려사항은 대부분의 e-business 응용프로그램에 있어서 가장 중심이 되는 대용량 트랜잭션 처리에 특히 중요합니다.

바람직한 접근 방법은 스토어드 프로시저에 캡슐화된 프로그래밍 및 비즈니스 로직으로 CGI 응용프로그램 프로그래밍을 결합하는 것입니다. Linux, UNIX 및 Windows용 DB2

데이터베이스 및 z/OS용 DB2, IBM i용 DB2 및 VSE용 DB2는 모두 동일한 매개변수 규칙을 공유하여 스토어드 프로시저를 호출합니다.

일반 웹 인터페이스 스크립트의 경우에는 웹 브라우저가 웹 서버에 양식을 제출하며, 웹 브라우저에서 웹 인터페이스 스크립트가 실행됩니다. 그러나, 각 개별 SQL문을 DB2 데이터베이스로 전송하는 대신, 스토어드 프로시저 실행 요청이 전송됩니다. 이 스토어드 프로시저는 달리 지정하지 않으면 개별적으로 실행되는 여러 SQL문을 캡슐화합니다. 스토어드 프로시저는 웹 인터페이스 스크립트와 백엔드 데이터베이스 사이에 앞뒤로 이동하는 메시지 수를 줄입니다.

스토어드 프로시저의 주요 이점은 HTTP 서버와 DB2 데이터베이스 백엔드 간의 네트워크 트래픽을 감소시킬 수 있다는 것입니다.

DB2 Connect 및 응용프로그램 서버(AS)

클라이언트 서버 응용프로그램의 증가로 인해 응용프로그램 설계자는 Windows와 같은 플랫폼에서 응용프로그램에 그래픽 사용자 인터페이스를 제공함으로써 유용성을 향상시키고 교육 비용을 줄일 수 있게 되었습니다. 동시에 다양한 운영 체제 및 하드웨어 플랫폼에서 강력한 데이터베이스 서버에 데이터베이스 관리 기능을 위임하는 융통성도 가능해졌습니다.

응용프로그램 로직이 클라이언트 워크스테이션으로 분산되는 클라이언트 서버 모델을 일반적으로 2티어 클라이언트 서버라고 합니다. 2티어 모델에서는 응용프로그램이 클라이언트 티어상에 배치되고 데이터베이스 서버는 서버 또는 백엔드 티어를 구현합니다. DB2 Connect는 데이터베이스 서버가 z/OS용 DB2, IBM i용 DB2 또는 DB2 서버인 2티어 클라이언트 서버 응용프로그램을 완벽하게 지원합니다.

클라이언트 서버 응용프로그램의 크기를 증가시키면 2티어 클라이언트 서버 모델에 중대한 한계점이 나타납니다. 수백 또는 수천 개의 클라이언트 워크스테이션에 대용량의 비즈니스 로직을 분산시키면 변경을 관리하기가 복잡하고 비용도 늘어납니다. 비즈니스 규칙을 변경하면 응용프로그램의 클라이언트 부분을 대체해야 합니다. 때로는 응용프로그램 롤아웃이 비즈니스 규칙이 일관성있게 적용되는지를 확인하는 시점에서 동시에 엔터프라이즈 내의 모든 클라이언트 워크스테이션에 있어야 합니다.

스케일에 따라 명백해지는 2티어 클라이언트 서버 모델의 또 다른 결점은 응용프로그램이 소비하는 자원의 양입니다. 2티어 클라이언트라고도 하는 FAT 클라이언트를 수백 또는 수천 개를 배치하면, 각 클라이언트 워크스테이션의 즉시 처리 능력 및 용량이 증가합니다. 더욱이, 데이터베이스 서버에 대한 요구는 각 클라이언트가 전용 데이터베이스 연결과 그러한 연결을 유지보수하는 데 연관된 자원을 요청할 때도 많이 증가합니다. 분산 비즈니스 로직의 2티어 클라이언트 서버 종속성은 스토어드 프로시저를 광범위하게 사용함으로써 어느 정도 줄일 수 있지만, 다른 결점은 모델을 변경하지 않고는 쉽게 해결되지 않습니다.

응용프로그램 서버 솔루션

증가된 2티어 클라이언트 서버 응용프로그램의 단가 및 복잡성으로 인해 대부분의 초대형 응용프로그램은 멀티 티어 클라이언트 서버로 전환되는 과정에 있게 됩니다. 멀티 티어 모델에서는 데이터베이스 티어의 역할이 변경되지 않고 남아 있습니다. 그러나, 클라이언트 티어에는 하나 이상의 미들 티어 즉, 일반적으로 하나의 티어가 추가되며, 이름은 3티어가 됩니다.

3티어 모델에서는 사용자 상호 작용에 대한 조절이 클라이언트에 위임되며 클라이언트에는 어떠한 비즈니스 로직도 포함되지 않습니다. 미들 티어는 하나 이상의 응용프로그램 서버로 구성됩니다. 응용프로그램 서버의 목표는 비즈니스 프로세스 및 비즈니스 규칙을 따르면서 강력하고도 효율적인 비용으로 구현하는 것입니다. 2티어 모델에서와 같이 비즈니스 규칙 구현에서도 성능을 개선하기 위해 스토어드 프로시저를 종종 사용합니다.

클라이언트 워크스테이션은 더 이상 많은 양의 응용프로그램 로직을 구현하지 않고 사용자 상호 작용만을 핸들하기 때문에, 클라이언트 티어에 대한 자원 요청도 상당히 감소합니다. 사실상, 3티어 모델에서의 클라이언트 티어를 종종 *thin 클라이언트*라고 합니다. 또한, 중앙 집중 응용프로그램 서버는 모든 클라이언트의 요청을 조절하기 때문에 모든 클라이언트 사이의 데이터베이스 연결과 같이 자원을 공유하는 기능을 가집니다. 결과적으로 데이터베이스 서버는 각 응용프로그램 사용자에게 대한 전용 연결을 더 이상 유지보수하지 않습니다.

오늘날 산업 부분에는 3티어 응용프로그램 서버에 대한 예들이 많이 있습니다. 거의 모든 엔터프라이즈 자원 플랜(ERP) 벤더는 SAP R/3 및 PeopleSoft V7 응용프로그램과 같은 3티어 모델을 사용하여 그 응용프로그램을 구현합니다. 기타 다른 예에는 Siebel 및 Vantive와 같은 엔터프라이즈 관계 관리 벤더가 포함됩니다.

응용프로그램 서버와 DB2 Connect

DB2 Connect 서버 제품은 멀티 티어 응용프로그램을 전개하기 위한 포괄적인 지원을 제공합니다. DB2 Connect가 제공하는 지원에는 DB2 계열 데이터베이스 서버와 상호작용하기 위한 전체 통신 인프라스트럭처 뿐만 아니라 응용프로그램 로직(ODBC, ADO.NET, DB2 CLI, Embedded SQL, JDBC, SQLJ, Perl, PHP 및 OLE DB)을 개발하는 데 사용할 수 있는 여러 API도 포함됩니다.

또한, DB2 Connect는 데이터베이스 티어가 여러 DB2 계열 데이터베이스 서버로 구성되는 구현도 지원합니다. 이를 통해 응용프로그램 서버는 단일 트랜잭션으로 여러 데이터베이스 서버에 상주하는 데이터를 갱신하는 트랜잭션을 구현할 수 있습니다.

DB2 Connect가 제공하는 2단계 커밋 프로토콜 지원은 이러한 분산 트랜잭션의 무결성을 보장합니다. 예를 들어, 응용프로그램은 동일한 트랜잭션에서 DB2 데이터베이스 및 Linux, UNIX 및 Windows용 DB2 데이터베이스에 있는 데

이터를 갱신할 수 있습니다. 분산 요청(DR) 지원이 설치되어 사용 가능한 경우에는 응용프로그램이 동일한 트랜잭션에서 Oracle 데이터베이스를 읽고 DB2 계열 데이터베이스를 갱신할 수 있습니다.

다음 다이어그램에서, 응용프로그램 서버와 백엔드 데이터베이스 서버 간의 연결 메커니즘 및 API는 DB2 Connect Enterprise Edition과 같은 DB2 Connect 서버 제품에서 제공합니다.

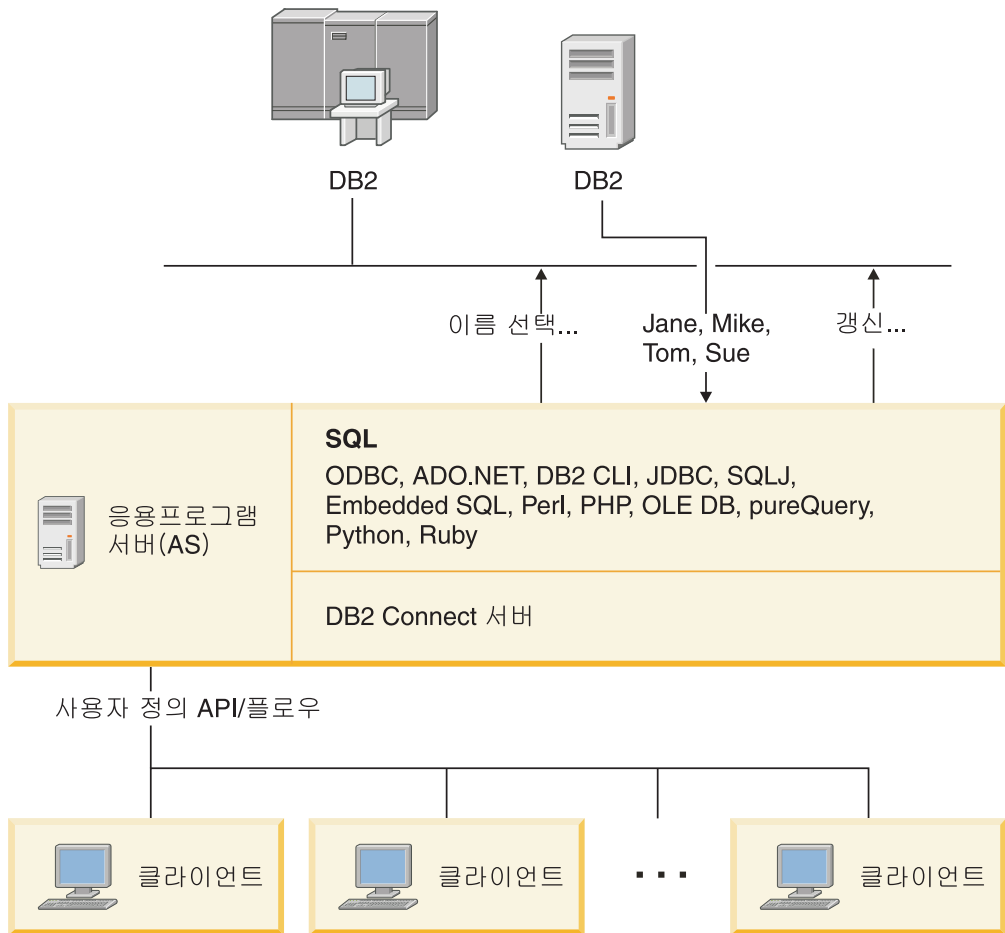


그림 6. 응용프로그램 서버에 대한 DB2 Connect 지원

연결 풀링과 같은 DB2 Connect의 고급 기능은 응용프로그램 자원 요구사항을 크게 줄여주고 응용프로그램 서버 구현을 간소화합니다.

DB2 Connect 및 응용프로그램 서버 구성

DB2 Connect 서버 제품은 응용프로그램 서버와 함께 사용하는 데 필요합니다. DB2 Connect Personal Edition은 지원되지 않으며 응용프로그램 서버에서 사용하도록 라이선스가 부여되지 않았습니다. 그리고, 응용프로그램 서버를 구현하는 고객은 DB2 Connect 사본에 제공된 조항 및 조건을 검토하여 획득해야 하는 사용자 라이선스 수를 파악해야 합니다.

응용프로그램 서버 환경에서 DB2 Connect를 전개하는 두 가지 방법이 있습니다. DB2 Connect 서버 제품을 다음 머신 중 하나에 설치할 수 있습니다.

- 응용프로그램 서버 머신
- 개별 통신 서버 머신

대부분의 경우, 응용프로그램 서버와 동일한 서버에 DB2 Connect 사본을 설치하는 방법이 바람직한 솔루션입니다. 응용프로그램 서버에 DB2 Connect를 설치하면 응용프로그램 서버가 구현 중일 수 있는 장애 복구 및 로드 밸런싱 스킴에 참여할 수 있습니다. 이렇게 설정하면 DB2 Connect를 개별 서버에 설치할 때 필요한 추가 네트워크 홉이 제거되므로 성능이 향상될 수 있습니다. 또한, 추가 서버를 설치 및 유지보수할 필요가 없기 때문에 관리가 간소화됩니다.

응용프로그램 서버가 실행 중인 운영 체제 또는 하드웨어에 대해 DB2 Connect 서버 제품을 사용할 수 없는 경우에는 DB2 Connect를 개별 서버에 설치하는 것이 좋습니다.

DB2 Connect 및 트랜잭션 처리 모니터

응용프로그램 서버는 많은 수의 사용자가 최소의 시스템 자원을 사용하여 응용프로그램을 실행할 수 있도록 합니다. 응용프로그램 서버는 응용프로그램 서버가 실행하는 응용프로그램으로부터 조정된 트랜잭션을 호출할 수 있도록 확장할 수 있습니다. 이 트랜잭션 조정을 일반적으로 트랜잭션 처리(TP) 모니터라고 합니다. TP 모니터는 응용프로그램 서버와 결합하여 작업합니다.

트랜잭션은 조직의 일일 조작을 실행한다는 점에서 일상적인 이벤트(일반적으로는 서비스에 대한 요청)와 같은 개념일 수 있습니다. 트랜잭션의 규칙적인 처리는 TP 모니터가 설계된 작업의 유형입니다.

트랜잭션 처리

모든 조직에는 어떻게 작동될 것인지에 대해 서술하는 규칙 및 프로시저가 있습니다. 이러한 규칙을 구현하는 사용자 응용프로그램을 *비즈니스 로직*이라고 합니다. 이 비즈니스 응용프로그램이 실행하는 트랜잭션을 종종 트랜잭션 처리 또는 온라인 트랜잭션 처리(OLTP)라고 합니다.

상용 OLTP의 주요 특성은 다음과 같습니다.

많은 사용자

많은 개인들이 비즈니스의 현재 상태에 영향을 주기 때문에 트랜잭션 처리는 일반적입니다.

반복적 컴퓨터와의 상호 작용 대부분이 반복해서 실행되는 동일한 프로세스인 경향이 있습니다. 예를 들어, 주문 입력이나 지불 처리가 매일 몇 번씩 사용됩니다.

단기 상호 작용

조직의 개인들이 트랜잭션 처리 시스템과 수행할 상호 작용의 지속기간이 대부분 짧습니다.

데이터 공유

데이터는 조직의 상태를 나타내기 때문에 데이터의 사본이 하나만 있을 수 있습니다.

데이터 무결성

데이터는 조직의 현재 상태를 나타내야 하며 내부적으로 일치해야 합니다. 예를 들어, 모든 주문은 고객 레코드와 연관되어야 합니다.

낮은 자원 사용/트랜잭션

트랜잭션 처리가 비즈니스 수행의 직접 비용을 나타내므로 시스템 자원 사용은 최소값이어야 합니다. DB2 Connect를 사용하면 Linux, UNIX 및 Windows에서 실행 중인 응용프로그램 서버의 제어 하에 있는 응용프로그램이 리모트 LAN 및 IBM 메인프레임 데이터베이스 서버에 대해 트랜잭션을 실행하고 TP 모니터가 이러한 트랜잭션을 조정하도록 할 수 있습니다.

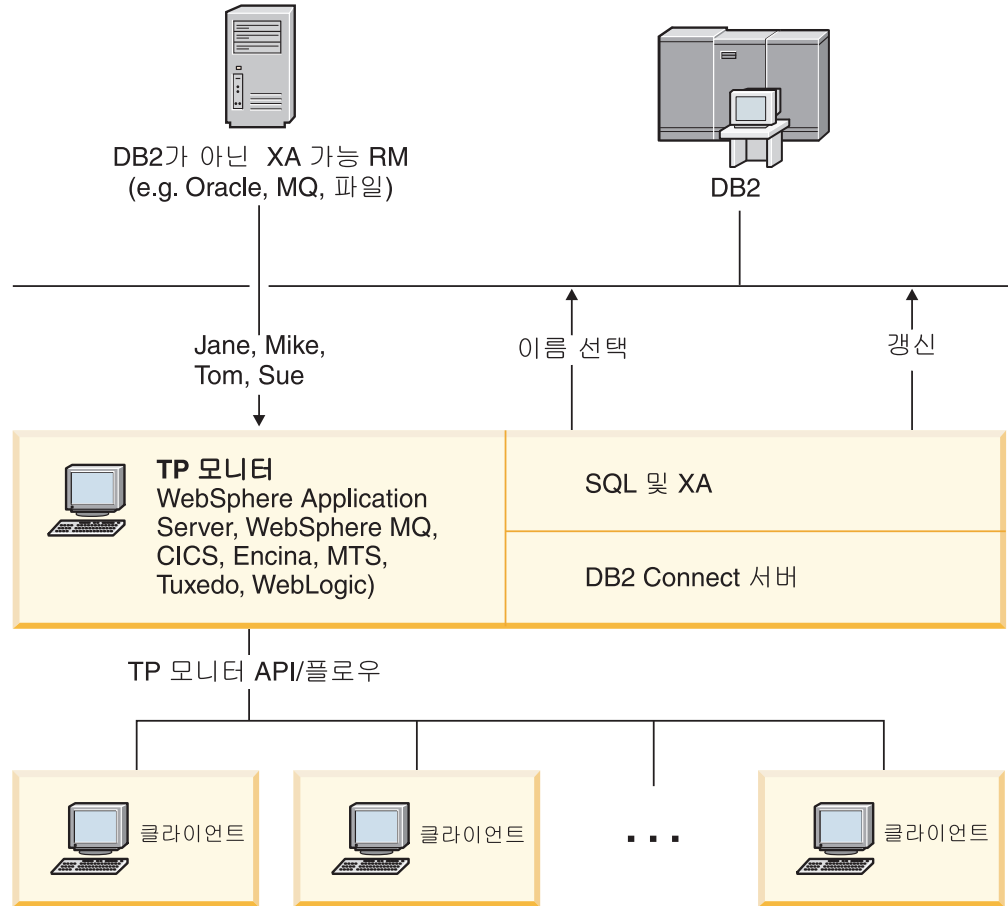


그림 7. TP 모니터에 대한 DB2 Connect 지원

그림 7에서 응용프로그램 서버와 백엔드 데이터베이스 서버 간의 연결 메커니즘 및 API는 DB2 Connect Enterprise Edition과 같은 DB2 Connect 서버 제품에서 제공합니다.

트랜잭션 처리 모니터의 예

오늘날 가장 많이 사용되는 TP 모니터는 다음과 같습니다.

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

리모트 IBM Power Systems, System z 및 LAN 데이터베이스 서버는 이러한 TP 모니터에서 조정된 트랜잭션 내에서 사용할 수 있습니다.

X/Open 분산 트랜잭션 처리(DTP) 모델

비즈니스 로직을 실행하는 응용프로그램은 단일 트랜잭션 내에서 여러 자원을 갱신해야 할 수 있습니다. 예를 들어, 한 계좌에서 다른 계좌로의 자금 이체를 구현하는 은행 응용프로그램에서는 한 데이터베이스("from" 계좌)에 차변을 기입하고 다른 데이터베이스("to" 계좌)에 예금할 수 있습니다.

서로 다른 벤더가 이 두 데이터베이스를 제공할 수도 있습니다. 예를 들어, 한 데이터베이스는 DB2이고 다른 데이터베이스는 Oracle 데이터베이스입니다. 모든 TP 모니터가 각 데이터베이스 벤더의 전매 트랜잭션 인터페이스를 구현하기 보다는, 응용프로그램이 액세스하는 자원과 TP 모니터 사이에 공통 트랜잭션 인터페이스가 정의됩니다. 이 인터페이스를 *XA 인터페이스*라고 합니다. XA 인터페이스를 사용하는 TP 모니터를 *XA 호환 트랜잭션 관리 프로그램(TM)*이라고 합니다. XA 인터페이스를 구현하는 갱신 가능한 자원을 *XA 호환 자원 관리자(RM)*라고 합니다.

위에 나열된 TP 모니터는 모두 XA 호환 TM입니다. DB2 Connect를 통해 액세스하는 경우, 리모트 호스트, IBM Power Systems 및 DB2 LAN 기반 데이터베이스는 XA 호환 RM입니다. 따라서, XA 호환 TM이 있는 TP 모니터는 트랜잭션을 실행하는 비즈니스 응용프로그램 내에서 호스트, IBM Power Systems 및 LAN 기반 DB2 데이터베이스를 사용할 수 있습니다.

제 2 부 DB2 Connect 참조

제 4 장 데이터베이스 디렉토리 갱신

DB2 Connect는 다음 디렉토리를 사용하여 데이터베이스 연결 정보를 관리합니다.

- **시스템 데이터베이스 디렉토리:** DB2 Connect가 액세스하는 모든 데이터베이스에 대한 이름, 노드 및 인증 정보가 있습니다.
- **노드 디렉토리:** DB2 Connect가 액세스하는 모든 IBM 메인프레임 데이터베이스 서버에 대한 네트워크 주소 및 통신 프로토콜 정보가 있습니다.
- **데이터베이스 연결 서비스(DCS) 디렉토리:** IBM 메인프레임 데이터베이스 서버 데이터베이스에 특정한 정보가 있습니다.

주:

1. 이러한 디렉토리를 갱신하기 전에 IBM 메인프레임 데이터베이스 서버 및 워크스테이션에서 통신을 구성해야 합니다.
2. 데이터베이스 디렉토리는 구성 지원 프로그램을 사용하여 갱신할 수 있습니다.

데이터베이스 디렉토리를 갱신하려면 다음을 수행하십시오.

1. 디렉토리 사용자 정의 워크시트를 사용하여 데이터베이스 디렉토리 정보를 수집하십시오.
2. 제어 센터의 『리모트 데이터베이스 서버 머신 정보를 사용하여 디렉토리 갱신』 주제를 참조하십시오.

시스템 데이터베이스 디렉토리 값

데이터베이스 관리 프로그램의 각 인스턴스에 시스템 데이터베이스 디렉토리가 존재하며 이 인스턴스에 카탈로그된 각 데이터베이스당 하나의 항목이 이 디렉토리에 포함됩니다. DB2 Connect 제품의 시스템 데이터베이스 디렉토리에는 각 데이터베이스의 이름, 별명, 노드 이름 및 인증 유형에 대한 정보가 있습니다.

다음 정보는 시스템 데이터베이스 디렉토리에서 지정할 수 있습니다.

데이터베이스 이름

DCS 디렉토리 매개변수 테이블의 값과 같습니다.

데이터베이스 별명

IBM 메인프레임 데이터베이스 서버의 별명. 이 이름은 데이터베이스를 액세스하는 다른 응용프로그램에서 사용될 수 있습니다. 디폴트값은 데이터베이스 이름으로 지정한 값이 사용됩니다.

형식: 1-8 사이의 단일 바이트의 영숫자 문자이며, 숫자 기호(#), at 기호(@), 달러 기호(\$) 및 밑줄(_)이 포함됩니다. 이 형식은 밑줄 문자나 숫자로 시작할 수 없습니다.

노드 이름

노드 디렉토리 매개변수 테이블의 값과 같습니다.

인증 DB2 Connect 서버에서 시작되는 연결에 대해 사용자 이름 및 암호의 유효성이 확인되는 위치를 지정합니다. 유효한 옵션은 SERVER, SERVER_ENCRYPT, CLIENT, KERBEROS, SERVER_ENCRYPT_AES 및 DATA_ENCRYPT입니다. 시스템 데이터베이스 디렉토리의 GSSPLUGIN 인증 유형에 대해서는 지원되지 않습니다.

노드 디렉토리 값

다음 정보는 노드 디렉토리에서 지정할 수 있습니다.

노드 이름

리모트 데이터베이스가 위치하는 IBM 메인프레임 데이터베이스 서버 시스템의 별칭. 이 이름은 사용자가 정의합니다. 노드 디렉토리 매개변수 테이블과 시스템 데이터베이스 디렉토리 매개변수 테이블 모두에 동일한 노드 이름을 사용하십시오.

형식: 1-8 사이의 단일 바이트의 영숫자 문자이며, 숫자 기호(#), at 기호(@), 달러 기호(\$) 및 밑줄(_)이 포함됩니다. 이 형식은 밑줄 문자나 숫자로 시작할 수 없습니다.

프로토콜

TCP/IP이어야 합니다.

보안 유형

보안 유형 확인이 수행됩니다. TCP/IP 노드에서, SECURITY SOCKS는 노드가 SOCKS 작동 가능하기 위해 지정하는 옵션이며, 이런 경우 SOCKS_NS 및 SOCKS_SERVER 환경 변수는 필수적이고 SOCKS를 작동 가능하기 위해 설정되어야 합니다.

TCP/IP 리모트 호스트 이름 또는 IP 주소

TCP/IP 노드를 정의할 때 리모트 TCP/IP 호스트 이름이나 리모트 TCP/IP 주소. 호스트 이름이 지정되면, 도메인 이름 서버(DNS) 찾아보기나 로컬 TCP/IP 호스트 항목을 통해 DB2 Connect 워크스테이션에서 이름이 해결되어야 합니다.

z/OS용 DB2 리모트 호스트의 경우, 호스트 이름은 분산 데이터 기능(DDF)이 시작될 때 DSNL004I 메시지(DOMAIN=hostname)에 표시됩니다. -DISplay DDF 명령도 사용할 수 있습니다.

z/OS 데이터 공유 그룹에 액세스하는 경우, 도메인 이름을 DB2 그룹 동적 VIPA 주소에 맵핑해야 합니다. 이 주소는 가장 적게 로드된 DB2 구성원으로 라우트됩니다. 특정 구성원에 액세스하려면 특정 DB2 구성원 동적 VIPA 주소를 사용하고 sysplex 라우팅을 해제하십시오. 각 구성원 DSNL004I 메시지는 구성원 특정 도메인 이름을 표시합니다.

TCP/IP 서비스 이름 또는 포트 번호

TCP/IP 노드를 정의할 때 리모트 TCP/IP 서비스 이름이나 포트 번호. 이것은 리모트 호스트에서 TCP/IP로 정의되어야 합니다. 포트 번호 446은 DRDA의 디폴트 포트 번호로 등록되어 있습니다.

z/OS용 DB2 리모트 호스트의 경우, 포트 번호는 부트 스트랩 데이터 세트 (BSDS)에 정의되고, 분산 데이터 기능(DDF)이 시작될 때 DSNL004I 메시지 (TCPPOPT=portnumber)에도 제공됩니다. -DISplay DDF 명령도 사용할 수 있습니다.

z/OS 데이터 공유 그룹에 액세스하는 경우, 도메인 이름을 DB2 그룹 동적 VIPA 주소에 맵핑해야 합니다. 이 주소는 가장 적게 로드된 DB2 구성원으로 라우트됩니다. 특정 구성원에 액세스하려면 특정 DB2 구성원 동적 VIPA 주소를 사용하고 sysplex 라우팅을 해제하십시오. 각 구성원 DSNL004I 메시지는 구성원 특정 도메인 이름을 표시합니다.

주: TCP/IP 연결을 통해 2단계 커밋 재동기화 조작에 사용되는 두 번째 포트는 서버에서 지정할 수 있습니다. 예를 들어, z/OS용 DB2 부트스트랩 데이터 세트는 z/OS용 DB2과의 인바운드 연결을 위한 재동기화에만 사용할 포트 번호(RESPORT)를 지정합니다. 여기서는 서비스 이름을 정의할 필요가 없습니다.

DCS 디렉토리 값

DCS 디렉토리에 다음 정보를 지정할 수 있습니다.

데이터베이스 이름

IBM 메인프레임 데이터베이스 서버의 사용자 정의 별칭. DCS 디렉토리 매개 변수 테이블과 시스템 데이터베이스 디렉토리 매개변수 테이블 모두에 동일한 데이터베이스 이름을 사용하십시오.

형식: 1-8 사이의 단일 바이트의 영숫자 문자이며, 숫자 기호(#), at 기호(@), 달러 기호(\$) 및 밑줄(_)이 포함됩니다. 이 형식은 밑줄 문자나 숫자로 시작할 수 없습니다.

목표 데이터베이스 이름

다음과 같이 IBM 메인프레임 데이터베이스 서버 시스템에 있는 데이터베이스

System z

LOCATION NAME으로 식별되는 z/OS용 DB2 서브시스템 또는 z/OS 서버에 정의된 별명 LOCATION 이름 중 하나입니다.

LOCATION NAME은 TSO에 로그인해서, 사용 가능한 쿼리 도구 중 하나를 사용하여 다음의 SQL 쿼리를 발행함으로써 결정될 수 있습니다.

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

여러 개의 LOCATION NAME은 부트 스트랩 데이터 세트(BSDS)에 정의되며 분산 데이터 기능(DDF)이 시작될 때 작성되는 DSNL004I 메시지(LOCATION=location)에도 정의됩니다. -DISplay DDF 명령도 사용할 수 있습니다.

z/OS 데이터 공유 그룹에 액세스하는 경우, 도메인 이름을 DB2 그룹 동적 VIPA 주소에 매핑해야 합니다. 이 주소는 가장 적게 로드된 DB2 구성원으로 라우트됩니다. 특정 구성원에 액세스하려면 특정 DB2 구성원 동적 VIPA 주소를 사용하고 sysplex 라우팅을 해제하십시오. 각 구성원 DSNL004I 메시지는 구성원 특정 도메인 이름을 표시합니다.

VSE 또는 VM

데이터베이스 이름(DBNAME)

IBM Power Systems

관계형 데이터베이스 이름(RDBNAME)

기타 Windows, Linux 및 UNIX 운영 체제의 경우, 데이터베이스 별명은 데이터베이스 디렉토리에 있습니다.

매개변수 문자열

디폴트값을 변경하려면, 다음 순서로 아래 매개변수의 일부나 전부를 지정하십시오.

맵 파일

디폴트 SQLCODE 매핑을 겹쳐쓰는 SQLCODE 매핑 파일 이름. SQLCODE 매핑을 작동 중지하려면, **NOMAP**을 지정하십시오.

주: 쿼리 요청을 처리할 때, DRDA 서버는 결과 세트를 나타내는 행 세트 형식으로 데이터를 리턴합니다. 각 행과 함께 일반적으로 0이거나 양수(예를 들면 +12 또는 +802)인 SQLCA가 리턴됩니다. DB2 Connect 서버에서 사용자 정의 매핑 과일을 사용하는 경우, 양수 sqlcode가 사용자 정의 매핑 파일에 들어 있고 매핑을 사용자 정의했다면(예를 들어, 다른

sqlcode 에 맵핑되거나 토큰 맵핑을 사용자 정의한 경우), 이러한 sqlcode가 맵핑되지 않습니다.

다음은 중요 사항입니다.

1. 양수의 sqlcode는 경고를 나타내고 음수의 sqlcode는 오류를 나타냅니다. 모든 음수의 sqlcode는 어떤 맵핑 파일이 사용되었는지에 상관없이 항상 모든 상황에 맵핑됩니다. 사용자 정의 맵핑 파일에 들어 있고 변경되지 않고 자체로 맵핑된 모든 양수의 sqlcode는 항상 맵핑됩니다. 또한, DB2 Connect 서버에서 사용자 정의 맵핑 파일에 들어 있지 않은 양수 sqlcode도 항상 맵핑됩니다.
2. 디폴트 맵핑 파일을 사용하거나 호스트 데이터베이스에 직접 연결하는 경우, sqlcode 맵핑은 모든 sqlcode에 대하여 이루어집니다.

,D 이것은 두 번째 위치 매개변수입니다. 이 매개변수를 지정하지 않는 경우, 다음 SQLCODES 중 하나가 리턴되면 응용프로그램이 IBM 메인프레임 데이터베이스 서버 데이터베이스로부터 연결이 끊깁니다.

```
SQL30000N  
SQL30040N  
SQL30050N  
SQL30051N  
SQL30053N  
SQL30060N  
SQL30070N  
SQL30071N  
SQL30072N  
SQL30073N  
SQL30074N  
SQL30090N
```

연결 끊기 매개변수 **,D**가 지정되지 않으면, 다음 SQLCODE가 리턴되는 경우에만 연결이 해제됩니다.

```
SQL30020N  
SQL30021N  
SQL30041N  
SQL30061N  
SQL30081N
```

메시지 참조서에서 이들 코드에 대한 자세한 내용을 참조하십시오.

주: DB2 Connect가 오류로 인해 연결이 해제되면, 자동으로 롤백이 됩니다.

„INTERRUPT_ENABLED

이것은 세 번째 위치 매개변수입니다. 마지막 서버가 인터럽트를 지원하지 않는 경우, INTERRUPT_ENABLED만 적용됩니다. 서버가 DRDA 인터럽트 플로우를 지원하면, DB2 Connect가 인터럽트 요청을 서버로 전달합니다.

INTERRUPT_ENABLED가 DB2 Connect 워크스테이션의 DCS 디렉토리에 구성되어 있고, 클라이언트 응용프로그램이 IBM 메인프레임 데이터베이스 서버에 연결되어 있는 동안 인터럽트를 발행하면, DB2 Connect가 연결을 삭제(drop)하고 작업 단위(UOW)를 롤백함으로써 인터럽트를 수행합니다. 이 인터럽트 작동은 AIX 및 Windows에서 지원됩니다.

응용프로그램은 서버와의 연결이 종료됨을 나타내는 sqlcode(-30081)를 수신합니다. 그런 다음, 응용프로그램이 추가 데이터베이스 요청을 처리하기 위해 IBM 메인프레임 데이터베이스 서버와 새로 연결해야 합니다. AIX V5.2 이상 및 Windows 이외의 다른 플랫폼의 경우, 자동 연결 끊기 옵션을 사용하는 응용프로그램이 인터럽트 요청을 수신하면 DB2 Connect가 이 옵션을 지원하지 않습니다.

주: 이런 지원은 모든 플랫폼의 TCP/IP 연결에 대해 작동합니다. 클라이언트가 소켓을 제거할 수 있지만, 서버 구현에 따라 미해결 수신이 있거나 없을 수 있습니다. z/OS용 DB2는 비동기 소켓 호출을 사용하여 처리 중인 긴 SQL문의 연결 손실을 감지하고 롤백할 수 있습니다.

„„„SYSPLEX

6번째 위치 매개변수인 이 매개변수는 특정 데이터베이스에 대해 DB2 Connect SYSPLEX 지원을 명시적으로 사용 가능하게 하는 데 사용됩니다.

„„„„LOCALDATE=<value>

7번째 위치 매개변수인 이 매개변수는 DB2 Connect 날짜 형식 지원을 사용 가능하게 하는 데 사용됩니다. 이는 다음과 같이 <value>의 날짜 마스크를 사용하여 구현됩니다.

다음 CLP(명령행 처리기) 명령문을 실행한다고 가정해 보십시오.

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

데이터베이스 별명 *newyork1*은 날짜 마스크가 지정되지 않았으므로 날짜 변형 없이 호스트 데이터베이스를 액세스하기 위해 사용됩니다.

그러나 이제는 새로운 날짜 형식화를 지원하므로 다음 CLP 명령을 사용할 수 있습니다. 이 경우, CLP가 사용되고 매개변수 문자열 자체는 큰 따옴표를 사용하여 지정하므로, 두 쌍의 큰 따옴표 안에 LOCALDATE 값을 지정해야 합니다. LOCALDATE 스펙에서 큰따옴표가 제거되지 않도록 운영 체제의 Esc 문자 "₩"(역슬래시) 사용에 유의하십시오.

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

데이터베이스 별명 *newyork2*는 같은 호스트 데이터베이스에 액세스하지만, 그 외에도 날짜 형식 마스크가 지정되어 있습니다. 이 예는 날짜 형식 마스크가 LOCALDATE 키워드를 사용하여 지정되며 DCS 디렉토리 항목의 PARMs 필드에서 7번째 위치 매개변수임을 보여줍니다.

날짜 마스크가 유효하려면 아래의 조건이 모두 충족되어야 합니다.

1. Y, M, D 중 최대 한 개의 시퀀스가 있습니다. 여기서 Y는 연도 자리, M은 월 그리고 D는 날짜를 나타내는 자리입니다.
2. Y의 최대수는 4입니다.
3. M의 최대수는 2입니다.
4. D의 최대수는 2입니다.

예를 들어, 다음은 모두 유효한 날짜 마스크입니다.

```
"YYyyMmDd" - Y, M, D 자리는 대소문자를 구별하지 않습니다.
"MM+DD+YYYY" - 마스크가 10바이트보다 길고
                마스크에 Y, M, D 이외의 문자를
                사용해도 좋습니다.
"abcYY+MM" - D 시퀀스는 사용하지 않아도 좋습니다.
```

다음은 모두 무효 날짜 마스크입니다.

```
"YYYYyMMDD" - Y가 5개이므로 잘못된
"YYYYMDDM" - M 시퀀스가 두 번 사용되었으므로 잘못된
```

날짜 형식 마스크가 잘못되었더라도 오류는 표시되지 않고 무시됩니다. 날짜 마스크가 유효하다고 해서 이 마스크가 반드시

사용되는 것은 아닙니다. 유효한 낱자 마스크에 기초를 둔 낱자 형식 변형은 다음 조건이 모두 충족되는 경우에만 수행됩니다.

1. SQL 오류가 없습니다.
2. 출력은 ISO같은(ISO와 JIS) 형식으로 된 낱자 값입니다.
3. 출력 데이터 영역의 길이는 최소한 10바이트입니다. 이것은 낱자 형식 변형이 수행되지 않더라도 낱자 값이 출력 데이터 영역에 저장되도록 하기 위한 최소 출력 데이터 영역 크기입니다. 이 요구 사항은 낱자 형식 마스크가 10바이트 보다 짧은 경우에도 적용됩니다.
4. DCS 디렉토리 항목에 유효한 낱자 형식 마스크가 지정되며 이 마스크는 출력 데이터 영역에 들어갑니다.

,,,,,,,BIDI=<ccsid>

8번째 위치 매개변수인 이 매개변수는 디폴트 서버 데이터베이스 BiDi CCSID를 대체하는 양방향(Bidi) CCSID를 지정하는데 사용됩니다. 예를 들면, 다음과 같습니다.

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

여기서, xyz는 CCSID를 대신한 것입니다.

디렉토리 사용자 정의 워크시트

디렉토리 사용자 정의 워크시트는 수집해야 할 정보를 보여줍니다. 워크시트 사본을 작성하여 시스템 값을 입력해 두면 편리합니다.

노드 디렉토리 매개변수

표 1. 노드 디렉토리 매개변수

매개변수	예	사용자 값
노드 이름	DB2NODE	
리모트 호스트 이름(TCP/IP 노드)	ZOSHOST	
서버(TCP/IP 서비스 이름 또는 포트 번호)	db2inst1c(또는 446)	

주:

1. DRDA의 디폴트 TCP/IP 포트 번호는 446입니다.
2. IBM 메인프레임 데이터베이스 서버가 SECURITY SOCKS를 지원하는지 여부를 잘 모르는 경우에는 TCP/IP 노드에 대해 SECURITY를 지정하지 마십시오.

DCS 디렉토리 매개변수

표 2. DCS 디렉토리 매개변수

매개변수	예	사용자 값
데이터베이스 이름	DB2DB	
목표 데이터베이스 이름	NEW_YORK3	
응용프로그램 리퀘스터		
매개변수 문자열	".....LOCALDATE=#"#"#YYMMDD#"#"#"	

시스템 데이터베이스 디렉토리 매개변수

표 3. 시스템 데이터베이스 디렉토리 매개변수

매개변수	예	사용자 값
데이터베이스 이름	DB2DB	
데이터베이스 별명	NYC3	
노드 이름	DB2NODE	
인증	SERVER	

동일한 데이터베이스에 대해서 여러 항목 정의

각 데이터베이스용으로, 세 개의 디렉토리(노드 디렉토리, DCS 디렉토리, 시스템 데이터베이스 디렉토리)에서 적어도 한 항목은 정의해야 합니다. 어떤 경우에는 데이터베이스용으로 둘 이상의 항목을 정의하려는 경우도 있습니다.

예를 들어, IBM 메인프레임 데이터베이스 서버로부터 포트된 응용프로그램의 SQLCODE 매핑을 해제하려고 할 수 있지만 클라이언트/서버 환경용으로 개발된 응용프로그램에 대해 디폴트 매핑을 그대로 사용할 수 있습니다. 이는 다음과 같이 실행됩니다.

- 노드 디렉토리에서 한 항목을 정의합니다.
- DCS 디렉토리에서 다른 데이터베이스 이름으로 두 항목을 정의합니다. 한 항목에는 매개변수 문자열에 NOMAP를 지정하십시오.
- 시스템 데이터베이스 디렉토리에서 다른 데이터베이스 별명으로 두 항목을 정의하고, DCS 디렉토리에서 지정한 두 데이터베이스 이름으로 정의합니다.

두 별명은 모두 동일한 데이터베이스를 액세스하며 하나는 SQLCODE 매핑이 있고 다른 하나는 SQLCODE 매핑이 없습니다.

BiDi 데이터 처리

다음 절은 z/OS 서버에만 적용됩니다. IBM i용 DB2 서버에서는 이미 전체 BiDi 지원이 이미 제공되어 있으므로 이 기능을 사용해서는 안 됩니다.

서로 다른 플랫폼에서 BiDi 데이터를 정확히 처리하려면, 다음의 BiDi 속성이 필요합니다.

- 숫자 모양(ARABIC 대 HINDI)
- 방향(RIGHT-TO-LEFT 대 LEFT-TO-RIGHT)
- 형태(SHAPED 대 UNSHAPED)
- 대칭 스와핑(YES 또는 NO)
- 텍스트 유형(LOGICAL 대 VISUAL)

서로 다른 플랫폼에서 디폴트값이 같지 않으므로 한 플랫폼에서 다른 플랫폼으로 DB2 데이터를 전송할 때 문제가 발생합니다. 예를 들어, Windows 플랫폼은 LOGICAL UNSHAPED 데이터를 사용하지만, z/OS 데이터는 보통 SHAPED VISUAL 형식입니다. 그러므로, BiDi 속성에 대한 지원이 없으면 z/OS용 DB2에서 Windows의 DB2 Connect로 전송된 데이터가 올바르게 표시되지 않습니다.

DB2 Connect와 서버의 데이터베이스 사이에 데이터가 교환되는 경우, 이것은 주로 수신 데이터에 대한 변환을 수행하는 수신자입니다. 동일한 규칙이 일반 코드 페이지 변환에 추가되어 BiDi 레이아웃 변환에도 적용됩니다. 그러나 BiDi 특정 CCSID 또는 BiDi 레이아웃 변환을 지원하는 호스트 DB2 제품은 현재 없습니다. 따라서, 서버 데이터베이스에서 수신되는 데이터 외에 서버 데이터베이스로 전송되는 데이터에 대해 BiDi 레이아웃 변환을 수행할 수 있도록 추가 기능이 DB2 Connect에 포함되어 있습니다.

DB2 Connect가 서버 데이터베이스로 나가는 데이터에 대해 BiDi 레이아웃 변환을 수행하려면 서버 데이터베이스의 BiDi CCSID를 겹쳐써야 합니다. 이것은 서버 데이터베이스에 대한 DCS 데이터베이스 디렉토리 항목의 PARMS 필드에서 BIDI 매개변수를 사용하여 수행됩니다.

이 기능의 사용은 예제에 가장 잘 표시되어 있습니다.

CCSID 62213(BiDi 문자열 유형 5)을 실행하는 히브리어 IBM Data Server Client가 있고 CCSID 424(BiDi 문자열 유형 4)를 실행하는 DB2 호스트 데이터베이스에 액세스하려는 경우를 생각해 보십시오. 그러나, DB2 호스트 데이터베이스에 포함된 데이터가 대신 CCSID 62245(BiDi 문자열 유형 10)에 기초한다는 것을 알고 있습니다.

이 경우에는 두 가지 문제가 발생합니다. 첫 번째는 DB2 호스트 데이터베이스가 CCSID 424 및 62245를 갖는 BiDi 문자열 유형 간의 차이를 모른다는 것입니다. 두 번째 문제점은 DB2 호스트 데이터베이스가 IBM Data Server Client CCSID 62213을 인식하지 못한다는 것입니다. 이는 CCSID 62213과 동일한 코드 페이지에 기반하고 있는 CCSID 62209(BiDi 문자열 유형 10)만을 지원합니다.

DB2 호스트 데이터베이스로 전송된 데이터가 시작하려는 BiDi 문자열 유형 6 형식인지 확인하고, 또한 DB2 Connect가 DB2 호스트 데이터베이스로부터 수신하는 데이터

에 대해 BiDi 레이아웃 변환을 수행해야 함을 알려야 합니다. DB2 호스트 데이터베이스에 대해 다음 카탈로그화를 사용할 수 있습니다.

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

이는 DB2 호스트 데이터베이스 CCSID 424를 62245로 겹쳐쓰도록 DB2 Connect에 지시합니다. 이 대체 작업에는 다음의 처리가 포함됩니다.

1. DB2 Connect가 CCSID 62209(BiDi 문자열 유형 10)를 사용하여 DB2 호스트 데이터베이스에 연결합니다.
2. DB2 Connect는 DB2 호스트 데이터베이스로 전송할 데이터에 대해 CCSID 62213(BiDi 문자열 유형 5)에서 CCSID 62209(BiDi 문자열 유형 10)로 BiDi 레이아웃 변환을 수행합니다.
3. DB2 Connect는 DB2 호스트 데이터베이스에서 수신하는 데이터에 대해 CCSID 62245(BiDi 문자열 유형 10)에서 CCSID 62213(BiDi 문자열 유형 5)으로 BiDi 레이아웃 변환을 수행합니다.

주:

1. BIDI 매개변수가 효력을 갖게 하려면 환경 변수나 레지스트리 값 **DB2BIDI**가 YES로 설정되어야 합니다. **DB2BIDI**는 DCS 데이터베이스 디렉토리 항목이 카탈로그화되어 있는 DB2 Connect 워크스테이션에서 설정되어야 합니다. 클라이언트에서 DB2 Connect 서버에 대해 리모트 실행 중인 응용프로그램의 경우 **DB2BIDI** 변수는 클라이언트에서도 설정되어야 합니다.
2. CCSID를 겹쳐쓸 필요가 없는 경우에도 DB2 Connect가 DB2 호스트 데이터베이스로 전송할 데이터에 대해 레이아웃 변환을 수행하도록 하려면 여전히 DCS 데이터베이스 디렉토리 PARMS 필드에 BIDI 매개변수를 추가해야 합니다. 이 경우, 제공해야 하는 CCSID는 디폴트 DB2 호스트 데이터베이스 CCSID입니다.
3. 양방향 CCSID를 사용할 경우 SQL 쿼리 자체가 수정되어 DB2 서버에서 인식하지 못하는 경우도 있습니다. 특히, 서로 다른 문자열 유형을 사용할 수 있는 경우에는 IMPLICIT CONTEXTUAL과 IMPLICIT RIGHT-TO-LEFT CCSID를 사용하지 말아야 합니다. CONTEXTUAL CCSID는 SQL 쿼리에 따옴표로 묶인 문자열이 들어 있는 경우 예상하지 못한 결과를 초래할 수 있습니다. SQL문에서는 따옴표가 있는 문자열을 사용하지 말고 가능하다면 호스트 변수를 사용하십시오.

특정 양방향 CCSID가 다음 권장사항을 수행하여 수정할 수 없는 문제가 발생한 경우, 환경 변수나 레지스트리 값 **DB2BIDI**를 NO로 설정해야 합니다.

매개변수 문자열 권장 스펙

다음은 DCS 매개변수의 예입니다. (각 행은 매개변수 세트입니다.)

```

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

```

또는 매개변수 문자열을 지정하지 않고 디폴트값을 사용할 수 있습니다.

주: UNIX 시스템의 운영 체제 명령행에서 CLP를 사용하는 경우, 매개변수 문자열에 LOCALDATE 마스크를 지정할 때 두 쌍의 큰 따옴표를 지정해야 하기 때문에 운영 체제의 Escape 문자 "\#" (백슬래시)를 사용해야 합니다. 예를 들면, 다음과 같습니다.

```
db2 catalog dcs db x as y parms \#,,,,,,LOCALDATE=\#\#YYMMDD\#\#\#
```

이 결과 다음의 DCS 디렉토리 항목이 생깁니다.

DCS 1 항목:

로컬 데이터베이스 이름	= X
목표 데이터베이스 이름	= Y
응용프로그램 리퀘스터 이름	=
DCS 매개변수	= ,,,,,,LOCALDATE="YYMMDD"
주석	=
DCS 디렉토리 릴리스 레벨	= 0x0100

제 5 장 DB2 Connect 보안

사용자는 액세스하려는 데이터가 있는 데이터베이스 및 DB2 Connect에 로컬 또는 리모트 모두에서 액세스할 수 있기 때문에 DB2 Connect를 사용할 때 사용자 인증은 중요합니다. 트러스트된 연결 및 Kerberos 지원은 호스트 머신의 보안 고려사항과 함께 나와 있습니다.

DB2 Connect를 통한 트러스트된 연결

일부 DB2 데이터베이스 서버는 트러스트된 컨텍스트를 지원합니다. 트러스트된 컨텍스트를 사용하면 클라이언트 응용프로그램이 트러스트된 연결을 작성하도록 허용될 조건을 데이터베이스 관리자가 정의할 수 있습니다. 트러스트된 연결에서는 일반 연결로는 할 수 없는 작업을 수행할 수 있습니다.

트러스트된 연결 유형에는 내재적 및 명시적 두 가지가 있습니다. 연결 작성시 명시적 트러스트된 연결, 내재된 트러스트된 연결 또는 일반 연결 중 어느 것을 사용할지는 사용자가 트러스트된 연결을 요청하는지 여부 및 연결이 표 4에 요약된 대로 서버의 트러스트된 컨텍스트에 정의된 기준에 부합하는지 여부에 따라 다릅니다.

표 4. 서로 다른 조치 조합의 결과로 표시되는 연결 유형

	연결이 트러스트되기 위한 서버의 기준에 부합함	연결이 트러스트되기 위한 서버의 기준에 부합하지 않음
연결이 트러스트되도록 요청	명시적 트러스트된 연결	일반 연결 및 경고 SQL20360W (SQLSTATE 01679)가 리턴됩니다.
연결이 트러스트되도록 요청하지 않음	내재적 트러스트된 연결	일반 연결

내재적 트러스트된 연결은 임시 역할 특권이 연결을 사용하는 동안 이 특권을 사용자에게 허용한다는 것을 제외하고는 일반 연결과 동일합니다. 허용된 역할 특권(있을 경우)은 연결이 트러스트되도록 만든 트러스트된 컨텍스트에서 지정됩니다.

내재적 트러스트된 연결은 DB2 Connect를 사용하여 연결하는 응용프로그램에 의해 작성될 수 있습니다. 내재적 트러스트된 연결은 일반 연결이 작성되고 사용되는 것과 동일한 방법으로 작성되고 사용됩니다. 이것은 응용프로그램이 DB2 Connect를 통해 연결하는 한 내재적 트러스트된 연결을 사용하기 위해 기존 응용프로그램에서 코드를 변경할 필요가 없다는 것을 의미합니다.

명시적 트러스트된 연결은 내재적 트러스트된 연결이 하는 것과 같은 방법으로 사용자에게 임시 역할 특권을 허용합니다. 또한 명시적 트러스트된 연결을 사용하면 해당 연결에 조치를 수행할 때 사용되는 권한 부여 ID를 변경할 수 있습니다. 명시적 트러스트

트된 연결에서 권한 부여 ID를 변경하는 것은 사용자를 전환하는 것으로 간주됩니다. 사용자가 전환할 수 있는 권한 부여 ID 및 전환시 지정된 권한 부여 ID에 암호가 필요한지 여부는 트러스트된 연결이 작성되도록 허용한 트러스트된 컨텍스트의 일부로서 정의됩니다.

암호가 필요 없는 사용자 이름을 사용할 경우 데이터베이스 서버에서 권한 부여 ID를 인증하지 않으므로, 사용자 전환을 사용하면 여러 사용자 사이에서 연결을 공유하는 오버헤드를 크게 줄일 수 있습니다. 그러나 이 기능을 사용할 때는 사용 중인 응용프로그램이 권한 부여 ID 유효성을 확인하고 인증하지 않고는 이 권한 부여 ID로의 전환을 허용하지 않는다는 것을 확인해야 합니다. 그렇지 않으면 시스템에 보안 허점을 만드는 것입니다.

XA 설정 연결을 포함하여 CLI 또는 JDBC를 사용하여 DB2 Connect로 연결하는 경우 명시적 트러스트된 연결을 작성하고 사용자를 전환할 수 있습니다. 명시적 트러스트된 연결을 작성하고 사용자를 전환하려면 특수 연결 속성을 설정해야 합니다. 이것은 명시적 트러스트된 연결을 사용하려면 기존 응용프로그램을 수정해야 한다는 의미입니다.

위에서 언급한 차이점을 제외하고는 트러스트된 연결(내재적 또는 명시적)을 일반 연결과 같은 방법으로 사용할 수 있습니다. 그러나 명시적 트러스트된 연결은(연결이 끊어진 상태이거나 연결이 해제된 상태인 경우에도) 사용이 종료되면 명시적으로 연결을 해제해야 합니다. 그렇지 않으면 이 연결에 사용되는 자원이 릴리스되지 않을 수 있습니다. 내재적 트러스트된 연결에서는 이것이 문제가 되지 않습니다.

주:

1.

중요: 암호를 제공하지 않고 사용자를 전환하면 데이터베이스 서버의 인증이 생략됩니다. 응용프로그램에서 이미 권한 부여 ID 유효성을 확인하고 인증한 경우가 아니라면 응용프로그램이 암호 없이 해당 권한 부여 ID로 전환하는 것을 허용하지 않아야 합니다. 다른 방법을 사용하면 보안에 허점이 생깁니다.

2. 명시적 트러스트된 연결은 CLIENT 인증을 사용하지 않아야 합니다. 이 문제는 내재적 트러스트된 연결에는 적용되지 않습니다.

3. 명시적 트러스트된 연결을 사용하는 응용프로그램은 암호로 보호되고 인증된 사람에게만 액세스가 허용되는 보안 시스템에서 실행되어야 합니다. 이 문제는 내재적 트러스트된 연결에는 적용되지 않습니다.

CLI를 통해 트러스트된 연결 작성 및 종료

연결 대상이 되는 데이터베이스 서버가 이를 허용하도록 구성된 경우 CLI를 통해 연결할 때 명시적 트러스트된 연결을 작성할 수 있습니다.

이 프로시저에서는 XA 트랜잭션 관리 프로그램을 사용하지 않는다고 간주합니다. XA 트랜잭션 관리 프로그램을 사용할 경우에는 이 프로그램이 `xa_open`을 호출할 때 구성 값 `TCTX`를 `TRUE`로 설정하도록 구성되었는지만 확인하면 됩니다. 그럴 경우 명시적 트러스트된 연결이 될 수 있는 모든 연결을 사용할 수 있습니다. 연결이 명시적 트러스트된 연결인지 확인하려면 3단계를 참조하십시오.

- 연결할 데이터베이스가 트러스트된 컨텍스트를 지원해야 합니다.
- 트러스트된 컨텍스트가 클라이언트를 트러스트 가능한 것으로 인식하도록 정의되어 있어야 합니다.
- 트러스트된 컨텍스트에서 지정된 시스템 권한 부여 ID를 알아야 합니다. 트러스트된 연결의 시스템 권한 부여 ID는 연결을 만들 때 사용자 이름으로 서버에 제공한 권한 부여 ID입니다. 사용 중인 연결이 특정 트러스트된 컨텍스트에 의해 트러스트되려면 시스템 권한 부여 ID가 해당 트러스트된 컨텍스트에서 지정된 것이어야 합니다. 유효한 시스템 권한 부여 ID 및 이 ID에 대한 암호는 보안 관리자에게 문의하십시오.

다음 지시사항의 예에서는 C 언어를 사용하며, `conn`이 올바르지만 연결되어 있지 않은 연결 핸들에 대한 포인터라고 가정합니다. 변수 `rc`는 데이터 유형 `SQLRETURN`을 갖는 것으로 가정됩니다.

1. 일반 연결에서 설정하는 연결 속성을 설정하는 것 이외에, `SQLSetConnectAttr` 함수의 호출로 연결 속성 `SQL_ATTR_USE_TRUSTED_CONTEXT`를 `SQL_TRUE`로 설정하십시오.

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

2. 인스턴스에 대한 `SQLConnect` 함수를 호출하여 일반 연결에서처럼 데이터베이스에 연결하십시오. 시스템 권한 부여 ID를 사용자 이름으로, 이 ID의 암호를 암호로 사용하십시오. 특히 표 5의 오류 및 경고가 발생하는지 점검하십시오.

표 5. 트러스트된 연결 작성 실패를 나타내는 오류

SQLCODE	SQLSTATE	의미
SQL20360W	01679	연결을 트러스트된 연결로 설정할 수 없습니다. 이 연결은 대신 일반 연결로 설정되었습니다.

별다른 오류 또는 경고 메시지가 없으면 연결이 설정되며 이는 명시적 트러스트된 연결입니다.

3. (선택사항) `SQLGetConnectAttr` 함수를 사용하여 연결 속성 `SQL_ATTR_USE_TRUSTED_CONTEXT`의 값을 확인하면 설정된 연결이 명시적 트러스트된 연결인지 확인할 수 있습니다. `SQL_TRUE`로 설정되었으면 이 연결은 명시적 트러스트된 연결입니다.

4. 연결 사용이 끝났으면 매우 신중하게 명시적으로 이 연결을 끊어야 합니다(연결이 끊어진 상태이거나 연결이 해제된 상태인 경우에도). 명시적 트러스트된 연결을 명시적으로 끊지 않으면 이 연결에 사용되는 일부 자원이 릴리스되지 않을 수 있습니다.

주:

1. 명시적 트러스트된 연결은 CLIENT 인증을 사용하지 않아야 합니다. 이 문제는 내재적 트러스트된 연결에는 적용되지 않습니다.
2. 명시적 트러스트된 연결을 사용하는 응용프로그램은 암호로 보호되고 인증된 사람에게만 액세스가 허용되는 보안 컴퓨터에서만 실행되어야 합니다. 이 문제는 내재적 트러스트된 연결에는 적용되지 않습니다.

CLI를 통해 트러스트된 연결에서 사용자 전환

명령 인터페이스(CLI)를 통한 명시적 트러스트된 연결에서 사용자를 전환할 수 있습니다. 트러스트된 연결을 사용하여 사용자를 전환하는 의미에 대한 설명은 관련 링크의 주제를 참조하십시오.

- 연결은 명시적 트러스트된 연결로 작성되어야 합니다.
- 명시적 트러스트된 연결이 트랜잭션에 있으면 안됩니다.
- 명시적 트러스트된 연결을 작성하도록 허용한 트러스트된 컨텍스트는 전환 대상이 되는 권한 부여 ID로 전환을 허용하도록 구성되어야 합니다.

다음 지시사항의 예에서는 C 언어를 사용하며, **conn**이 연결된 명시적 트러스트된 연결에 대한 포인터라고 가정합니다. 변수 *rc*는 데이터 유형 **SQLRETURN**을 갖는 것으로 가정됩니다. 변수 *newuser*는 전환 대상 사용자의 권한 부여 ID를 보유하는 문자열을 가리키는 포인터로 가정합니다. 변수 *passwd*는 해당 권한 부여 ID에 대한 암호를 포함하는 문자열을 가리키는 포인터로 가정합니다.

1. **SQLSetConnectAttr** 함수를 호출하여 **SQL_ATTR_TRUSTED_CONTEXT_USERID** 속성을 설정하십시오. 이 속성을 전환하려는 권한 부여 ID로 설정하십시오.

```
rc = SQLSetConnectAttr(
    conn,
    SQL_ATTR_TRUSTED_CONTEXT_USERID, newuser, SQL_NTS
);
//Check for errors
```

특히 표 6의 오류 및 경고가 발생하는지 점검하십시오.

표 6. 사용자 전환시 새 권한 부여 ID 설정 실패를 나타내는 오류

SQLCODE	의미
CLI0106E	연결이 연결되지 않았습니다.
CLI0197E	연결이 트러스트된 연결이 아닙니다.

표 6. 사용자 전환시 새 권한 부여 ID 설정 실패를 나타내는 오류 (계속)

SQLCODE	의미
CLI0124E	제공된 값에 문제가 있습니다. 예를 들어 값이 널(null)이 아닌지, 너무 길지는 않은지를 확인하십시오.
CLI0196E	사용자 전환을 금지하는 작업단위(UOW)와 관련된 연결입니다. 사용자를 전환하려면 연결이 트랜잭션 상태여서는 안됩니다.

2. (이 트러스트된 연결을 허용하는 트러스트된 컨텍스트가 전환 중인 권한 부여 ID의 암호를 요구하지 않는 경우에는 선택사항) `SQLSetConnectAttr` 함수를 호출하여 `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` 속성을 설정하십시오. 이 속성을 새 권한 부여 ID의 암호로 설정하십시오.

```
rc = SQLSetConnectAttr(
    conn,
    SQL_ATTR_TRUSTED_CONTEXT_PASSWORD, passwd, SQL_NTS
);
//Check for errors
```

48 페이지의 표 6 및 표 7에 나열된 오류 및 경고가 발생하는지 확인하십시오.

표 7. 사용자 전환시 암호 설정 실패를 나타내는 오류

SQLCODE	의미
CLI0198E	<code>SQL_ATTR_TRUSTED_CONTEXT_USERID</code> 속성이 설정되지 않았습니다.

3. 일반 연결에서처럼 진행하십시오. XA 트랜잭션 관리 프로그램을 사용할 경우 사용자 전환은 다음 요청의 일부로 시도됩니다. 그렇지 않으면 데이터베이스(예를 들면, `SQLExecDirect`)에 액세스하는 다음 함수 호출을 초기화하기 바로 전에 사용자 전환이 시도됩니다. 둘 중 어느 경우이든 일반적으로 점검해야 하는 오류 및 경고 이외에도 표 8에 나열된 오류를 점검해야 합니다. 표 8의 오류는 사용자 전환이 실패했음을 나타냅니다.

표 8. 사용자 전환 실패를 나타내는 오류

SQLCODE	의미
SQL1046N	이 트러스트된 연결을 허용한 트러스트된 컨텍스트가 전환 대상 권한 부여 ID로 전환을 허용하도록 구성되지 않았습니다. 트러스트된 컨텍스트를 변경해야 해당 권한 부여 ID로 전환할 수 있습니다.
SQL30082N	제공된 암호가 전환 대상 권한 부여 ID에 대해 올바르지 않습니다.
SQL0969N, 원시 오류 -20361	해당 사용자로 전환을 제한하는 데이터베이스 레벨 제한조건이 있습니다.

사용자 전환에 실패하면 다른 사용자로 전환할 때까지 연결이 연결되지 않은 상태로 남아 있습니다. 연결되지 않은 상태의 트러스트된 연결에서 사용자를 전환할 수는 있지만 데이터베이스 서버에 액세스할 수는 없습니다. 연결되지 않은 상태의 연결은 이 연결에 사용자를 전환할 때까지 그 상태로 남아 있습니다.

주:

1. **중요사항:** 암호를 제공하지 않고 사용자를 전환하면 데이터베이스 서버의 인증이 생략됩니다. 응용프로그램에서 이미 권한 부여 ID 유효성을 확인하고 인증한 경우가 아니라면 응용프로그램이 암호 없이 해당 권한 부여 ID로 전환하는 것을 허용하지 않아야 합니다. 다른 방법을 사용하면 보안에 허점이 생깁니다.
2. `SQL_ATTR_TRUSTED_CONTEXT_USERID` 속성에 `NULL` 값을 지정하는 것은 트러스트된 컨텍스트 시스템 권한 부여 ID(명시적 트러스트된 연결을 작성할 때 사용한 사용자 ID)를 지정하는 것과 같습니다.
3. 명시적 트러스트된 연결에 `SQL_ATTR_TRUSTED_CONTEXT_USERID` 연결 속성 값을 설정하면 이 연결이 즉시 재설정됩니다. 재설정의 결과는 해당 연결의 원래 연결 속성을 사용하여 새 연결을 만든 것과 같습니다. 연결 속성으로 설정한 값이 시스템 권한 부여 ID 또는 `NULL` 또는 속성이 현재 보유한 동일한 값인 경우에도 이 재설정이 발생합니다.
4. 트러스트된 연결을 허용한 트러스트된 컨텍스트가 해당 권한 부여 ID에 대해 전환 사용자의 인증을 요구하지 않더라도 `SQL_ATTR_TRUSTED_CONTEXT_PASSWORD` 속성이 설정되면 전환 사용자 처리 동안 암호가 인증됩니다. 이 결과 불필요한 오버헤드가 발생합니다. 이 규칙은 트러스트된 컨텍스트 시스템 권한 부여 ID에 적용되지 않습니다. 전환시 트러스트된 컨텍스트 시스템 권한 부여 ID가 인증을 요구하지 않으면 암호를 제공하더라도 인증되지 않습니다.

DB2 Connect 인증 고려사항

DB2 Connect 관리자는 System z 또는 IBM Power Systems 데이터베이스 관리자와 함께 사용자 이름 및 암호가 유효한 위치를 판별할 수 있습니다.

- 클라이언트에서
- System z 또는 IBM Power Systems 서버에서
- 써드 파티 시스템(Kerberos)을 통한 단일 로그인 및 유효성 점검

주: 리모트 클라이언트에서 인증 유형을 지정하지 않은 경우, 클라이언트는 디폴트로 `SERVER_ENCRYPT` 값을 지정합니다. 서버에서 이 유형을 받아들이지 않으면 클라이언트는 서버에서 리턴된 적절한 값을 사용하여 다시 시도합니다. 성능 최적화에 도움이 되도록 항상 클라이언트에서 인증 유형을 지정하여 이러한 필요없는 네트워크 플로우가 발생하지 않도록 하십시오.

DB2 Connect 버전 8.2.2(버전 8.1 FixPak 9에 해당) 이후로 게이트웨이는 인증 조정 동안 더 이상 수동적인 구성원(participant)이 아닙니다. 대신 게이트웨이는 적극적인 역할을 맡게 됩니다. 게이트웨이의 데이터베이스 디렉토리 항목에 지정된 인증 유형은 클라이언트에서 카탈로그된 인증 유형을 겹쳐줍니다. 클라이언트, 게이트웨이 및 서버 모두 호환 가능한 유형을 지정해야 합니다. 게이트웨이에서 카탈로그된 인증 유형이 데이터베이스 디렉토리 항목에 지정되지 않았으면 `SERVER` 인증이 서버에서 요구되는 기

본 유형이 됩니다. 그러나 서버가 SERVER 인증을 지원하지 않으면 클라이언트와 서버 사이에서 조정이 계속 발생합니다. 이 동작은 인증 유형이 지정되지 않은 경우 SERVER_ENCRYPT를 기본값으로 사용하는 클라이언트와 대조됩니다.

클라이언트 API 설정의 DB2NODE 또는 SQL_CONNECT_NODE 옵션이 클라이언트에 설정되지 않았으면 게이트웨이에서 카탈로그된 인증 유형이 사용되지 않습니다. 이 경우 클라이언트와 서버 사이에서 조정이 계속됩니다.

다음 인증 유형이 DB2 Connect에서 허용됩니다.

CLIENT

사용자 이름과 암호의 클라이언트에서 유효합니다.

DATA_ENCRYPT

클라이언트/서버 통신 중에 사용자 데이터를 암호화할 수 있는 기능을 제공합니다. 이 인증 유형은 IBM Power Systems 데이터베이스 서버에서 지원되지 않습니다.

KERBEROS

전형적인 ID 및 암호 조합 대신에 Kerberos 인증을 사용하여 클라이언트가 서버에 로그인할 수 있도록 합니다. 이 인증 유형을 사용하려면 서버와 클라이언트 모두 Kerberos가 사용 가능해야 합니다.

SERVER

사용자 이름 및 암호가 System z 또는 IBM Power Systems 서버 데이터베이스에서 유효성이 확인됩니다.

SERVER_ENCRYPT

SERVER 인증에서와 같이 System z 또는 IBM Power Systems 데이터베이스 서버에서 사용자 이름 및 암호의 유효성이 확인되지만 전송된 사용자 ID 및 암호가 클라이언트에서 암호화됩니다.

SERVER_ENCRYPT_AES

전송된 사용자 ID 및 암호는 클라이언트에서 AES(Advanced Encryption Standard) 암호화 알고리즘을 사용하여 암호화되고 System z 데이터베이스 서버에서 유효성이 확인됩니다.

Kerberos 인증은 클라이언트가 사용자 ID와 암호를 서버에 직접적으로 전달하지 않는 환경에 유용합니다. 대신, Kerberos는 써드 파티 인증 메커니즘으로 작동합니다. 사용자가 일단 클라이언트 터미널에서 ID와 암호를 입력하면 Kerberos가 이 로그온의 유효성을 확인합니다. 그런 다음에 Kerberos가 사용자의 권한 부여를 요청된 로컬 및 네트워크 서비스로 안전하게 자동으로 전달합니다. 이는 사용자가 리모트 DB2 서버에 로그인하기 위해 ID 및 암호를 다시 입력하지 않아도 됨을 의미합니다. Kerberos 인증에서 제공하는 단일 로그온 기능을 사용하려면 DB2 Connect 및 연결되어 있는 데이터베이스 서버가 모두 Kerberos 지원을 제공해야 합니다.

주: GSSPLUGIN 인증 유형에 대해서는 지원되지 않습니다.

Kerberos 지원

티케팅 시스템을 처리하는 Kerberos 인증 계층은 Windows 2000 Active Directory 메커니즘으로 통합되었습니다. 응용프로그램의 클라이언트와 서버 쪽에서는 각각 Kerberos SSP(Security Support Provider) 클라이언트 및 서버 모듈과 통신합니다. SSPI(Security Support Provider Interface)는 Kerberos SSP 및 기타 보안 프로토콜에 상위 레벨의 인터페이스를 제공합니다.

일반 설치

Kerberos 인증을 사용하여 DB2를 구성하려면 다음을 설정하십시오.

- 네트워크에서 공유되는 Active Directory의 DB2에 대한 권한 부여 규정(서비스로) 및
- Kerberos KDC(Key Distribution Center) 간의 신뢰 관계

가장 간단한 시나리오에서는 구성할 최소 하나의 KDC 신뢰 관계가 있습니다. 즉, 클라이언트 워크스테이션을 제어하는 KDC와 IBM Power Systems 또는 System z 간의 신뢰 관계입니다. OS/390® 버전 2 릴리스 10 또는 z/OS 버전 1 릴리스 2는 호스트가 UNIX KDC로 작동할 수 있도록 하는 RACF® 기능을 통해 Kerberos 티켓 처리를 제공합니다.

DB2 Connect는 일반적으로 3 티어 설정에서 라우터 기능을 제공합니다. Kerberos 보안이 사용될 때 인증에서 어떠한 역할도 가정하지 않습니다. 대신, 단지 클라이언트의 보안 토큰을 IBM i용 DB2 또는 DB2로 전달하기만 합니다. DB2 Connect 게이트웨이가 클라이언트 또는 호스트의 Kerberos 범주 구성원이 될 필요가 없습니다.

하위 레벨 호환성

Kerberos 지원을 위한 DB2 최소 요구사항은 다음과 같습니다.

IBM Data Server Client:

버전 8

DB2 Connect:

버전 8

DB2: 버전 7

z/OS 보안에 대한 힌트 및 추가 정보

이 주제에서는 z/OS용 DB2 데이터베이스 서버에 연결하는 DB2 Connect의 보안에 대한 일부 힌트 및 추가 정보를 제공합니다.

확장 보안 필드

z/OS용 DB2 확장 보안 필드가 YES로 설정되었는지 확인하십시오. 이 필드는 z/OS용 DB2 DSNTIPR 패널에 표시됩니다.

확장 보안 코드

z/OS용 DB2® Universal Database™ 버전 5.1까지, 사용자 ID 또는 암호를 제공한 연결 요청이 SQL30082 이유 코드 0과 함께 실패할 수 있지만 문제점이 무엇인지에 대한 다른 지시사항은 없었습니다.

z/OS용 DB2 버전 5.1은 확장된 보안 코드에 대한 지원을 제공하는 방향으로 향상되었습니다. 확장된 보안 코드를 지정하면 이유 코드 이외에 (PASSWORD EXPIRED)와 같은 추가 진단을 제공합니다.

이 기능을 사용하려면, 확장된 보안을 위해 z/OS용 DB2 ZPARM 설치 매개변수를 YES값으로 설정해야 합니다. z/OS용 DB2 설치 패널 DSN6SYSP를 사용하여 EXTSEC=YES로 설정하십시오. 또한 DDF 패널 1(DSNTIPR)을 사용하여 설정할 수도 있습니다. 디폴트값은 EXTSEC=N0입니다. 만기된 암호의 경우, DB2 Connect를 사용하는 Windows, Linux, UNIX 및 웹 응용프로그램이 SQL30082 오류 메시지를 수신합니다.

이미 확인된 TCP/IP 보안

DB2 보안 옵션 AUTHENTICATION=CLIENT에 대한 지원을 제공하려면, z/OS용 DB2 설치 패널 DSNTIP4(DDF 패널 2)를 사용하여 TCP/IP가 이미 검증한 보안을 YES로 설정하십시오.

데스크탑 ODBC 및 Java 응용프로그램 보안

워크스테이션 ODBC 및 Java 응용프로그램은 동적 SQL을 사용합니다. 이로 인해 일부 설치에서 보안 문제점이 생길 수 있습니다. z/OS용 DB2는 소유자나 바인더의 인증 하에서 동적 SQL을 실행할 수 있는 새 바인드 옵션 DYNAMICRULES(BIND)를 도입합니다.

DB2 및 DB2 Connect는 DB2CLI.INI 구성 파일에서 새 CLI/ODBC 구성 매개변수 CURRENTPACKAGESET을 제공합니다. 이 매개변수는 적합한 특권이 있는 스키마 이름으로 설정해야 합니다. SQL SET CURRENT PACKAGESET Schema 명령문이 응용프로그램용 연결마다 자동으로 실행됩니다.

ODBC 관리 프로그램을 사용하여 DB2CLI.INI를 갱신하십시오.

암호 변경 지원

사용자 ID의 암호가 만기된 경우, SQL CONNECT문이 SQLCODE -30082 이유 코드 1과 같은 오류 메시지를 리턴합니다. DB2 Connect를 사용할 경우, 암호를 리모트로 변경할 수 있습니다. z/OS용 DB2은 다음 CONNECT문을 발행하여 DRDA를 통해 암호를 변경할 수 있습니다.

```
CONNECT TO <database> USER <userid> USING <password>  
NEW <new_password> CONFIRM <new_password>
```

DB2 구성 지원 프로그램의 "암호 변경" 대화 상자를 사용하여 암호를 변경할 수도 있습니다.

DB2 Connect에서 지원되는 인증 유형

이 주제에서는 DB2 Connect에서 지원되는 인증 및 보안 설정의 여러 조합을 나열합니다.

TCP/IP 연결의 인증 유형

TCP/IP 통신 프로토콜은 네트워크 프로토콜 계층에서 인증 옵션을 지원하지 않습니다. 인증 유형은 인증 수행 위치를 판별합니다. 이 표에 표시된 조합만 DB2 Connect에서 지원됩니다. 인증 설정은 DB2 Connect 서버의 데이터베이스 디렉토리 항목에 있습니다.

표 9. 유효한 인증 시나리오

시나리오	인증 설정	유효성 점검
1	CLIENT	클라이언트
2	SERVER	IBM 메인프레임 데이터베이스 서버
3	SERVER_ENCRYPT	IBM 메인프레임 데이터베이스 서버
4	KERBEROS	Kerberos 보안
5	DATA_ENCRYPT	호스트
6	SERVER_ENCRYPT_AES	호스트 데이터베이스 서버

인증 유형에 대한 시나리오

다음 설명은 위에 설명된 표 9에 나열된 연결에 적용됩니다. 다음은 각각의 시나리오에 대한 자세한 설명입니다.

- 시나리오 1에서는 사용자 이름과 암호의 유효성이 리모트 클라이언트에서만 확인됩니다. 로컬 클라이언트에서는 사용자 이름과 암호가 DB2 Connect 서버에서만 확인됩니다.

사용자는 로그인한 위치에서 인증됩니다. 사용자 ID는 네트워크를 통하여 송신되지만 암호는 그렇지 않습니다. 모든 클라이언트 워크스테이션이 믿을 수 있는 적합한 보안 기능을 가지고 있을 경우에만, 이 보안 유형을 사용하십시오.

- 시나리오 2에서는 사용자 이름 및 암호가 IBM 메인프레임 데이터베이스 서버에서만 유효성이 확인됩니다. 사용자 ID 및 암호는 네트워크를 통해 리모트 클라이언트에서 DB2 Connect 서버로 전송되고 DB2 Connect 서버에서 IBM 메인프레임 데이터베이스 서버로 전송됩니다.
- 시나리오 3은 사용자 ID 및 암호가 암호화된다는 점을 제외하고는 시나리오 2와 동일합니다.
- 시나리오 4의 경우 Kerberos KDC에서 클라이언트가 Kerberos 티켓을 확보합니다. 티켓은 DB2 Connect를 통해 그대로 서버로 전달되고 서버에서 유효성이 확인됩니다.
- 시나리오 5는 사용자 데이터도 암호화되고 DATA_ENCRYPT가 IBM Power Systems 데이터베이스 서버를 지원하지 않는다는 점을 제외하고 시나리오 3과 동일합니다.
- 시나리오 6은 AES(Advanced Encryption Standard) 암호화 알고리즘이 사용된다는 점을 제외하고 시나리오 3과 동일합니다.

제 6 장 바인딩 응용프로그램 및 유틸리티(DB2 Connect)

Embedded SQL을 사용하여 개발된 응용프로그램은 해당 프로그램이 작동하는 데이터베이스마다 바인드되어야 합니다. 이러한 기능이 사용 가능한 플랫폼에서, 명령 센터와 클라이언트 구성 지원 프로그램을 사용하여 이를 실행할 수 있습니다.

바인딩은 각 데이터베이스에서 응용프로그램마다 일단 실행되어야 합니다. 바인드 프로세스 중에 데이터베이스 액세스 플랜은 각 SQL문에 대해 저장됩니다. 이러한 액세스 플랜은 응용프로그램 개발자에 의해 제공되며, 프리컴파일 중에 작성되는 바인드 파일에 포함됩니다. 바인딩은 IBM 메인프레임 데이터베이스 서버에서 이러한 바인드 파일을 처리하는 프로세스입니다.

DB2 Connect에 제공되는 여러 유틸리티는 Embedded SQL을 사용하여 개발되므로 이러한 유틸리티를 해당 시스템에서 사용하려면 IBM 메인프레임 데이터베이스 서버에 바인드해야 합니다. DB2 Connect 유틸리티 및 인터페이스를 사용하지 않는 경우, 각 IBM 메인프레임 데이터베이스 서버에 바인드할 필요가 없습니다. 이 유틸리티에 필요한 바인드 파일의 목록에는 다음 파일이 있습니다.

- System z용 ddcsmvs.lst
- VSE용 ddcsvse.lst
- VM용 ddcsvm.lst
- IBM Power Systems용 ddcs400.lst

데이터베이스에 이 파일 목록 중 하나를 바인딩하면 해당 데이터베이스에 개별적인 각 유틸리티를 바인드합니다.

DB2 Connect 서버 제품이 설치된 경우, DB2 Connect 유틸리티를 해당 시스템에서 사용하려면 각 IBM 메인프레임 데이터베이스 서버에 바인드해야 합니다. 클라이언트가 동일한 FixPack 레벨에 있는 경우, 관련된 클라이언트 플랫폼 수에 상관없이 유틸리티를 한 번만 바인드해야 합니다.

예를 들어, 10개의 Windows 클라이언트 및 10개의 AIX 클라이언트가 Windows 서버의 DB2 Connect Enterprise Edition을 통해 DB2에 연결되어 있는 경우, 다음 중 하나를 수행하십시오.

- Windows 클라이언트 중 하나에서 ddcsmvs.lst를 바인드하십시오.
- AIX 클라이언트 중 하나에서 ddcsmvs.lst를 바인드하십시오.
- DB2 Connect 서버로부터 ddcsmvs.lst를 바인드하십시오.

이 예는 다음의 가정을 바탕으로 합니다.

- 모든 클라이언트가 동일한 서비스 레벨에 있습니다. 그렇지 않으면, 추가로 특정 서비스 레벨의 각 클라이언트에서 바인드해야 합니다.
- 서버가 클라이언트와 동일한 서비스 레벨에 있습니다. 그렇지 않은 경우, 서버에서도 바인드해야 합니다.

DB2 Connect 유틸리티 이외에 Embedded SQL을 사용하는 다른 응용프로그램도 같이 작업하려는 데이터베이스마다 바인드되어야 합니다. 바인드되지 않은 응용프로그램을 실행하면 자주 SQL0805N 오류 메시지가 나타납니다. 바인드해야 할 모든 응용프로그램에 추가 바인드 목록 파일을 작성할 수 있습니다.

바인드할 각 IBM 메인프레임 데이터베이스 서버에 대해 다음을 수행하십시오.

1. IBM 메인프레임 데이터베이스 서버 관리 시스템에 대한 충분한 권한이 있는지 확인하십시오.

System z

요구되는 권한 부여는 다음과 같습니다.

- SYSADM 또는
- SYSCTRL 또는
- BINDADD 및 CREATE IN COLLECTION NULLID

주: BINDADD 및 CREATE IN COLLECTION NULLID 권한은 패키지가 아직 없는 경우에만 충분한 권한을 제공합니다. 예를 들어, 처음으로 패키지를 작성하고 있는 경우가 해당됩니다.

패키지가 이미 있고 다시 패키지를 바인드하는 중이면, 태스크 완료에 필요한 권한은 처음에 바인드를 수행했던 사용자에게 있습니다.

A) 원래 바인드가 있고 바인드를 다시 수행하는 경우, 위에 나열된 권한이 있으면 바인드를 완료할 수 있습니다.

B) 원래 바인드를 다른 사용자가 수행했으며 두 번째 바인드를 수행하려는 경우, 바인드를 완료하려면 SYSADM 또는 SYSCTRL 권한이 필요합니다. BINDADD와 CREATE IN COLLECTION NULLID 권한만으로는 바인드를 완료할 수 없습니다. SYSADM이나 SYSCTRL 권한이 없는 경우라도 패키지를 작성할 수 있습니다. 이런 상황에서는 바꾸려는 기존의 패키지마다 적절한 BIND 권한이 필요할 것입니다.

VSE 또는 VM

필요한 권한 부여는 DBA 권한입니다. 바인드 명령에서 GRANT 옵션을 사용하려면(개별적으로 DB2 Connect 패키지마다 액세스 권한을 부여하지 않도록), NULLID 사용자 ID에 다음 테이블에 대해 다른 사용자에게 권한을 부여하는 권한이 있어야 합니다.

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

VSE 또는 VM 시스템에서 다음을 실행할 수 있습니다.

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

IBM Power Systems

NULLID 컬렉션에 대한 CHANGE 권한 또는 그 이상

2. 다음의 명령을 실행하십시오.

```
db2 connect to DBALIAS user USERID using PASSWORD
db2 bind path@ddcsmvs.lst blocking all
      sqlerror continue messages ddcsmvs.msg grant public
db2 connect reset
```

여기서 *DBALIAS*, *USERID* 및 *PASSWORD*는 IBM 메인프레임 데이터베이스 서버에 적용되고, *ddcsmvs.lst*는 z/OS의 바인드 목록 파일이며, *path*는 바인드 목록 파일의 위치를 나타냅니다.

예를 들어, *drive:\sqllib\bnd*는 모든 Windows 운영 체제에 적용되고 *INSTHOME/sqllib/bnd/*는 모든 Linux 및 UNIX 운영 체제에 적용됩니다. 여기서, *drive*는 DB2 Connect가 설치된 논리 드라이브를 나타내고 *INSTHOME*은 DB2 Connect 인스턴스의 홈 디렉토리를 나타냅니다.

bind 명령의 권한 부여 옵션을 사용하면 EXECUTE 특권을 PUBLIC으로 또는 지정된 사용자 이름이나 그룹 ID에 권한을 부여할 수 있습니다. *bind* 명령의 권한 부여 옵션을 사용하지 않으면, 개별적으로 GRANT EXECUTE(RUN)해야 합니다.

바인드 파일용 패키지 이름을 찾으려면, 다음 명령을 입력하십시오.

```
ddcspkgn @bindfile.lst
```

예를 들면, 다음과 같습니다.

```
ddcspkgn @ddcsmvs.lst
```

위 명령은 다음과 같이 출력됩니다.

바인드 파일	패키지 이름
f:wsqllib#bnd#db2ajgrt.bnd	SQLAB6D3

DB2 Connect의 경우, 이 값을 알아내려면 *ddcspkgn* 유틸리티를 실행하십시오. 예를 들면, 다음과 같습니다.

```
ddcspkgn @ddcsmvs.lst
```

선택적으로 이 유틸리티를 사용하면 개별적인 바인드 파일의 패키지 이름을 알 수 있으며 그 예는 다음과 같습니다.

```
ddcspkgn bindfile.bnd
```

주:

- a. 바인드 옵션 `sqlerror continue`의 사용은 필수입니다. 그러나, 이 옵션은 DB2 도구 또는 명령행 처리기(CLP)를 사용하여 응용프로그램을 바인드할 때 자동으로 지정됩니다. 이 옵션을 지정하면 바인드 오류가 경고로 바뀌어 오류가 있는 파일의 바인딩이 여전히 패키지를 생성할 수 있습니다. 그런 다음, 특정 서버 구현이 다른 서버의 SQL 구문이 유효하지 않은 것으로 플래그하는 경우에도 여러 서버에 대해 하나의 바인드 파일을 사용할 수 있도록 합니다. 이러한 이유로 인해, 특정 IBM 메인프레임 데이터베이스 서버에 대해 목록 파일 `ddcsxxx.lst`를 바인딩하면 일부 경고가 생성됩니다.
 - b. DB2 Connect를 통해 DB2 데이터베이스에 연결하는 경우, 바인드 목록 `db2ubind.lst`를 사용하고 `sqlerror continue`를 지정하지 마십시오. 이 옵션은 IBM 메인프레임 데이터베이스 서버에 연결할 때만 유효합니다. 또한, DB2 데이터베이스에 연결하려면 DB2에 제공되고 DB2 Connect에는 제공되지 않는 DB2 클라이언트를 사용하도록 권장됩니다.
3. 유사한 명령문을 사용하여 각 응용프로그램이나 응용프로그램 목록을 바인드하십시오.
 4. 이전 릴리스의 리모트 DB2 클라이언트가 있는 경우, 해당 클라이언트의 유틸리티를 DB2 Connect로 바인드해야 합니다.

제 7 장 다중 사이트 갱신

분산 작업 단위(DUOW) 및 2단계 커미트라고도 하는 다중 사이트 갱신은 응용프로그램이 무결성을 손상시키지 않고 여러 개의 리모트 데이터베이스 서버에 있는 데이터를 갱신할 수 있도록 하는 기능입니다. 다중 사이트 갱신의 좋은 예로, 한 계좌의 돈을 다른 데이터베이스 서버에 있는 또 다른 계좌로 이동하는 은행 트랜잭션이 있습니다.

이러한 트랜잭션에서는 한 계좌에 대한 대변을 처리하는 데 필요한 갱신이 커미트되지 않는 한, 다른 계좌에서 차변 조작을 수행하는 갱신도 커미트되지 않도록 하는 것이 중요합니다. 이러한 계좌를 나타내는 데이터가 두 개의 서로 다른 데이터베이스 서버에 의해 관리될 때 다중 사이트 갱신 고려사항이 적용됩니다.

DB2 제품은 다중 사이트 갱신을 위한 포괄적인 지원을 제공합니다. 이러한 지원은 X/Open XA 인터페이스 스펙을 구현하는 트랜잭션 모니터(TP 모니터) 제품을 활용하는 응용프로그램 뿐만 아니라 정규 SQL을 사용하여 개발된 응용프로그램에도 사용할 수 있습니다. 이러한 TP 모니터 제품의 예로는 Microsoft Transaction Server (MTS), BEA Tuxedo 및 기타 몇몇 제품뿐만 아니라, IBM TxSeries (CICS 및 Encina), IBM Message and Queuing Series, IBM Component Broker Series, IBM San Francisco Project가 있습니다. 원시(native) SQL 다중 사이트 갱신 또는 TP 모니터 다중 사이트 갱신 중 어떤 것을 사용하는지에 따라 설정 요건이 달라집니다.

원시 SQL 및 TP 모니터 다중 사이트 갱신 프로그램은 CONNECT 2 SYNCPOINT TWOPHASE 옵션을 사용하여 사전에 컴파일되어야 합니다. 이 둘 모두 SQL Connect 명령문을 사용하여 후속 SQL문에 사용할 데이터베이스를 나타낼 수 있습니다. DB2에 트랜잭션을 조정할 것임을 알리는 TP 모니터가 없으면(TP 모니터로부터 xa_open 호출을 수신하여 데이터베이스 연결을 설정하는 DB2에 표시된 대로), DB2 소프트웨어를 사용하여 트랜잭션을 조정합니다.

TP 모니터 다중 사이트 갱신을 사용할 경우, 응용프로그램은 CICS SYNCPOINT, Encina Abort(), MTS SetAbort()와 같은 TP 모니터의 API를 사용하여 커미트 또는 롤백을 요청해야 합니다. 원시 SQL 다중 갱신을 사용하는 경우, 일반적인 SQL COMMIT 및 ROLLBACK을 사용해야 합니다.

TP 모니터 다중 사이트 갱신은 DB2 및 Oracle, Informix 또는 SQL Server와 같은 비DB2 자원 관리자에 액세스하는 트랜잭션을 조정할 수 있습니다. 원시(native) SQL 다중 사이트 갱신은 DB2 서버에서만 사용됩니다.

다중 사이트 갱신 트랜잭션이 작동하려면 분산 트랜잭션에 참여하는 각 데이터베이스가 분산 작업 단위(DUOW)를 지원할 수 있어야 합니다. 현재, 다음 DB2 서버가 분산 트랜잭션에 참여할 수 있도록 하는 DUOW 지원을 제공합니다.

- Linux, UNIX 및 Windows용 DB2 버전 8 이상
- z/OS용 DB2 버전 7 이상
- IBM i용 DB2

분산 트랜잭션은 지원되는 데이터베이스 서버의 어떤 조합도 갱신할 수 있습니다. 예를 들어, 응용프로그램은 단일 트랜잭션 내에서 Windows의 DB2 데이터베이스, z/OS용 DB2 데이터베이스 및 i용 DB2 데이터베이스에 있는 여러 테이블을 갱신할 수 있습니다.

제어 센터에서 다중 사이트 갱신 가능

제어 센터에서 다중 사이트를 갱신할 수 있습니다.

다중 사이트를 갱신하려면 다음을 수행하십시오.

1. 제어 센터를 실행하십시오.
2. [+] 부호를 눌러 트리 보기를 확장하십시오.
3. 오른쪽 마우스 단추를 사용하여 구성하려는 인스턴스를 선택하십시오. 팝업 메뉴가 열립니다.
4. **다중 사이트 갱신** → **구성** 메뉴 항목을 선택하십시오. 다중 사이트 갱신 마법사가 열립니다.
5. 아래 지정된 **TP 모니터 사용**을 선택하고 트랜잭션 프로세서(TP) 모니터를 지정하십시오. 이 필드는 사용 가능한 TP 모니터에 대한 디폴트를 보여줍니다. TP 모니터를 사용하지 않으려면 **TP 모니터 사용 안함**을 선택하십시오.

다음을 누르십시오.

6. TP 모니터를 사용 중인 경우, 동기점 관리 프로그램 설정값을 지정하십시오. TP 모니터를 사용 중이지 않은 경우, 트랜잭션 관리 프로그램 데이터베이스를 지정하십시오.
7. 완료를 누르십시오.

제어 센터를 사용한 다중 사이트 갱신 테스트

제어 센터를 사용하여 다중 사이트 갱신 설정을 테스트할 수 있습니다.

다중 사이트 갱신을 테스트하려면 다음을 수행하십시오.

1. 오른쪽 마우스 단추로 인스턴스를 선택한 후 팝업 메뉴에서 **다중 사이트 갱신** → **테스트** 메뉴 옵션을 선택하십시오. 다중 사이트 갱신 테스트 창이 열립니다.
2. **사용 가능한 목록** 상자의 **사용 가능한 데이터베이스**에서 테스트할 데이터베이스를 선택하십시오. 가운데에 있는 화살표 단추(> 및 >>)를 사용하여 선택된 목록 상자

로 또는 이 목록 상자로부터 선택사항을 이동할 수 있습니다. 또한 선택된 사용자 ID와 암호를 선택된 목록 상자에서 직접 편집하여 변경할 수도 있습니다.

3. 선택이 완료되었으면 확인을 누르십시오. 다중 사이트 갱신 테스트 결과 창이 열립니다.
4. 다중 사이트 갱신 테스트 결과 창에서는 선택한 데이터베이스 중 갱신 테스트에 성공 또는 실패한 데이터베이스를 보여줍니다. 창에서는 실패한 데이터베이스에 대해 SQL 코드와 오류 메시지도 표시합니다. 닫기를 눌러 창을 닫습니다.
5. 닫기를 눌러 다중 사이트 갱신 테스트 창을 닫습니다.

다중 사이트 갱신 및 동기점 관리 프로그램

IBM 메인프레임 데이터베이스 서버에서는 DB2 Connect가 Linux, Windows, UNIX 및 웹 응용프로그램에서 시작되는 분산 트랜잭션에 참여해야 합니다. 그리고, IBM 메인프레임 데이터베이스 서버와 관련된 여러 다중 사이트 갱신 시나리오에서 동기점 관리 프로그램(SPM) 구성요소를 구성해야 합니다. DB2 인스턴스가 작성되면, DB2 SPM 이 디폴트 설정으로 자동 구성됩니다.

SPM의 필요 여부는 선택된 프로토콜(TCP/IP)과 사용되는 TP 모니터에 의해 결정됩니다. 다음 테이블에서 SPM을 사용해야 하는 시나리오에 대한 요약 내용을 참조하십시오. 또한, 이 테이블은 Intel[®] 또는 UNIX 머신에서 IBM 메인프레임에 액세스하는 데 DB2 Connect가 필요한지 여부를 표시합니다. 다중 사이트 갱신에서, TP 모니터를 사용하는 경우에는 DB2 Connect의 APM 구성요소가 필요합니다.

표 10. SPM - TCP/IP가 필요한 다중 사이트 갱신 시나리오

트랜잭션 프로세서 모니터가 사용됩니까?	동기점 관리 프로그램이 필요합니까?	필요한 제품(한 개 선택)	지원되는 IBM 메인프레임 데이터베이스
예	예	DB2 Connect 서버 제품 DB2 Connect 라이선스가 적용된 DB2 Enterprise Server Edition	z/OS용 DB2 V7 DB2 V8 이상
아니오	아니오	DB2 Connect Personal Edition DB2 Connect 서버 제품 DB2 Connect 라이선스가 적용된 DB2 Enterprise Server Edition	DB2 V7 DB2 V8 이상

주: 분산 트랜잭션은 지원되는 데이터베이스 서버의 어떤 조합도 갱신할 수 있습니다. 예를 들어, 응용프로그램은 단일 트랜잭션 내에서 Windows의 DB2 데이터베이스, DB2 데이터베이스 및 IBM i용 DB2 데이터베이스에 있는 여러 테이블을 갱신할 수 있습니다.

XA 호환 트랜잭션 관리 프로그램으로 DB2 Connect 구성

이 주제에서는 TP 모니터 내에서 IBM Power Systems 및 System z 데이터베이스 서버를 사용하는 데 필요한 구성 단계에 대해 설명합니다.

작동 가능한 TP 모니터가 있어야 하고 DB2 Connect가 설치되어 있어야 하며, IBM 메인프레임 데이터베이스 서버에 대한 연결이 구성되어 있고 테스트가 완료되어야 합니다.

DB2 Connect가 TP 모니터 내에서 IBM Power Systems 및 System z 데이터베이스 서버를 사용하도록 구성하려면 다음 단계를 수행하십시오.

1. DB2 XA 스위치에 액세스할 수 있도록 TP 모니터를 구성하십시오. DB2 XA 스위치는 TP 모니터에 DB2 Connect의 XA API 주소를 제공합니다. TP 모니터마다 이를 수행하기 위한 서로 다른 방법을 가지고 있습니다.
2. DB2의 XA_OPEN 문자열이 있는 TP 모니터를 구성하십시오. 각 TP 모니터는 자신만의 방법으로 이를 수행합니다. TP 모니터가 사용할 DB2의 XA OPEN 문자열을 구성하는 방법에 대한 내용을 해당 TP 모니터 문서에서 참조하십시오.
3. 필요할 경우, DB2 Connect SPM(Sync Point Manager) 디폴트 구성 매개변수를 수정하십시오. IBM 호스트 및 System i(버전 5 릴리스 3 이전) 데이터베이스 서버는 아직 XA 인터페이스를 지원하지 않습니다. System i 버전 5 릴리스 4 이상은 XA를 완벽하게 지원합니다.

SPM은 XA 2단계 커밋 프로토콜을 IBM 메인프레임 데이터베이스 서버에서 사용되는 2단계 커밋 프로토콜로 맵핑하는 DB2 Connect의 구성요소입니다. 디폴트로, DB2 인스턴스는 사전 정의된 SPM 구성 매개변수 값을 가집니다. 가장 중요한 매개변수는 데이터베이스 관리 프로그램 구성 매개변수 SPM_NAME입니다. 디폴트값은 TCP/IP 호스트 이름의 첫 7자를 변형한 값입니다.

TCP/IP를 사용하여 DB2에 연결하는 경우, 디폴트 설정을 변경할 필요가 없습니다. 이러한 경우에는 이미 조작 가능하므로 SPM 구성이 필요하지 않습니다.

느슨하게 결합된 트랜잭션에 대한 DB2 Connect 지원

DB2 Connect 내에서 느슨하게 결합된 트랜잭션에 대한 지원은 IBM i용 DB2 버전 5 릴리스 4 이상과 DB2 버전 7 이상에 액세스하는 XA 분산 응용프로그램을 구현하는 사용자를 위한 것입니다. 이 지원은 동일한 전역 트랜잭션의 다른 분기가 DB2에서 잠금 스페이스를 공유할 수 있도록 합니다.

느슨하게 결합된 트랜잭션에 대한 지원은 .NET 및 COM+ 응용프로그램을 위한 것입니다.

이 기능으로 분산 트랜잭션의 분기에서 동일한 전역 트랜잭션 내의 다른 분기 때문에 잠금 시간종료나 교착 상태가 발생하는 창을 감소시킬 수 있습니다.

제 8 장 DB2 Connect에서 데이터 이동

호스트 데이터베이스 시스템 및 워크스테이션 사이에서 데이터를 이동해야 하는 복합 환경에서 작업하는 경우 호스트 및 워크스테이션 사이에서 데이터 전송을 담당하는 게이트웨이로 DB2 Connect를 사용할 수 있습니다(그림 8 참조).

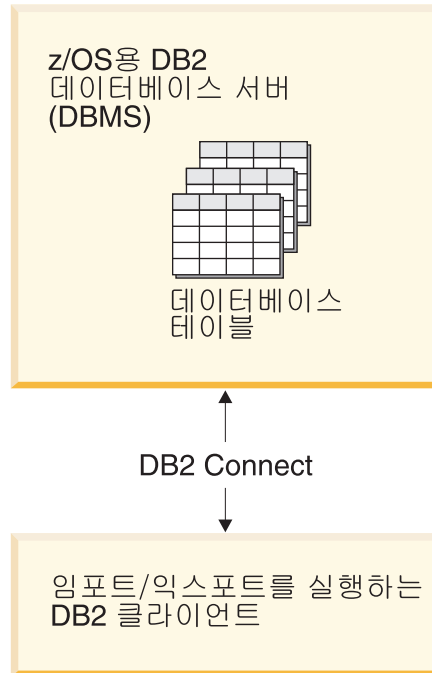


그림 8. DB2 Connect를 통해 임포트/익스포트

DB2 익스포트 및 임포트 유틸리티를 사용하면 IBM 메인프레임 서버 데이터베이스에서 DB2 Connect 워크스테이션의 파일로(또는 이 반대 방향) 데이터를 이동할 수 있습니다. 그러면 이 익스포트 또는 임포트 형식을 지원하는 다른 응용프로그램 또는 관계형 데이터베이스 관리 시스템에서 데이터를 사용할 수 있습니다. 예를 들어, IBM 메인프레임 서버 데이터베이스에서 PC/IXF 파일로 데이터를 익스포트한 후 Linux, UNIX 및 Windows용 DB2 데이터베이스 데이터베이스로 임포트할 수 있습니다.

데이터베이스 클라이언트 또는 DB2 Connect 워크스테이션에서 익스포트 및 임포트 작업을 수행할 수 있습니다.

주:

1. 익스포트 또는 임포트할 데이터는 두 데이터베이스에서 적용 가능한 크기 및 데이터 유형 제한사항을 준수해야 합니다.

2. 임포트 성능을 향상시키기 위해 복합 쿼리를 사용할 수 있습니다. 임포트 유틸리티에서 compound 파일 유형 수정자를 지정하여 지정된 수의 쿼리 명령문을 하나의 블록으로 그룹화합니다. 그러면 네트워크 오버헤드가 줄어들고 응답 시간이 향상될 수 있습니다.

DB2 Connect에서 익스포트 및 임포트 조작은 다음 조건을 만족해야 합니다.

- 파일 유형은 PC/IXF여야 합니다.
- 데이터와 호환 가능한 속성을 포함하는 목표 테이블은 이를 임포트하기 전에 목표 서버에서 작성해야 합니다. db2look 유틸리티를 사용하여 소스 테이블의 속성을 가져올 수 있습니다. INSERT만 지원되는 옵션이므로 DB2 Connect를 통한 임포트로는 테이블을 작성할 수 없습니다.

이 조건을 만족하지 못하면 조작에 실패하고 오류 메시지가 리턴됩니다.

주: 인덱스 정의는 익스포트에서 저장되거나 임포트에서 사용되지 못합니다.

혼합된 데이터(1바이트 및 2바이트 데이터를 모두 포함하는 컬럼)를 익스포트 또는 임포트하는 경우 다음을 고려합니다.

- EBCDIC로 데이터를 저장하는 시스템(MVS™, System z, IBM Power Systems, VM 및 VSE)의 경우, 시프트 아웃(Shift-Out) 및 시프트 인(Shift-In) 문자는 2바이트 데이터의 시작 및 끝을 표시합니다. 데이터베이스 테이블의 컬럼 길이를 정의하는 경우 이 문자를 표시할 공간이 충분히 있어야 합니다.
- 컬럼 데이터의 패턴이 일관되지 않으면 가변 길이 문자 컬럼이 권장됩니다.

워크스테이션에서 호스트 서버로 데이터 이동

호스트 또는 System i 서버 데이터베이스에서 데이터를 이동하려면 다음을 수행합니다.

1. DB2 테이블에서 PC/IXF 파일로 데이터를 익스포트하십시오.
2. INSERT 옵션을 사용하여 호스트 서버 데이터베이스에서 호환 가능한 테이블로 PC/IXF 파일을 임포트하십시오.

호스트 서버 데이터베이스에서 워크스테이션으로 데이터를 이동하려면 다음을 수행합니다.

1. 호스트 서버 데이터베이스 테이블에서 PC/IXF 파일로 데이터를 익스포트하십시오.
2. DB2 테이블로 PC/IXF 파일을 임포트하십시오.

예

이 예에서는 워크스테이션에서 호스트 또는 System i 서버 데이터베이스로 데이터를 이동하는 방법을 보여줍니다.

다음 명령을 실행하여 외부 IXF 형식으로 데이터를 익스포트합니다.

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

다음 명령을 실행하여 목표 DB2 데이터베이스에 대한 DRDA 연결을 설정합니다.

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

아직 없는 경우 목표 DB2 데이터베이스 인스턴스에서 목표 테이블을 작성합니다.

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

데이터를 임포트하려면 다음 명령을 실행하십시오.

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

데이터의 각 행은 IXF 형식으로 파일에서 읽히지며 SQL INSERT문이 실행되어 행을 테이블 mydb.staff로 삽입합니다. 모든 데이터를 목표 테이블로 이동할 때까지 단일 행이 계속 삽입됩니다.

세부사항 정보는 IBM Redbooks® 서적의 "DB2 계열에서 데이터 이동"을 참조하십시오. 이 Redbooks 서적은 다음 URL에서 찾을 수 있습니다.

<http://www.redbooks.ibm.com/redbooks/SG246905>.

제 9 장 SQLCODE 맵핑

다른 IBM 관계형 데이터베이스 제품은 유사한 오류에 대해 항상 동일한 SQLCODE를 생성하지는 않습니다. SQLCODE가 같더라도 다르게 지정된 토큰이 수반될 수 있습니다. 토큰 목록은 SQLCA의 SQLERRMC 필드에서 전달됩니다. 디폴트로, DB2 Connect는 각 IBM 메인프레임 데이터베이스 서버의 SQLCODE 및 토큰을 적절한 DB2 SQLCODE로 맵핑합니다.

SQLCODE 맵핑을 해제하려면 DCS 디렉토리의 매개변수 문자열에서 **NOMAP**을 지정하십시오.

IBM 메인프레임 데이터베이스 서버(예: z/OS용 DB2)에서 응용프로그램을 직접 포트하는 경우, SQLCODE 맵핑을 해제하려고 할 수 있습니다. 이 경우, 참조하는 SQLCODE를 변경하지 않고 응용프로그램을 사용할 수 있습니다.

SQLCODE 맵핑 해제

SQLCODE 맵핑을 해제하려면 DCS 디렉토리의 매개변수 문자열에서 NOMAP를 지정하십시오.

IBM 메인프레임 데이터베이스 서버(예: z/OS용 DB2)에서 응용프로그램을 직접 포트하는 경우, SQLCODE 맵핑을 해제하려고 할 수 있습니다. 이 경우, 참조하는 SQLCODE를 변경하지 않고 응용프로그램을 사용할 수 있습니다.

SQLCODE 맵핑 조정

디폴트로, DB2 Connect는 각 IBM 메인프레임 데이터베이스 서버의 SQLCODE 및 토큰을 적절한 DB2 SQLCODE로 맵핑합니다. 다음 파일은 디폴트 SQLCODE 맵핑의 복사 파일입니다.

- dcs1dsn.map은 z/OS용 DB2 SQLCODE를 맵핑합니다.
- dcs1ari.map은 DB2 서버 SQLCODE를 맵핑합니다.
- dcs1qsq.map은 IBM i용 DB2 SQLCODE를 맵핑합니다.

Linux 또는 UNIX 운영 체제의 DB2에는 맵핑이 필요하지 않습니다.

1. 디폴트 SQLCODE 맵핑을 겹쳐쓰거나 SQLCODE 맵핑이 없는 IBM 메인프레임 데이터베이스 서버(비IBM 데이터베이스 서버)를 사용하는 경우, 이러한 파일 중 하

나를 복사하여 새 SQLCODE 맵핑 파일의 기본 파일로 사용할 수 있습니다. 파일을 직접 편집하지 않고 복사하면 필요할 경우 항상 원래 SQLCODE 맵핑을 참조할 수 있습니다.

2. DCS 디렉토리의 매개변수 문자열에서 새 SQLCODE 맵핑 파일의 파일 이름을 지정하십시오.
3. 각 맵핑 파일은 ASCII 파일이며 ASCII 편집기를 사용하여 작성하고 편집합니다. 처음 설치하면, 파일은 설치 경로의 map 디렉토리에 저장됩니다.

파일에는 다음의 특정 유형의 행이 포함될 수 있습니다.

&& 파일의 논리적 시작입니다. 처음으로 &&가 발견되기 전의 모든 행은 특별한 형식이 없는 주석으로 가정되며 무시됩니다. 파일에서 && 다음에 아무 것도 없으면, SQLCODE 맵핑이 수행되지 않습니다. 이전에 설명한 대로 NOMAP 매개변수로 SQLCODE 맵핑을 중단할 수도 있습니다.

***** 한 행의 첫 번째 문자로서 주석을 나타냅니다.

W 행에서 오직 하나의 문자로서 경고 플래그가 다시 맵핑되어야 함을 나타냅니다. 디폴트로 원래 경고 플래그가 전달됩니다. W는 대문자여야 합니다.

&& 이후의 다른 행은 모두 공백이거나 다음 형식의 맵핑 명령문이어야 합니다.

```
input_code [, output_code [, token_list]]
```

*input_code*는 다음 중 하나입니다.

sqlcode

IBM 메인프레임 데이터베이스 서버의 SQLCODE

U 정의되지 않은 모든 음수 SQLCODE(이 파일에 나열되지 않음)가 지정된 *output_code*로 맵핑됩니다. 이 행에 *output_code*를 지정하지 않으면, 원래 SQLCODE가 사용됩니다. 이 문자는 대문자여야 합니다.

P 정의되지 않은 모든 양수 SQLCODE(이 파일에 나열되지 않음)가 지정된 *output_code*로 맵핑됩니다. 이 행에 *output_code*를 지정하지 않으면, 원래 SQLCODE가 사용됩니다. 이 문자는 대문자여야 합니다.

ccnn IBM 메인프레임 데이터베이스 서버의 SQLSTATE 클래스 코드. *nn*은 다음 중 하나입니다.

00 규정되지 않은 성공적 완료

01 경고

02 데이터 없음

21 카디널리티(cardinality) 위반

22 데이터 예외

23 제한 조건 위반

- 24 유효하지 않은 커서 안정성
- 26 유효하지 않은 SQL문 ID
- 40 트랜잭션 롤백
- 42 액세스 위반
- 51 유효하지 않은 응용프로그램 상태
- 55 오브젝트가 전제조건 상태에 없음
- 56 기타 SQL 또는 제품 오류
- 57 자원 사용 불능 또는 조작원 개입
- 58 시스템 오류

지정된 *output_code*는 맵핑 파일에 명시적으로 지정되지 않은 이 클래스 코드를 갖는 모든 SQLCODE에 사용됩니다. 이 행에 *output_code*를 지정하지 않으면, 원래 SQLCODE가 토큰 복사 없이 자체적으로 맵핑됩니다.

문자 **cc**는 소문자여야 합니다.

동일한 *input_code*가 맵핑 파일에 두 번 이상 나타나면, 첫 번째 항목이 사용됩니다. *output_code*는 출력 SQLCODE입니다. 지정된 값이 없으면, 원래 SQLCODE가 사용됩니다.

출력 코드를 지정하면, 다음 중 하나를 지정할 수도 있습니다.

- (s) 입력 SQLCODE+제품 ID(ARI, DSN 또는 QSQ)가 SQLCA 메시지 토큰 필드로 들어갈 것입니다.

원래 SQLCODE가 유일한 토큰으로 리턴됩니다. 이 옵션은 +965와 -969를 제외하고 정의되지 않은 SQLCODE를 처리하도록 설계되었습니다. +965 또는 -969가 *output_code*인 경우, SQLCA의 SQLERRMC 필드에 리턴된 토큰 목록에 원래 SQLCODE, 제품 ID, 원래 토큰 목록이 차례대로 포함됩니다.

문자 **s**는 소문자여야 합니다.

(token-list)

토큰 목록은 쉼표로 구분됩니다. 특정한 토큰을 생략하려면 쉼표만 지정하십시오. 예를 들어, 형식(*t2*,*t4*)은 첫 번째 및 세 번째 출력 토큰이 널(NULL)이라는 것을 의미합니다.

토큰마다 숫자의 형식(*n*)이 있으며, 선택적으로 **c**가 앞에 오거나, 선택적으로 **c** 또는 **i**가 올 수 있습니다. 다음과 같이 해석됩니다.

- c** 여기서, 토큰의 데이터 유형은 CHAR(디폴트값)입니다. **c**가 *n*전에 나오는 경우 입력 토큰을 참조하고, *n* 다음에 나오면 출력 토큰을 참조하십시오. 문자 **c**는 소문자여야 합니다.

i 여기서, 토큰의 데이터 유형은 INTEGER입니다. **i**가 *n* 다음에 나오면, 출력 토큰을 참조하십시오. IBM 메인프레임 데이터베이스 서버 제품은 CHAR 토큰만 지원하므로 **i**가 *n* 앞에 오면 안 됩니다. 문자 **i**는 소문자여야 합니다.

n 사용되는 IBM 메인프레임 데이터베이스 서버 토큰을 나타내는 번호. 번호는 출력 SQLCA에서 예정된 위치 순서로 배열됩니다. 번호는 IBM 메인프레임 데이터베이스 서버 토큰을 나타내며, 배열은 SQLCA에서 토큰이 위치하는 순서를 나타냅니다.

예를 들어, IBM 메인프레임 데이터베이스 서버가 토큰 1 및 2를 리턴할 수 있습니다. 출력 SQLCA에서 토큰 2가 토큰 1 앞에 오도록 하려면 (2,1)을 지정하십시오.

많은 토큰 번호는 마침표로 연결하여 CHAR 출력 토큰 형식으로 결합할 수 있습니다.

쉼표는 출력 토큰을 분리하는 데 사용됩니다. 쉼표 앞에 지정된 토큰이 없으면, 거기서 SQLCA에 포함되는 출력 토큰은 없습니다. 마지막으로 지정된 토큰 다음에 나타나는 출력 SQLCA의 토큰은 모두 널(NULL) 토큰으로 매핑됩니다.

그림 9는 샘플 SQLCODE 매핑 파일을 나타냅니다.

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

그림 9. SQLCODE 매핑 파일

다음 설명은 앞의 그림에 있는 행 번호에 대응됩니다.

1. SQLCODE가 -007에서 -007까지 매핑됩니다. IBM 메인프레임 데이터베이스 서버로부터 수신되는 첫 번째 입력 토큰은 첫 번째 출력 토큰으로 사용되며, 디폴트 CHAR로 지정됩니다. 다른 토큰은 전송되지 않습니다.
2. SQLCODE가 -010에서 -010까지 매핑됩니다. (지정된 출력 SQLCODE가 없습니다.) 출력 SQLCA로 들어간 토큰은 없습니다.

3. SQLCODE가 -060에서 -171까지 맵핑됩니다. IBM 메인프레임 데이터베이스 서버로부터 수신되는 첫 번째 입력 토큰을 버립니다. 두 번째 토큰은 출력 SQLCA에서 첫 번째 토큰으로 사용되었고, CHAR입니다. 출력 SQLCA에 두 번째 토큰은 없습니다.
4. SQLCODE가 -204에서 -204까지 맵핑됩니다. IBM 메인프레임 데이터베이스 서버로부터 수신되는 첫 번째 및 두 번째 토큰은 CHAR입니다. 두 입력 토큰은 CHAR 출력 토큰을 형성하기 위해 결합되며, SQLCA의 첫 번째 출력 토큰이 됩니다.
5. SQLCODE가 -633에서 -206까지 맵핑됩니다. IBM 메인프레임 데이터베이스 서버로부터 수신되는 첫 번째 입력 토큰은 CHAR입니다. 토큰은 INTEGER로 변환되며 출력 SQLCA에서 두 번째 토큰으로 사용됩니다. 출력 SQLCA의 첫 번째 토큰은 널(NULL)이고 쉼표로 나타납니다.
6. SQLCODE가 -30021에서 -30021까지 맵핑됩니다. IBM 메인프레임 데이터베이스 서버로부터 수신되는 첫 번째 및 두 번째 입력 토큰은 CHAR이며, 출력 SQLCA에서 첫 번째 및 두 번째 토큰으로 사용됩니다.
7. 00 클래스에서 SQLSTATE와 함께 SQLCA의 SQLCODE는 모두 SQLCODE +000로 맵핑됩니다.
8. 정의되지 않은 모든 SQLCODE는 -969로 맵핑됩니다. 이 옵션은 모든 맵핑이 가능하고 식별할 수 있으며 맵핑이 필요없는 코드를 포함한 코드가 나열된 경우에만 사용할 수 있습니다. (s) 옵션은 SQLCA의 SQLERRMC 필드에서 리턴된 토큰 목록에 원래 SQLCODE가 있으며, 오류가 발생한 제품, 원래 토큰 목록이 다음에 온다는 것을 나타냅니다. U 항목이 없으면 나열되지 않은 모든 코드는 맵핑없이 전달됩니다.
9. 정의되지 않은 양수 SQLCODE는 모두 +965로 맵핑됩니다. 이 옵션은 모든 맵핑이 가능하고 식별할 수 있으며 맵핑이 필요없는 코드를 포함한 코드가 나열된 경우에만 사용할 수 있습니다. (s) 옵션은 SQLCA의 SQLERRMC 필드에서 리턴된 토큰 목록에는 원래 SQLCODE가 있으며, 경고가 발생한 제품, 원래 토큰 목록이 다음에 온다는 것을 나타냅니다. P 항목이 없으면, 나열되지 않은 모든 양수 코드는 맵핑없이 전달됩니다.

제 10 장 데이터베이스 시스템 모니터링과 DB2 Connect

DB2 Connect를 사용하여 환경에서 연결 및 성능을 모니터링하는 여러 가지 방법이 설명되어 있습니다. 수행된 모니터링 유형은 운영 체제에 따라 고유합니다.

리모트 클라이언트에 대한 연결 모니터링

데이터베이스 시스템 모니터와 함께 DB2 Connect Enterprise Edition과 같은 DB2 Connect 서버 제품을 사용하여 리모트 클라이언트 연결을 모니터링할 수 있습니다. 서버 자체에서 실행되는 DB2 Connect 서버의 로컬 클라이언트를 모니터링하려면 다음 변수를 설정해야 합니다.

```
db2set DB2CONNECT_IN_APP_PROCESS=NO
```

예를 들어, IBM 메인프레임 시스템에서 오류가 발생하면 시스템 관리자가 문제점이 DB2 Connect 워크스테이션에 있는지 판별할 수 있습니다. 데이터베이스 시스템 모니터는 다음과 관련이 있습니다.

- 비보호 대화에 사용되는 DRDA 상관 토큰(CRRTKN).
- DRDA-3 동기점 관리 프로그램(TCP/IP 연결에 사용된)에 의해 보호되는 2단계 연결용 작업 단위 ID(UOWID)
- DB2 Connect 연결 ID(응용프로그램 ID)

이 정보를 통해 문제가 발생한 DB2 Connect 연결을 알 수 있으므로, 시스템 관리자는 DB2 Connect 연결을 사용 중인 다른 클라이언트에 영향을 주지 않으면서 시스템에서 개별적으로 클라이언트 응용프로그램을 강제 중단할 수 있습니다.

모니터 스위치의 상태 나열

모니터 스위치 상태를 나열하려면, db2 get monitor switches 명령을 사용하십시오.

Windows 성능 모니터를 사용하여 성능 모니터링

Windows 운영 체제는 DB2 응용프로그램의 성능을 모니터링하는 유용한 도구를 제공합니다. Windows 관리 도구 중 하나인 성능 모니터는 시스템 성능을 그래픽으로 표시합니다. 모니터링할 여러 가지 시스템, 데이터베이스 및 통신 관련 항목을 선택하여 이들을 그래픽 화면으로 함께 맵핑할 수 있습니다.

예를 들어, GET SNAPSHOT FOR ALL DCS DATABASES 또는 GET SNAPSHOT FOR ALL DCS APPLICATIONS 명령을 통해 사용 가능한 보고서를 실시간으로 모니터를 사용하여 그래프 처리하고 CPU 사용과 같은 값과 직접 비교할 수

있습니다. 데이터베이스 또는 통신 성능에 대한 여러 설정값의 효과를 직접 비교할 수 있습니다. PMC 파일에 설정값의 특수화된 구성을 저장할 수 있으며 이 파일은 나중에 검색할 수 있습니다.

예를 들어, 아래 그림에는 CPU 사용에 대한 여러 개의 DB2 측정이 그래프로 표시되었습니다. 도형화된 값의 콜렉션은 db2chart.pmc 파일에 저장되어 있습니다. 각각 시스템 성능의 서로 다른 측면을 반영하는 PMC 파일을 원하는 수만큼 저장할 수 있습니다.

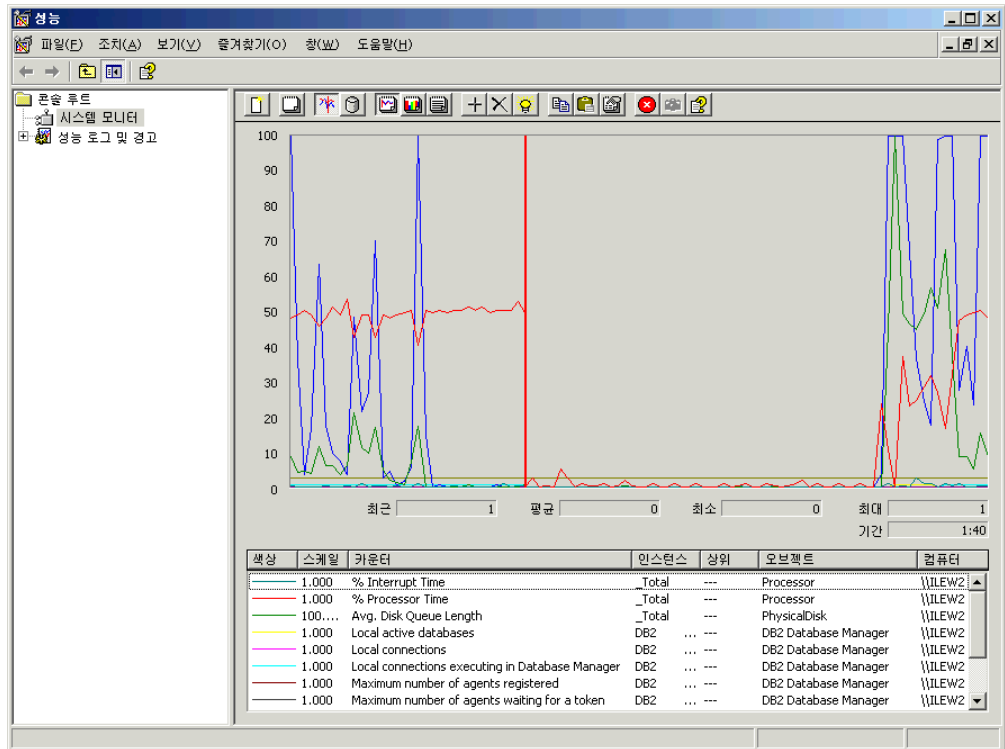


그림 10. 성능 모니터

로컬 응용프로그램의 모니터링을 사용하려면 DB2CONNECT_IN_APP_PROCESS 환경 변수를 해제해야 합니다.

GET SNAPSHOT 명령 사용

DB2 모니터는 중요한 시스템 정보의 현재 기록을 유지보수합니다. GET SNAPSHOT 명령을 실행하여 시스템 상태의 요약본을 가져올 수 있습니다. 모니터할 데이터베이스 관리 프로그램 인스턴스에 대해 SYSMANT, SYSCTRL 또는 SYSADM 권한이 있는 경우에는 모니터 스냅샷을 작성할 수 있습니다.

DCS 정보를 모니터하기 위해 다음의 5개 스냅샷 명령을 사용할 수 있습니다.

- GET SNAPSHOT FOR ALL DCS DATABASES

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

각 스냅샷 명령은 사용자가 요청한 영역에 관해 상세한 보고서를 작성합니다.

인스턴스의 경우, GET SNAPSHOT FOR DCS DATABASE ON DCSDB를 발행하면 다음의 보고서가 작성됩니다.

DCS 데이터베이스 스냅샷

DCS 데이터베이스 이름	= DCSDB
호스트 데이터베이스 이름	= GILROY
첫 번째 데이터베이스 연결 시간 소인	= 12-15-2001 10:28:24.596495
연결한 가장 최신의 경과 시간	= 0.950561
가장 최신의 경과된 연결 지속기간	= 0.000000
호스트 응답 시간(sec.ms)	= 0.000000
최종 재설정 시간소인	=
시도된 SQL문의 수	= 2
시도된 Commit문	= 1
시도된 Rollback문	= 0
실패한 명령문 조작	= 0
전체 게이트웨이 연결 수	= 1
현재 게이트웨이 연결 수	= 1
호스트 응답을 대기중인 게이트웨이 연결	= 0
클라이언트 응답을 대기중인 게이트웨이 연결	= 1
호스트로의 게이트웨이 통신 오류	= 0
마지막 통신 오류의 시간소인	= 없음
게이트웨이 연결에 할당된 상위 워터 마크	= 1
선택된 행	= 0
송신된 아웃바운드 바이트 수	= 140
수신된 아웃바운드 바이트 수	= 103

이 보고서에서는 데이터베이스 연결, 성능, SQL 요청의 오류 및 처리량에 관한 정보를 제공합니다. 실제로는 DB2 모니터 스냅샷이 훨씬 상세합니다. 인스턴스의 경우, GET SNAPSHOT FOR ALL DCS APPLICATIONS 명령을 실행하면 다음과 유사한 보고서를 볼 수 있습니다.

DCS 응용프로그램 스냅샷

클라이언트 응용프로그램 ID	= 09150F74.B6A4.991215152824
시퀀스 번호	= 0001
권한 부여 ID	= SMITH
응용프로그램 이름	= db2bp
응용프로그램 핸들	= 1
응용프로그램 상태	= 요청 대기
상태 변경 시간	= 12-15-2001 10:29:06.707086
클라이언트 노드	= sys143
클라이언트 릴리스 레벨	= SQL06010
클라이언트 플랫폼	= AIX
클라이언트 프로토콜	= TCP/IP
클라이언트 코드페이지	= 850
클라이언트 응용프로그램의 프로세스 ID	= 49074

클라이언트 로그인 ID	= smith
호스트 응용프로그램 ID	= G9150F74.B6A5.991215152825
시퀀스 번호	= 0000
게이트웨이의 데이터베이스 별명	= MVSDB
DCS 데이터베이스 이름	= DCSDB
호스트 데이터베이스 이름	= GILROY
호스트 릴리스 레벨	= DSN05012
호스트 CCSID	= 500
아웃바운드 통신 주소	= 9.21.21.92 5021
아웃바운드 통신 프로토콜	= TCP/IP
인바운드 통신 주소	= 9.21.15.116 46756
첫 번째 데이터베이스 연결 시간 소인	= 12-15-2001 10:28:24.596495
호스트 응답 시간(sec.ms)	= 0.000000
게이트웨이 처리에 소요된 시간	= 0.000000
최종 재설정 시간소인	=
선택된 행	= 0
시도된 SQL문의 수	= 2
실패한 명령문 조작	= 0
Commit문	= 1
Rollback문	= 0
수신된 인바운드 바이트 수	= 404
송신된 아웃바운드 바이트 수	= 140
수신된 아웃바운드 바이트 수	= 103
송신된 인바운드	= 287
열린 커서의 수	= 0
응용프로그램 유효 시간	= 1 분 32 초
UOW 완료 상태	=
이전의 UOW 완료 시간소인	= 12-15-2001 10:28:25.592631
UOW 시작 시간소인	= 12-15-2001 10:29:06.142790
UOW 중지 시간소인	=
마지막 완료된 uow의 경과시간(sec.ms)	= 0.034396
가장 최근의 조작	= 즉시 실행
가장 최근의 조작 시작 시간소인	= 12-15-2001 10:29:06.142790
가장 최근의 조작 중지 시간소인	= 12-15-2001 10:29:06.707053
명령문	= 즉시 실행
섹션 번호	= 203
응용프로그램 작성자	= NULLID
패키지 이름	= SQLC2C07
SQL 컴파일러 자원 사용 추정치(timeron)	= 0
SQL 컴파일러 카디널리티 추정치	= 0
명령문 시작 시간소인	= 12-15-2001 10:29:06.142790
명령문 중지 시간소인	= 12-15-2001 10:29:06.707053
호스트 응답 시간(sec.ms)	= 1.101612
마지막 완료된 stmt의 경과시간(sec.ms)	= 0.564263
페치된 행	= 0
게이트웨이 처리에 소요된 시간	= 0.013367
명령문에 대해 수신된 인바운드 바이트 수	= 220
명령문에 대해 송신된 아웃바운드 바이트 수	= 130
명령문에 대해 수신된 아웃바운드 바이트 수	= 49
명령문에 대해 수신된 인바운드 바이트 수	= 27
SQL문 텍스트:	
create table t12 (col1 int, col2 char)	

DCS 응용프로그램 상태

시스템 모니터는 다음과 같은 세 가지 형태의 LIST DCS APPLICATIONS 명령을 제공합니다.

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

다음에 나오는 출력에서 호스트 응용프로그램 ID 및 클라이언트 응용프로그램 ID의 형식은 IBM 메인프레임 데이터베이스 버전 및 TCP/IP 지원 레벨에 따라 다를 수 있습니다.

표 11. 호스트 버전 및 TCP/IP 지원 레벨에 기초한 응용프로그램 ID 형식

시나리오	응용프로그램 ID 형식
7 미만의 RDB Manager Level 지원으로 데이터 서버에 액세스하는 클라이언트	G91A0D3A.P8BC.060306212019
TCP/IP v4를 통해 8 이상의 RDB Manager 레벨 지원으로 데이터 서버에 액세스하는 클라이언트	9.26.13.61.65289.060306213816
TCP/IP v6를 통해 8 이상의 RDB Manager 레벨 지원으로 데이터 서버에 액세스하는 클라이언트	2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741

LIST DCS APPLICATIONS

응용프로그램 레벨에서 모니터가 제공하는 정보를 보려면, DB2 LIST DCS APPLICATIONS 명령을 실행하십시오.

TCP/IP 연결(DB2 Connect에서 DB2으로 연결)에 대해 다음 정보를 리턴합니다.

```

권한 ID  응용프로그램 이름 App1.  호스트 응용프로그램 ID
                                           핸들
-----
NEWTON db2cli.exe      7      G91A0D3A.P8BC.060306212019
NEWTON db2cli.exe      25     9.26.13.61.65289.060306213816
NEWTON db2cli.exe      20     2002:91a:519:13:209:6bff:fe14:4fbb.7684.060306213741
  
```

권한 ID

IBM 메인프레임 데이터베이스 서버에 로그인하는 데 사용된 권한 부여 ID. 이는 응용프로그램을 실행 중인 사용자를 식별합니다.

응용프로그램 이름

DB2 Connect에 알려진 대로 클라이언트에서 실행 중인 응용프로그램의 이름. 최종 경로 분리자 다음의 첫 20바이트만이 사용 가능합니다.

Appl. 핸들

DB2 Connect 워크스테이션에서 실행 중인 에이전트. 이 요소를 사용하여 데이터베이스 시스템 모니터 정보를 다른 진단 정보와 연결할 수 있습니다. 에이전트 ID는 또한 FORCE USERS 명령이나 API를 사용할 경우에 필요합니다.

호스트 응용프로그램 ID

다음 중 하나입니다.

- 비보호 대화에 사용되는 DRDA 상관 토큰(CRRTKN).
- DRDA-3 동기점 관리 프로그램(TCP/IP 연결에 사용됨)에 의해 보호되는 2 단계 연결용 작업 단위(UOW) ID(UOWID)

이 고유 ID는 응용프로그램이 IBM 메인프레임 데이터베이스 서버에 연결할 때 생성됩니다. 응용프로그램 ID와 함께 이 요소를 사용하여 응용프로그램 정보의 클라이언트와 서버 부분을 연관시킬 수 있습니다.

LIST DCS APPLICATIONS SHOW DETAIL

DB2 LIST DCS APPLICATIONS SHOW DETAIL 명령 형식이 지정된 경우, 다음의 추가 정보가 표시됩니다.

권한 ID	응용프로그램 이름		Appl. 핸들	클라이언트 응용프로그램 ID	
NEWTON	db2cli.exe		37	2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224	
Seq#	Client DB 별명	Client 노드	Client 릴리스	Client 코드	호스트 응용프로그램 Id 페이지
00001	MDB	SAYYID	SQL09000	1252	G91A0D3A.P982.060306214231
Seq#	호스트 DB 이름		호스트 릴리스		
00001	MEXICO		DSN08015		

클라이언트 응용프로그램 ID

DB2 Connect 워크스테이션에 연결된 응용프로그램을 고유하게 식별합니다. 응용프로그램 ID의 형식은 클라이언트와 DB2 Connect 워크스테이션간의 통신 프로토콜에 따라 다릅니다.

이 값을 사용하면 클라이언트에서 DB2 Connect 워크스테이션으로의 연결 및 DB2 Connect 워크스테이션에서 IBM 메인프레임 데이터베이스 서버로의 연결을 상관시킬 수 있습니다.

클라이언트 시퀀스 번호(Seq#)

클라이언트 시퀀스 번호는 트랜잭션 시퀀스 번호입니다. 이 번호를 사용하여 다른 시스템간의 트랜잭션을 연관하도록 도움을 줍니다.

클라이언트 DB 별명

데이터베이스에 연결하기 위해 응용프로그램에 의해 제공된 데이터베이스 별명입니다. 이 요소를 사용하면 데이터베이스가 액세스하는 실제 데이터베이스를 식별할 수 있습니다. 이 이름과 데이터베이스 이름간의 맵핑은 클라이언트 노드 및 데이터베이스 관리 프로그램 서버 노드의 데이터베이스 디렉토리를 사용하여 이루어질 수 있습니다.

클라이언트 NNAME(노드)

클라이언트 응용프로그램이 실행 중인 노드를 식별합니다. 정보는 사용 중인 클라이언트 프로토콜에 따라 달라집니다. TCP/IP를 통해 연결된 클라이언트에서는 호스트 이름입니다.

클라이언트 제품 ID(클라이언트)

클라이언트에서 실행 중인 제품 및 버전입니다. 클라이언트 제품 ID는 다음과 같습니다.

- DB2 Universal Database와 DB2 Connect 제품 및 해당 클라이언트 버전 7.1의 경우 SQL07010
- DB2 Universal Database와 DB2 Connect 제품 및 해당 클라이언트 버전 8.1의 경우 SQL08010
- DB2 Universal Database 및 DB2 Connect 제품 및 해당 클라이언트 버전 8.2의 경우 SQL08020
- DB2 제품 버전 9.1, DB2 Connect 제품 및 해당 클라이언트의 경우 SQL09120

코드 페이지 ID

모니터된 응용프로그램이 시작한 노드의 코드 페이지 ID입니다.

이 정보를 사용하면 응용프로그램 코드 페이지와 데이터베이스 코드 페이지(또는 IBM 메인프레임 데이터베이스 서버, IBM 메인프레임 데이터베이스 서버 CCSID) 간에 데이터 변환을 지원할 수 있습니다.

응용프로그램 코드 페이지가 데이터베이스 시스템 모니터가 실행 중인 코드 페이지와 다른 경우, 이 코드 페이지 요소는 응용프로그램으로부터 전달되고 데이터베이스 시스템 모니터에 의해 표시되는 데이터를 직접 변환하는 데 도움을 줄 수 있습니다. 예를 들어, 이 요소를 사용하여 응용프로그램 이름을 변환할 수 있습니다.

아웃바운드 시퀀스 번호

이 번호는 아웃바운드 시퀀스 번호를 나타냅니다. 그리고 다른 시스템에서 트랜잭션의 상호 관련을 위해 사용됩니다.

호스트 데이터베이스 이름

응용프로그램이 연결되는 실제 데이터베이스 이름입니다. DCS 디렉토리에서, 이 이름은 목표 데이터베이스 이름입니다.

호스트 제품 ID

서버에서 실행 중인 제품 및 버전입니다. 이것은 다음과 같이 *PPPVVRRM* 형식으로 되어 있습니다.

PPP IBM 메인프레임 데이터베이스 서버 제품(예: z/OS용 DB2의 경우 DSN, VSE & VM용 DB2 서버의 경우 ARI, IBM i용 DB2의 경우 QSQ)을 식별합니다.

VV 08과 같은 두 자리 버전 번호입니다.

RR 01과 같은 두 자리 릴리스 번호입니다.

M 1개 문자 수정 레벨입니다(0-9 또는 A-Z).

LIST DCS APPLICATIONS EXTENDED

EXTENDED 옵션에서 LIST DCS APPLICATIONS 명령을 사용하여 확장 보고서를 생성할 수 있습니다. 확장 보고서에는 명령에 SHOW DETAIL 옵션이 지정된 경우 나열되는 모든 필드와 아래와 같은 9개의 새 필드가 나열됩니다.

- DCS 응용프로그램 상태
- 상태 변경 시간
- 클라이언트 플랫폼
- 클라이언트 프로토콜
- 호스트의 코드화된 문자 세트 ID(CCSID)
- 클라이언트 로그인 ID
- 클라이언트 응용프로그램의 프로세스 ID
- 게이트웨이에서의 데이터베이스 별명
- DCS 데이터베이스 이름

기존 명령 옵션은 응용프로그램당 한 행씩 필드를 수평으로 나열하는 반면, 새 옵션은 한 행에 한 필드씩 수직으로 나열합니다.

다음은 새 명령 구문입니다.

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

그리고 다음은 새 옵션 EXTENDED를 사용한 명령의 샘플 출력입니다.

DCS 응용프로그램 목록 - 확장 보고서

```
클라이언트 응용프로그램 ID           =
2002:91a:519:13:209:6bff:fe14:4fbb.8196.060306214224
시퀀스 번호                             = 00001
권한 부여 ID                           = NEWTON
트러스트된 권한 부여 ID               =
응용프로그램 이름                     = db2cli.exe
응용프로그램 핸들                     = 37
응용프로그램 상태                     = 요청 대기
상태 변경 시간                         = Not Collected
클라이언트 노드                       = SAYYID
클라이언트 릴리스 레벨                 = SQL09000
클라이언트 플랫폼                     = NT
클라이언트 프로토콜                   = TCP/IP
클라이언트 코드 페이지                 = 1252
클라이언트 응용프로그램의 프로세스 ID = 1192
클라이언트 로그인 ID                  = ISAYYID
호스트 응용프로그램 ID                 = G91A0D3A.P982.060306214231
시퀀스 번호                             = 00001
게이트웨이에서의 데이터베이스 별명    = MDB
DCS 데이터베이스 이름                  = MDB
호스트 데이터베이스 이름               = MEXICO
호스트 릴리스 레벨                     = DSN08015
호스트 CCSID                           = 1208
```

응용프로그램 상태 필드에는 다음 세 값 중 하나가 들어 있습니다.

1. 연결 보류 중 - 아웃바운드. IBM 메인프레임 데이터베이스에 대한 연결 요청이 발행되었으며 DB2 Connect가 연결될 때까지 기다리고 있음을 의미합니다.
2. 요청 대기. IBM 메인프레임 데이터베이스와 연결되었으며 DB2 Connect가 클라이언트 응용프로그램으로부터 SQL문을 기다리고 있음을 의미합니다.
3. 응답 대기. SQL문이 IBM 메인프레임 데이터베이스로 전송되었음을 의미합니다.

또한 상태 변경 시간은 시스템 모니터 UOW 스위치가 처리하는 동안 작동된 경우에만 보고서에 나타납니다. 그렇지 않으면, "수집되지 않음"이 나타납니다.

Health Monitor 및 경보

DB2 Health Monitor는 오브젝트 유지보수 규정을 주기적으로 평가합니다. Health Monitor가 오브젝트의 유지보수가 필요한 것으로 판단하면 Health 경보가 작성됩니다. Health 경보에 대응하여 취해야 할 조치를 보고 실행을 위해 제출한 다음 저장할 수 있습니다.

z/OS용 DB2 Health Monitor 개요

z/OS 시스템의 경우 z/OS용 DB2 Health Monitor는 모니터링할 각 DB2 서브시스템의 태스크로서 또는 데이터 공유 그룹의 전용 구성원에 대해 시작됩니다.

z/OS용 DB2 Health Monitor는 규정에 정의된 스케줄된 시간 및 간격에 따라 오브젝트 유지보수 규정 평가를 트리거합니다. 오브젝트 유지보수 규정은 DB2 제어 센터의 오브젝트 유지보수 규정 작성 마법사를 사용하여 작성됩니다. 각 규정 평가 중 오브젝트 유지보수 규정에 설정된 임계값을 유지보수 권장 기준에 따라 점검하여 오브젝트 유지보수 필요성(예를 들어, COPY, REORG, RUNSTATS, STOSPACE, ALTER TABLESPACE 또는 ALTER INDEX가 필요한지 여부)을 판별하고 해당 테이블 스페이스, 인덱스 및 스토리지 그룹 오브젝트의 제한된 상태(예: CHKP)를 식별합니다. 규정 평가 중 오브젝트가 경보 상태인 것으로 식별되면 해당 전자 우편 주소 또는 호출기 번호로 규정 Health 정보 문의처에 알려줍니다. 각 DB2 서브시스템의 Health 정보 문의처 목록은 제어 센터에서 정의되고 관리됩니다.

Health Monitor가 규정 평가 트리거 시점을 판별하기 위해 사용하는 규정 평가 스케줄 스냅샷은 Health Monitor를 시작할 때 처음 캡처됩니다. 이 스케줄 스냅샷은 Health Monitor 시작 시 또는 Health Monitor가 새로 고침 명령을 받을 때 지정된 새로 고침 시간에 새로 고쳐집니다. 규정 평가 스케줄 변경사항은 스케줄을 새로 고칠 때 Health Monitor에 반영됩니다.

Health Monitor는 각각 MVS 시스템 START 및 STOP 명령을 사용하여 콘솔에서 시작되고 중지됩니다.

DB2 Health Monitor를 시작하는 샘플 카탈로그 프로시저(DSNHMONP)와 MVS 시스템 또는 Parallel Sysplex®에서 여러 DB2 Health Monitor를 시작하는 샘플 카탈로그 프로시저(DSNHMONA)는 설치 작업 DSNTIJHM으로 프로시저 라이브러리에 배치됩니다.

db2 Health Monitor 또는 아래 나열된 관련 태스크가 사용하는 뷰, 테이블, 데이터 세트, 카탈로그 프로시저, 스토어드 프로시저, 사용자 정의 함수 및 결과 세트 테이블은 설치 작업 DSNTIJCC 및 DSNTIJHM으로 작성되고 설치됩니다. DSNTIJCC 및 DSNTIJHM은 FMIDs JDB771D 및 JDB881D와 함께 제공됩니다.

규정 평가 로그

DB2 Health Monitor가 트리거한 규정 평가는 DSNACC.HM_EVAL_LOG 테이블에 로그됩니다. 규정 평가가 시작되고 규정 평가가 종료될 때 항목이 로그됩니다. 로그 항목은 7일 간 보존된 후 테이블에서 삭제됩니다. DB2 뷰 DSNACC.HM_ALERT_PO_EV(DSNTIJCC 설치 작업으로 이 테이블에서 작성)는 마지막 평가 반복이 성공하지 못한 모든 규정을 표시하는 데 사용됩니다.

z/OS용 DB2 Health Monitor 시작, 중지 및 새로 고침

z/OS 시스템의 경우 z/OS용 DB2 Health Monitor는 모니터링 각 DB2 서브시스템의 태스크로서 또는 데이터 공유 그룹의 전용 구성원에 대해 시작됩니다.

- DB2 Health Monitor를 시작하려면 다음 START MVS 시스템 명령을 실행하십시오.

S membername, DB2SSN=ssid, JOBNAME=HMONssid, TRACE=trace, REFRESH=nn

TRACE 및 REFRESH 매개변수는 선택사항입니다.

membername

DB2 Health Monitor를 시작하기 위해 실행되는 프로시저 라이브러리 구성원 (DSNHMONP)을 지정합니다. 이 카탈로그된 프로시저는 DSNTIJHM 설치 작업으로 작성됩니다.

ssid

모니터할 DB2 서브시스템의 이름 또는 ID를 지정합니다.

trace

추적 플래그를 지정합니다. 가능한 값은 다음과 같습니다.

- ON - 추적을 켭니다. 추적 레코드는 SYSOUT에 기록됩니다.
- OFF - 추적을 켜지 않습니다.

디폴트값은 OFF입니다.

nn

Health Monitor가 규정 평가를 트리거하기 위해 사용하는 평가 스케줄 스냅샷을 새로 고치는 시간(24시간 시계 사용)을 지정합니다. 디폴트값은 22입니다.

- 여러 DB2 Health Monitor를 시작하려면 다음 START MVS 시스템 명령을 실행하십시오.

S membername

membername

여러 DB2 Health Monitor를 시작하기 위해 실행되는 프로시저 라이브러리 구성원 (DSNHMONA)입니다.

주: 하나의 START 명령에 DSNHMONA를 사용하여 여러 DB2 Health Monitor를 시작하기 전에 DSNHMONA 프로시저에 지정된 HMONPARM 데이터 세트를 모니터할 서브시스템의 목록으로 채워야 합니다. 카탈로그된 프로시저와 데이터 세트는 DSNTIJHM 설치 작업으로 작성됩니다.

- DB2 Health Monitor가 규정 평가 트리거 시점을 판별하기 위해 사용하는 규정 평가 스케줄 스냅샷을 새로 고치려면 다음 MODIFY MVS 시스템 명령을 실행하십시오.

F HMONssid,APPL=REFRESH

ssid

현재 새로 고치는 DB2 Health Monitor가 모니터하는 DB2 서브시스템의 이름 또는 ID입니다.

- DB2 Health Monitor를 중지하려면 다음 STOP MVS 시스템 명령을 실행하십시오.

STOP HMONssid or P HMONssid

ssid

현재 중지하는 DB2 Health Monitor가 모니터하는 DB2 서브시스템의 이름 또는 ID입니다.

권장 조치의 보기, 제출 및 저장

규정 평가 중 식별된 경보 오브젝트에 대해 권장되는 조치를 보고 제출 및 저장하려면 DB2 스토어드 프로시저 SYSPROC.DSNACCHR을 호출하십시오. 이 프로시저는 DSNTIJCC 설치 작업으로 작성됩니다. DSNACCHR은 규정 평가 중 식별되는 경보 오브젝트에 대해 권장되는 조치를 판별하고 권장 조치를 실행할 JCL 작업을 생성하는 스토어드 프로시저입니다.

다음 구문 다이어그램은 DSNACCHR을 호출하는 SQL CALL문을 보여줍니다. DSNACCHR에 대한 링크 규칙은 GENERAL WITH NULLS이므로 호스트 변수의 매개변수를 전달하는 경우 모든 호스트 변수와 널(NULL) 표시기를 포함해야 합니다. CALL문을 실행하기 전에 입력 호스트 변수의 널(NULL) 표시기를 초기화해야 합니다.

구문

```

▶—CALL DSNACCHR (—query-type,—health-ind,—policy-id,—work-set,——————▶
▶┌dataset-name,┌member-name,┌save-opt,┌trace-flag,—————▶
▶└NULL┘└NULL┘└NULL┘
▶—job-id,—jobname,—jcl-proc-time,—trace-flag,—last-statement,—————▶
▶—return-code,—error-msg )—————▶▶

```

query-type

규정 평가시 경보 상태로 식별되는 오브젝트에 권장되는 조치에 대처하는 방법을 지정합니다. 가능한 값은 다음과 같습니다.

- 0 - 경보 오브젝트에 대한 권장 조치를 JCL 작업으로 보기
- 1 - 경보 오브젝트에 대한 권장 조치를 실행하는 JCL 작업 제출

- 2 - 경보 오브젝트에 대한 권장 조치를 실행하는 JCL 작업 제출 및 작업을 대기 큐에 지정
 - 3 경보 오브젝트에 대한 권장 조치를 라이브러리 구성원의 JCL 작업으로 저장
- query-type*은 INTEGER 유형의 입력 매개변수입니다.

health-ind

DSNACCHR이 JCL 작업에 포함하는 경보 유형을 지정합니다. 가능한 값은 다음과 같습니다.

- RS - 제한된 상태
- EX - Extent 초과
- RR - REORG 필수
- CR - COPY 필수
- RT - RUNSTATS 필수
- SS - STOSPACE 필수

*health-ind*는 VARCHAR(4) 유형의 입력 매개변수입니다.

policy-id

오브젝트 유지보수 규정을 지정합니다. *policy-id*는 VARCHAR(7) 유형의 입력 매개변수입니다.

work-set

DSNACCHR이 JCL 작업에 포함하는 경보 오브젝트를 식별한 오브젝트 유지보수 규정의 작업 세트를 지정합니다. 이 작업 세트는 *policy-id* 및 *health-ind* 매개변수에 지정된 경보 유형 및 규정으로 식별해야 합니다. *work-set*는 INTEGER 유형의 입력 매개변수입니다.

dataset-name

완전한 파티션된 데이터 세트(PDS) 또는 파티션된 데이터 세트 확장(PDSE) 이름을 지정합니다. 이 값은 *query-type*이 3인 경우 지정해야 합니다. *dataset-name*은 VARCHAR(44) 유형의 입력 매개변수입니다.

member-name

오브젝트 유지보수 JCL 작업이 저장될 *dataset-name* 매개변수에 지정된 여러 파티션된 데이터 세트(PDS) 또는 파티션된 데이터 세트 확장(PDSE) 구성원을 지정합니다. 이 값은 *query-type*이 3인 경우 지정해야 합니다. *member-name*은 VARCHAR(8) 유형의 입력 매개변수입니다.

save-opt

오브젝트 유지보수 JCL 작업 저장 방법을 지정합니다. 이 값은 *query-type*이 3인 경우 지정해야 합니다. 가능한 값은 다음과 같습니다.

- R - 바꾸기
- A - 추가
- NM - 새 구성원

*save-opt*는 VARCHAR(2) 유형의 입력 매개변수입니다.

trace-flag

추적을 켜거나 끌지 여부를 지정합니다. 가능한 값은 다음과 같습니다.

- Y - 추적 켜기
- N - 추적 끄기

*trace-flag*는 CHAR(1) 유형의 입력 매개변수입니다.

job-ID

*query-type*이 1 또는 2이면 제출된 작업의 작업 ID를 지정합니다. *job-id*는 VARCHAR(8) 유형의 출력 매개변수입니다.

jobname

*query-type*이 1 또는 2이면 제출된 작업의 이름을 지정합니다. *jobname*은 VARCHAR(8) 유형의 출력 매개변수입니다.

jcl-proc-time

요청 처리 시간을 지정합니다. *jcl-proc-time*은 TIMESTAMP 유형의 출력 매개변수입니다.

last-statement

DSNACCHR이 심각한 오류(리턴 코드 12)를 리턴하는 경우 이 필드에는 오류 발생 시 실행된 SQL문이 포함됩니다. *last-statement*는 VARCHAR(2500) 유형의 출력 매개변수입니다.

return-code

DSNACCHR 실행의 리턴 코드입니다. 가능한 값은 다음과 같습니다.

- 0 - DSNACCHR 실행 성공

- 12 - DSNACCHR 종료되었으나 심각한 오류. *error-msg* 매개변수에는 오류에 대해 설명하는 메시지가 포함됩니다. *last-statement* 매개변수에는 오류 발생시 실행된 SQL문이 포함됩니다.

*return-code*는 INTEGER 유형의 출력 매개변수입니다.

error-msg

DSNACCHR이 심각한 오류(리턴 코드 12)를 리턴하는 경우 이 필드에는 오류 메시지 (예: 형식화된 SQLCA)가 포함됩니다. *error-msg*는 VARCHAR(1331) 유형의 출력 매개변수입니다.

query-type 매개변수가 0이면 DSNACCHR이 하나의 결과 세트를 리턴합니다. 결과 세트에는 DSNACCHR로 생성된 JCL 작업이 포함됩니다. DSNACCHR 결과 세트 테이블은 DSNTIJCC 설치 작업으로 작성됩니다. 표 12는 결과 세트 형식입니다.

표 12. DSNACCHR 결과 세트 형식

컬럼 이름	데이터 유형	설명
JCLSEQNO	INTEGER	테이블 행의 시퀀스 번호 (1,...,n)
JCLSTMT	VARCHAR(80)	JCL문을 지정합니다.

상태 경보 요약 보기

HEALTH_OVERVIEW 함수는 Health 경보 요약 VSAM KSDS 데이터 세트의 정보를 DB2 테이블로 리턴합니다. 이 데이터 세트는 DSNTIJHM 설치 작업으로 작성됩니다.

Health 경보 요약 데이터 세트에는 해당 MVS 시스템 또는 병렬 Sysplex의 Health Monitor가 이전에 모니터링했거나 현재 모니터링하는 모든 DB2 서브시스템의 DB2 Health Monitor 및 경보 요약 통계 상태에 대한 정보가 포함됩니다. 이 정보는 클라이언트에게 각 DB2 서브시스템 및 경보 권장사항의 경과 함께 리턴됩니다.

함수 결과는 다음 컬럼이 포함된 DB2 테이블입니다.

ip-addr

DB2 서버의 IP 주소입니다. VARCHAR(40) 유형의 컬럼입니다.

db2-ssid

DB2 서브시스템의 서브시스템 ID. VARCHAR(4) 유형의 컬럼입니다.

health-ind

경보 유형입니다. 가능한 값은 다음과 같습니다.

- RS - 제한된 상태
- EX - Extent 초과
- RR - REORG 필수
- CR - COPY 필수
- RT - RUNSTATS 필수
- SS - STOSPACE 필수
- PO - 실패한 규정 평가
- HM - Health Monitor 상태

*health-ind*는 유형 VARCHAR(4)의 컬럼입니다.

host-name

DB2 서버의 완전한 도메인 이름입니다. VARCHAR(255) 유형의 컬럼입니다.

summary-stats

*health-ind*가 'HM'인 경우 DB2 Health Monitor의 상태입니다. 가능한 값은 다음과 같습니다.

- 0 Health Monitor가 시작되지 않음
- 1 Health Monitor가 시작됨
- -1 Health Monitor 상태를 알 수 없음

그렇지 않으면 *health-ind*에 지정된 경보 유형의 총 경보 오브젝트 수입니다. INTEGER 유형의 컬럼입니다.

alert-state

*health-ind*에 지정된 경보의 상태입니다. 가능한 값은 다음과 같습니다.

- 5 - 알람
- 4 - 주의
- 3 - 경고
- 0 - 정상

*health-ind*가 'HM'인 경우 *alert-state*는 항상 0입니다. INTEGER 유형의 컬럼입니다.

함수의 외부 프로그램 이름은 HEALTH_OVERVIEW이고 특정 이름은 DSNACC.DSNACCHO입니다. 이 함수는 DSNTIJCC 설치 작업으로 작성됩니다.

예: DB2 서브시스템 'ABCD'의 COPY가 필요한 총 경보 오브젝트 수를 찾으십시오.

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

상태 경고 오브젝트 보기

규정 평가를 마지막으로 반복하여 성공할 때 식별되는 경고 오브젝트는 해당 오브젝트 유형에 따라 이 경고 오브젝트 저장소 테이블에 저장됩니다.

경고 오브젝트는 다음과 같습니다.

- DSNACC.HM_MAINT_TS(테이블 스페이스)
- DSNACC.HM_MAINT_IX(인덱스)
- DSNACC.HM_MAINT_SG(스토리지 그룹)

DB2는 이 경고 오브젝트 저장소 테이블에 여러 개의 뷰를 작성합니다. 뷰 및 경고 오브젝트 저장소 테이블은 DSNTIJCC 설치 작업으로 작성됩니다. 표 13에는 각 뷰가 정의된 테이블과 뷰 설명이 나열됩니다. 모든 뷰 이름과 테이블 이름의 규격자는 DSNACC입니다.

표 13. Health 경고 오브젝트 보기

뷰 이름	테이블	뷰 설명
HM_ALERT_TS_RS	HM_MAINT_TS	제한 상태의 모든 테이블 스페이스 표시
HM_ALERT_TS_EX	HM_MAINT_TS	Extent가 사용자 지정 한계를 초과한 모든 테이블 스페이스 표시
HM_ALERT_TS_RR	HM_MAINT_TS	REORG가 필요한 모든 테이블 스페이스 표시
HM_ALERT_TS_CR	HM_MAINT_TS	COPY가 필요한 모든 테이블 스페이스 표시
HM_ALERT_TS_RT	HM_MAINT_TS	RUNSTATS가 필요한 모든 테이블 스페이스 표시
HM_ALERT_IX_RS	HM_MAINT_IX	제한 상태의 모든 인덱스 표시
HM_ALERT_IX_EX	HM_MAINT_IX	Extent가 사용자 지정 한계를 초과한 모든 인덱스 표시
HM_ALERT_IX_RR	HM_MAINT_IX	REORG가 필요한 모든 인덱스 스페이스 표시
HM_ALERT_IX_CR	HM_MAINT_IX	COPY가 필요한 모든 인덱스 표시
HM_ALERT_IX_RT	HM_MAINT_IX	RUNSTATS가 필요한 모든 인덱스 표시
HM_ALERT_SG_SS	HM_MAINT_SG	STOSPACE가 필요한 모든 스토리지 그룹 표시

제 3 부 고가용성 및 DB2 Connect

DB2 Connect를 사용하는 환경에서는 고가용성과 관련된 특정 고려사항이 있습니다. 어떠한 이유로 네트워크의 데이터베이스 서버를 사용할 수 없게 된 경우 클라이언트 워크스테이션을 네트워크의 대체 데이터베이스 서버에 리라우트하는 기능이 중요합니다.

제 11 장 호스트 데이터베이스 연결을 위한 고가용성 및 로드 밸런스

오늘날의 정보 기술 시장에서는 하루 24시간 데이터를 사용하기 위한 요구사항이 매우 높습니다. 경쟁업체와의 비즈니스에서 경쟁하고 지속적인 성장을 유지하려면 이러한 요구사항이 충족되어야 합니다. 현재 많은 웹, e-business 및 스프레드시트 응용프로그램에서 기업의 데이터에 액세스합니다. IBM 메인프레임 데이터베이스에 대해 믿을 수 있고 빠르고 안전한 연결을 설정해야 합니다. 이 연결은 계속 사용 가능해야 하며 중요한 로드 조건에서도 많은 연결 요청을 처리할 수 있어야 합니다. 이러한 연결을 구축하기 위한 방법에 대해 알아보겠습니다.

사용 가능성을 높이기 위한 시나리오

Windows, Linux 및 UNIX에서 실행되는 여러 대의 워크스테이션 및 응용프로그램 서버가 있는 경우를 생각해보겠습니다. 이러한 머신은 여러 IBM 메인프레임 데이터베이스에 있는 데이터에 액세스해야 합니다. 이러한 머신에서 실행되는 응용프로그램에서는 데이터베이스로의 빠르고 안정적인 연결을 요구합니다. 시스템 전체는 TCP/IP를 사용하는 이더넷 네트워크에 의해 연결됩니다.

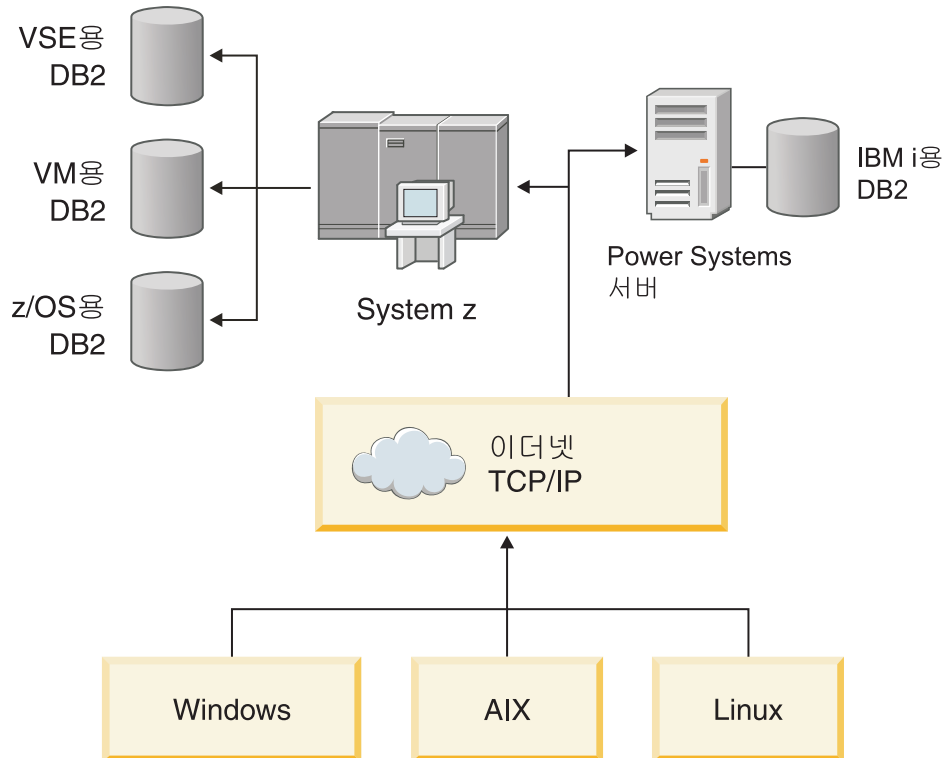


그림 11. 샘플 네트워크 시나리오

워크스테이션 및 응용프로그램 서버가 IBM 메인프레임 데이터베이스에 액세스하려면 중개자로 연결 구성요소가 필요합니다. 이 구성요소는 IBM 메인프레임 데이터베이스에 대한 고가용성의 강력하고 빠른 연결을 제공해야 합니다. 또한 연결 볼륨의 향후 성장을 예상하여 확장할 수 있어야 합니다.

이 주제에서의 관련 링크를 사용하여 DB2 Connect 및 자동 클라이언트 리라우트 기능을 사용하는 솔루션을 참조하십시오.

제 12 장 자동 클라이언트 리라우트 설명 및 설정(DB2 Connect)

자동 클라이언트 리라우트 기능의 주 목적은 IBM Data Server Client 응용프로그램이 최소한의 인터럽트로 계속 작업할 수 있도록 통신 실패로부터 복구할 수 있도록 하는 것입니다. 이름에서 알 수 있듯이 리라우트가 연속적인 조작 지원의 핵심입니다. 그러나 클라이언트 연결에 대해 식별되는 대체 위치가 있는 경우에만 리라우트가 가능합니다. 비DB2 Connect 고가용성 환경에서, 액세스되고 있는 데이터베이스는 일반적으로 다양한 방법(예: 고가용성 클러스터 멀티프로세서(HACMP™) 또는 고가용성 재해 복구(HADR)) 중 하나를 사용하여 기존 DB2 서버와 대체 DB2 서버 사이에서 동기화됩니다.

그러나 DB2 Connect 서버의 경우 로컬 데이터베이스 동기화에 대한 요구사항이 없으므로, 기존 및 대체 DB2 Connect 서버에 모두 동일한 데이터베이스 별명을 사용하여 액세스할 수 있는 방법으로 카탈로그된 목표 IBM 메인프레임 데이터베이스가 있는지 확인해야만 합니다.

주: DB2 Connect 서버 환경에서 대체 DB2 Connect 서버는 클라이언트와 DB2 Connect 서버 간에 자동 리라우트를 사용하도록 지정할 수 있습니다. DB2 Connect 개인용 또는 서버 제품과 IBM 메인프레임 데이터베이스 서버 간에 발생하는 리라우트의 경우, 리모트 서버가 대체 주소를 제공해야 합니다. DB2의 경우, 데이터베이스가 Sysplex 데이터 공유 환경에 있으면 여러 개의 주소를 알 수 있습니다.

Sysplex 지원이 사용되는 경우 DB2 Connect 및 호스트 데이터베이스 서버 사이에 Sysplex의 리라우트 기능을 구성할 수 있습니다. Sysplex의 리라우트 기능은 DB2 Connect가 Sysplex 그룹의 원래 구성원과의 통신이 유실된 후 다른 구성원에 대해 연결을 재시도하도록 허용하는 DB2 Connect 기능입니다. DB2 Connect에서 Sysplex의 리라우트 기능을 사용할 수 있도록 하기 위해 대체 서버를 데이터베이스 디렉토리에 카탈로그화할 필요는 없습니다. 기본적으로 Sysplex의 리라우트 기능은 Sysplex 지원이 사용될 경우에 사용될 수 있습니다.

IBM Data Server Client가 통신이 유실된 경우 자동 클라이언트 리라우트를 사용하여 DB2 Connect 서버로 복구할 수 있으려면 대체 DB2 Connect 서버 위치는 통신이 유실되기 전에 지정되어야 합니다. UPDATE ALTERNATE SERVER FOR DATABASE 명령은 특정 IBM 메인프레임 데이터베이스에 대한 대체 DB2 Connect 서버 위치를 정의하는 데 사용됩니다. 대체 호스트 이름 및 포트 번호는 명령의 일부로 지정됩니다. 위치는 DB2 Connect 서버의 시스템 데이터베이스 디렉토리 파일에 저장됩니다. 지정된 대체 DB2 Connect 서버 위치가 모든 클라이언트의 데이터베이스에 적용되는지 확인하려면, 대체 서버 위치는 DB2 Connect 서버측에 지정되어야 합니다. 클라이언트 인스턴스에서 설정될 경우 대체 서버가 무시됩니다.

예를 들어, DB2 Connect 서버 S1(호스트 이름이 db2conn1이고 포트 번호가 122)에서 데이터베이스 별명 db1을 사용하여 IBM 메인프레임 데이터베이스가 카탈로그된다고 가정하십시오. 데이터베이스 관리자는 포트 번호가 123인 호스트 이름 db2conn2에서 대체 DB2 Connect 서버 S2를 지정하려고 합니다. DB2 Connect 서버 S1에서 데이터베이스 관리자가 실행할 명령은 다음과 같습니다.

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

DB2 Connect 서버 S1에서 데이터베이스 별명 db1의 대체 DB2 Connect 서버 위치를 지정한 후에는 대체 서버 위치가 연결 프로세스의 일부로 IBM Data Server Client에 리턴됩니다. IBM Data Server Client와 DB2 Connect 서버 S1 사이의 통신이 어떤 이유(일반적으로 SQL 코드 -30081 또는 SQL 코드 -1224 같은 통신 오류)로 인해 유실되면 IBM Data Server Client에서는 기존 DB2 Connect 서버(S1) 또는 대체 DB2 Connect 서버(S2)를 통해 두 서버 사이의 통신을 교대로 수행하며 db1에 재연결하려고 합니다. 시도 사이의 시간 간격은 신속하게 시작한 후 각 시도에서 점차적으로 늘립니다.

연결이 완료되면, SQL 코드 -30108은 통신 실패 후 데이터베이스 연결이 재설정되었음을 나타내기 위해 리턴됩니다. 호스트 이름 또는 IP 주소 및 서비스 이름 또는 포트 번호가 리턴됩니다. IBM Data Server Client는 클라이언트 통신의 재설정이 원래 서버나 대체 서버에 대해 불가능한 경우 응용프로그램에 원래의 통신 실패에 대한 오류만 리턴합니다.

DB2 Connect 서버 환경에서의 대체 서버 연결성을 포함하는 다음 고려사항에도 주의해야 합니다.

- 리모트 클라이언트 및 로컬 클라이언트 대신에 IBM 메인프레임 데이터베이스에 액세스하기 위해 DB2 Connect 서버를 사용할 때, 시스템 데이터베이스 디렉토리 항목의 대체 서버 연결 정보가 혼동될 수 있습니다. 이러한 혼동을 최소화하려면, 시스템 데이터베이스 디렉토리에서 두 항목의 카탈로그를 고려하여 동일한 IBM 메인프레임 데이터베이스를 표시하십시오. 리모트 클라이언트와 로컬 클라이언트에 대해 각각 한 항목을 카탈로그하십시오.
- 목표 DB2 서버에서 리턴된 SYSPLEX 정보는 DB2 Connect 서버의 캐시에서만 보존됩니다. 하나의 대체 서버만 디스크에 기록됩니다. 대체 또는 활성 서버가 여러 개 있는 경우 해당 정보는 메모리에만 유지되므로 프로세스가 종료될 때 손실됩니다.

제 13 장 클라이언트 연결 분배기 기술에 대한 자동 클라이언트 리라우트 구성

WebSphere EdgeServer 같은 분배기 또는 디스패처 기술은 기본 데이터베이스 서버가 실패하는 경우 클라이언트 응용프로그램 재연결 요청을 정의된 시스템 세트에 분배합니다. DB2 자동 클라이언트 리라우트와 함께 분배기 기술을 사용 중인 경우 분배기 자체를 DB2 자동 클라이언트 리라우트에 대한 대체 서버로 식별해야 합니다.

다음과 비슷한 환경에서 분배기 기술을 사용 중일 수 있습니다.

클라이언트 → 분배기 기술 → (DB2 Connect 서버 1 또는 DB2 Connect 서버 2) → DB2 z/OS

각 부분의 설명은 다음과 같습니다.

- 분배기 기술 구성요소는 DThostname의 TCP/IP 호스트 이름을 갖습니다.
- DB2 Connect 서버 1은 GWYhostname1의 TCP/IP 호스트 이름을 갖습니다.
- DB2 Connect 서버 2는 GWYhostname2의 TCP/IP 호스트 이름을 갖습니다.
- DB2 z/OS 서버는 zOShostname의 TCP/IP 호스트 이름을 갖습니다.

클라이언트는 분배기 기술을 이용하여 DB2 Connect 서버 중 하나에 액세스하기 위해 **DThostname**을 사용하여 카탈로그됩니다. 중간 분배기 기술이 **GWYhostname1** 또는 **GWYhostname2**를 사용할 것을 결정합니다. 결정한 후에 클라이언트는 이러한 두 DB2 Connect 게이트웨이 중 하나에 직접 소켓 연결됩니다. 소켓 연결성이 선택된 DB2 Connect 서버에 설정된 후에, 일반 클라이언트 대 DB2 Connect 서버 대 DB2 z/OS 연결성을 갖습니다.

예를 들어 분배기가 **GWYhostname2**를 선택한다고 가정하십시오. 이것은 다음 환경을 생성합니다.

클라이언트 → DB2 Connect 서버 2 → DB2 z/OS

분배기는 통신 실패가 있는 경우 어떤 연결도 재시도하지 않습니다. 그러한 환경에 있는 데이터베이스에 대해 자동 클라이언트 리라우트 기능을 사용하려는 경우, DB2 Connect 서버에 있는 연관된 데이터베이스에 대한 대체 서버(DB2 Connect 서버 1 또는 DB2 Connect 서버 2)가 분배기(DThostname)가 되도록 설정되어야 합니다. 그런 다음 DB2 Connect 서버 1이 어떤 이유로 잠기는 경우, 자동 클라이언트 리라우트가 트리거되고 클라이언트 연결은 기본 및 대체 서버 둘 다로서의 분배기를 사용하여 재시도됩니다. 이 옵션을 사용하면 분배기 기능을 DB2 자동 클라이언트 리라우트 기능과 결합하고 유지보수할 수 있습니다. 대체 서버를 분배기 호스트 이름이 아닌 호스트로 설

정해도 클라이언트에 자동 클라이언트 리라우트 기능을 제공합니다. 그러나 클라이언트는 정의된 대체 서버로의 직접 연결을 설정하고 분배기 기술을 생략하며, 이는 분배기 및 분배기로 인한 값을 제거합니다.

자동 클라이언트 리라우트 기능이 다음 SQL 코드를 가로챍니다.

- sqlcode -20157
- sqlcode -1768(이유 코드 = 7)

주: "TCP Keepalive" 운영 체제 구성 매개변수의 설정이 너무 높게 설정되면 소켓 실패가 클라이언트 리라우트에 시기 적절하게 알려지지 않을 수 있습니다. (이 구성 매개변수의 이름은 플랫폼에 따라 다름을 주의하십시오.)

제 4 부 성능 조정 및 DB2 Connect

데이터베이스 요청을 이동할 DB2 Connect를 사용하는 데이터베이스 환경 및 클라이언트 워크스테이션과 데이터베이스 서버 간의 응답에 문제가 있으면 이를 해결할 때 특별히 고려해야 할 사항이 있습니다. 이 환경에서 성능을 향상시키거나 유지보수하는 여러 가지 방법이 있습니다.

제 14 장 DB2 Connect 성능 고려사항

성능은 컴퓨터 시스템이 주어진 특정 워크로드를 수행하는 능력입니다. 이는 사용 가능한 자원의 영향을 받고 사용 및 공유된 방식의 영향을 받습니다. 성능을 향상시키려면, 성능에 필요한 사항을 먼저 결정해야 합니다.

다음과 같은 다른 많은 성능 메트릭을 선택할 수 있습니다.

응답 시간

응용프로그램이 데이터베이스 요청을 송신하는 시간과 응용프로그램이 응답을 수신하는 시간 사이의 간격

트랜잭션 처리량

시간 단위마다 완료되는 작업 단위(UOW)의 수. 작업 단위(UOW)는 행 폐치와 갱신처럼 단순할 수도 있으며, 수백 개의 SQL문을 포함하여 복잡할 수도 있습니다.

데이터 전송률

시간 단위당 DB2 Connect 응용프로그램과 IBM 메인프레임 데이터베이스 간에 전송되는 데이터 바이트 수

성능은 사용 가능한 하드웨어와 소프트웨어 자원의 영향을 받습니다. CPU, 메모리 및 네트워크 어댑터가 하드웨어 자원의 예입니다. 소프트웨어 자원의 예로는 통신 서브시스템, 페이징 서브시스템, AIX용 mbuf가 있습니다.

데이터 플로우

106 페이지의 그림 12는 IBM 메인프레임 데이터베이스 서버와 워크스테이션 간에 DB2 Connect를 통해 이동하는 데이터 경로를 나타냅니다.

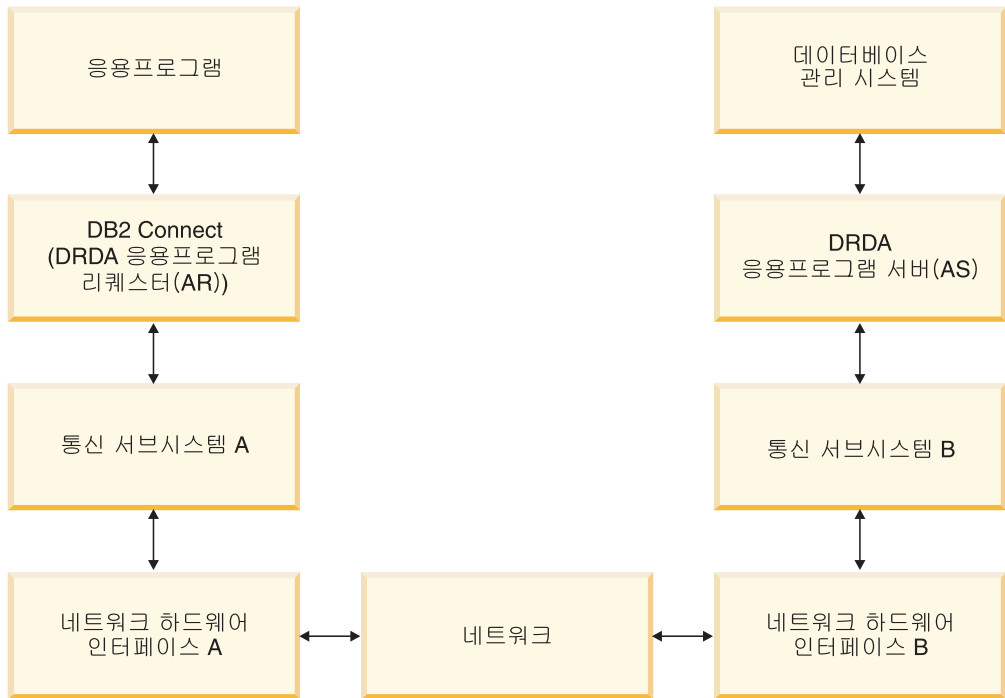


그림 12. DB2 Connect의 데이터 플로우

- IBM 메인프레임 데이터베이스 및 통신 서브시스템 B의 일부는 대개 동일한 시스템에서 실행됩니다. 이 시스템은 하나 이상의 CPU, 기본 스토리지, I/O 서브시스템, DASD, 운영 체제로 이루어져 있습니다. 다른 프로그램이 이러한 구성요소를 공유하기 때문에 자원 경합으로 인해 성능 문제가 발생할 수 있습니다.
- 네트워크는 케이블, 허브, 통신 회선, 스위치, 기타 통신 제어기의 결합으로 이루어져 있습니다. 예를 들어, 네트워크 하드웨어 인터페이스 B는 통신 제어기(예: 3745 또는 3172)이거나 IBM Power Systems 서버의 토큰링 어댑터일 수 있습니다. 네트워크 하드웨어 A와 B 사이에는 하나 이상의 전송 미디어가 있습니다.
- 네트워크 하드웨어 인터페이스 A는 토큰 링, 이더넷**, 기타 LAN 어댑터 또는 SDLC나 X.25 프로토콜을 지원하는 어댑터가 될 수 있습니다.
- DB2 Connect 및 통신 서브시스템 A는 일반적으로 같은 시스템에 있습니다. 여기에서는 응용프로그램도 동일한 시스템에 있는 것으로 가정합니다.

병목 현상

트랜잭션 처리량은 시스템의 가장 속도가 느린 구성요소에 따라 결정됩니다. 성능 병목 현상을 식별하는 경우, 구성 매개변수를 변경하거나 문제되는 구성요소에 더 많은 자원을 할당하거나 구성요소를 업그레이드하거나 새 구성요소를 추가하여 일부 작업을 로드 해제함으로써 문제를 완화시킬 수 있습니다.

다양한 도구를 사용하여 구성요소마다 걸리는 쿼리 시간을 결정할 수 있습니다. 성능을 향상하기 위해 조정되고 업그레이드된 구성요소를 알 수 있습니다. 예를 들어, 쿼리에

걸리는 시간이 DB2 Connect 머신 시간의 60%라고 결정하면, DB2 Connect를 조정하거나(리모트 클라이언트가 있으면) 또 다른 DB2 Connect 머신을 네트워크에 추가할 수 있습니다.

벤치마킹

벤치마킹은 특정 환경에서의 성능을 다른 환경에서의 성능과 비교하는 것입니다. 벤치마킹은 정상적인 환경에서 테스트 응용프로그램을 실행하여 시작할 수 있습니다. 성능 문제의 범위를 줄임에 따라, 특정한 테스트 케이스가 기능을 테스트하고 관찰할 범위를 제한하도록 개발될 수 있습니다.

벤치마킹은 복잡할 필요가 없습니다. 지정된 테스트 케이스는 원하는 정보를 얻기 위해 응용프로그램 전체를 애플레이트할 필요가 없습니다. 간단한 수치로 시작하여 보증이 되는 경우에만 복잡성을 계속 증가시키십시오.

바람직한 벤치마킹의 특성은 다음과 같습니다.

- 각 테스트는 반복 가능합니다.
- 각 테스트 반복은 동일한 시스템 상태에서 시작됩니다.
- 벤치마킹에 사용된 하드웨어와 소프트웨어는 사용자의 제품 환경과 일치합니다.
- 시나리오에 시스템에서 진행 중인 기타 다른 활동이 포함되어 있지 않으면 측정 중인 시스템 이외의 다른 시스템에 활동 중인 기능 또는 응용프로그램이 없습니다.

주: 시작된 응용프로그램은 최소화되거나 유휴 상태가 되더라도 메모리를 사용합니다. 이것은 페이지징을 일으키고 벤치마크 결과와 다르게 할 수 있습니다.

성능 도구

다음 표는 시스템 성능 측정에 도움이 되는 몇 가지 도구를 나열합니다. 이러한 도구는 자체적으로 시스템 자원을 사용하므로, 도구의 상태가 항상 사용 중일 수는 없습니다.

표 14. CPU 및 메모리 사용에 대한 성능 분석 도구

시스템	도구	설명
AIX	vmstat, time, ps, tprof	DB2 Connect 워크스테이션과 리모트 클라이언트에서 CPU 또는 메모리 경합 문제점에 대한 정보를 제공합니다.
HP-UX	vmstat, time, ps, monitor 및 glance(사용 가능한 경우)	
Windows	Microsoft Performance Monitor	

표 15. 데이터베이스 활동에 대한 성능 분석 도구

시스템	도구	설명
모두	데이터베이스 모니터	문제가 데이터베이스에서 발생했는지 판별합니다.

표 15. 데이터베이스 활동에 대한 성능 분석 도구 (계속)

시스템	도구	설명
System z	IBM Tivoli® OMEGAMON® XE for DB2 Performance Monitor on z/OS, ASG-TMON for DB2 (ASG) 및 CA Insight Performance Monitor for DB2 for z/OS(Computer Associates International, Inc.)	
Windows	Microsoft Performance Monitor	

표 16. 네트워크 활동에 대한 성능 분석 도구

시스템	도구	설명
AIX	netpmon	초당 수신된 패킷 수 또는 프레임 수와 같은 TCP/IP 통계를 포함하여 낮은 레벨의 네트워크 통계를 보고합니다.
3745와 같은 네트워크 제어기	NetView® Performance Monitor	통신 제어 및 VTAM®의 활용도를 보고합니다.
Linux 및 UNIX	netstat	TCP/IP 통신량을 처리합니다.

제 15 장 ODBC 액세스 최적화

DB2 데이터베이스는 ODBC를 통해 통신 성능을 향상시키도록 설계된 특수 최적화를 제공합니다. 이러한 향상 기능은 Microsoft Access, Lotus Approach® 또는 Visual Basic에서 사용 가능합니다. DB2의 클라이언트 구성 지원 프로그램을 사용하여 보다 빨라진 ODBC의 장점을 누릴 수 있습니다.

최적화된 ODBC를 활성화하려면 다음을 수행하십시오.

- 새 연결을 정의할 경우 다음을 수행하십시오.
 1. DB2 CA를 시작하십시오.
 2. 선택 항목 메뉴를 열고 마법사를 사용한 데이터베이스 추가...를 선택하십시오.
 3. 데이터 소스 페이지에 도달할 때까지 마법사 페이지를 따라 가십시오.
 4. **CLI/ODBC**에 대해 이 데이터베이스 등록을 선택하십시오.
 5. 이 데이터베이스에 액세스하는 CLI/ODBC 응용프로그램의 등록 방법을 지정하십시오.
 - 시스템 데이터 소스로는 시스템의 모든 사용자가 데이터베이스를 사용할 수 있음을 의미합니다.
 - 사용자 데이터 소스로는 사용자가 데이터베이스에 액세스할 수 있는 유일한 사용자임을 의미합니다.
 - 파일 데이터 소스로는 데이터 소스 정보를 포함하는 파일이 작성됨을 의미합니다. TCP/IP 연결이 있는 경우, 이 데이터 소스 파일을 다른 워크스테이션과 공유할 수 있습니다. 그렇지 않으면, 파일을 이 컴퓨터에서만 사용할 수 있습니다.
 6. 데이터 소스 이름을 입력하십시오.
 7. (선택사항) 특정 응용프로그램에 대한 데이터 소스 설정을 최적화하려면 최적화 대상 응용프로그램 목록에서 응용프로그램을 선택하십시오.
 8. 확인을 누르고 CA를 종료하십시오.
- 기존 연결을 갱신하려면 다음을 수행하십시오.
 1. DB2 CA를 시작하십시오.
 2. 최적화할 데이터베이스 별명을 더블 클릭하십시오.
 3. 데이터 소스를 누르십시오.
 4. **CLI/ODBC**에 대해 이 데이터베이스 등록을 선택하십시오.
 5. 이 데이터베이스에 액세스하는 CLI/ODBC 응용프로그램의 등록 방법을 지정하십시오.

- 시스템 데이터 소스로는 시스템의 모든 사용자가 데이터베이스를 사용할 수 있음을 의미합니다.
 - 사용자 데이터 소스로는 사용자가 데이터베이스에 액세스할 수 있는 유일한 사용자임을 의미합니다.
 - 파일 데이터 소스로는 데이터 소스 정보를 포함하는 파일이 작성됨을 의미합니다. TCP/IP 연결이 있는 경우, 이 데이터 소스 파일을 다른 워크스테이션과 공유할 수 있습니다. 그렇지 않으면, 파일을 이 컴퓨터에서만 사용할 수 있습니다.
6. 데이터 소스 이름을 입력하십시오.
 7. (선택사항) 특정 응용프로그램에 대한 데이터 소스 설정을 최적화하려면 최적화 대상 응용프로그램 목록에서 응용프로그램을 선택하십시오.
 8. 확인을 누르고 CA를 종료하십시오.

제 16 장 응용프로그램 설계

응용프로그램을 작성할 때 여러 방법으로 성능을 향상시킬 수 있습니다.

복합 SQL 및 스토어드 프로시저

많은 명령과 응답을 송신하고 수신하는 응용프로그램에서 네트워크 오버헤드는 심각한 문제입니다. 복합 SQL과 스토어드 프로시저는 이 오버헤드를 줄이는 방법입니다.

응용프로그램이 프로그래밍 로직이 간섭 없이 여러 SQL문을 송신하는 경우, 복합 SQL을 사용할 수 있습니다. SQL문 그룹에 프로그래밍 로직이 필요한 경우에는 스토어드 프로시저를 사용할 수 있습니다.

다음 명령을 제외한 모든 실행 가능한 명령문은 복합 SQL문 안에 들어 있습니다.

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

스토어드 프로시저는 프로그램 로직을 서버에 배치함으로써 네트워크 통신량을 줄여줍니다. 프로시저를 종료할 때 자동으로 커밋할 수 있습니다. 또한 결과 집합을 리턴할 수도 있어서 클라이언트에서의 응용프로그램 로직이 최소화됩니다.

요청 그룹화

관련 데이터베이스 요청(SQL문)을 하나의 데이터베이스 요청으로 그룹화하면 네트워크를 통하여 전송되는 요청 수와 응답 수를 줄일 수 있습니다.

예를 들어, 다음 명령문을

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

다음과 같이

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

그룹화하면 네트워크를 통하여 더 적은 요청을 송신합니다.

IN과 BETWEEN과 같은 키워드를 사용하여 리턴되는 행 수를 줄일 수도 있습니다. 추가로, UPDATE와 DELETE문에서 WHERE, IN, BETWEEN 키워드를 사용할 수 있습니다.

술어 로직

술어 로직을 사용하여 필요한 행과 컬럼만을 요청할 수 있습니다. 그러면 데이터 전송에서 네트워크 통신량과 CPU 오버헤드를 최소화합니다.

예를 들어, 다음의 쿼리는 사용하지 마십시오.

```
SELECT * FROM TABLEA
```

ROW_ID=1인 TABLEA의 첫 번째 행이 정말로 필요하다거나 컬럼 1과 컬럼 2만 필요한 경우입니다.

데이터 블로킹

서버로부터 많은 양의 데이터를 원하는 경우, 데이터 블로킹을 사용해야 합니다. 블로킹은 네트워크 대역폭의 사용을 향상시키고 IBM 메인프레임 데이터베이스 서버 및 DB2 Connect 서버의 CPU 오버헤드를 줄여줍니다. 크기에 관계없이 메시지마다 고정된 CPU 양과 네트워크 오버헤드가 있습니다. 데이터 블로킹은 같은 양의 데이터 전송에 필요한 메시지 수를 줄입니다.

블로킹을 사용하면, 쿼리로부터의 첫 번째 데이터 행은 첫 번째 블록이 수신될 때까지 응용프로그램에 전달되지 않습니다. 블로킹으로 첫 번째 행에 대한 검색 시간은 증가하지만, 다음 행에 대한 검색 시간은 개선됩니다.

사용된 메모리 양을 고려해 볼 수 있습니다. 메모리 작업 세트는 블로킹이 작동되면 보통 증가합니다.

DB2 Connect에서 각 블록 내에서 전송되는 데이터 양을 제어할 수 있습니다. 블로킹을 호출하려면, prep 또는 bind 명령의 BLOCKING 옵션을 사용하십시오. 블로킹은 다음 경우에 작동됩니다.

- 커서가 읽기 전용인 경우
- 커서가 앰비규어스 상태이고 블로킹이 prep 또는 bind 중에 지정된 경우

주: 동적 SQL을 사용할 경우, 커서는 항상 앰비규어스 상태입니다.

BLOCKING을 사용하는 SQL문

갱신 가능한 SELECT문(UPDATE/DELETE WHERE CURRENT OF문 사용)은 비블로킹 쿼리이므로 절대적으로 필요한 경우에만 사용해야 합니다.

갱신 가능한 SELECT를 사용하면 SELECT가 완료된 시간과 UPDATE/DELETE가 발행된 시간 사이에 행이 변경되지 않습니다. 이동시 처리 레벨이 사용자 응

응용프로그램에서 중요하지 않으면, 갱신이 가능하지 않은 SELECT로부터 리턴된 값에 기초한 검색 기준으로 DELETE 또는 UPDATE를 사용할 수 있습니다.

읽기 전용 SELECT에서 FOR FETCH ONLY를 지정하십시오(VM과 VSE에서는 지원되지 않으므로 제외).

정적 SQL 및 동적 SQL

가능한 정적 SQL을 사용하십시오. 이것은 런타임 SQL 섹션 준비와 앰비규어스 커서를 피합니다. 동적 SQL을 피할 수 없는 경우, 다음을 수행하여 네트워크 트래픽을 최소화하고 성능을 향상시킬 수 있습니다.

- 명령문이 SELECT이고 준비되었으면, PREPARE ... INTO SQLDA를 수행하십시오. SQLDA는 설정에 필요한 전체 크기로 할당되어야 합니다. 최대 컬럼 수가 x 이고 변경되지 않는 경우 x SQLVAR로 SQLDA를 할당하십시오. 가능한 컬럼 수를 알 수 없으면(메모리는 문제가 되지 않음), 최대 SQLVAR 수(256)를 사용하십시오.

SQLDA 할당이 충분하지 않아서 리턴하는 SQLDA를 저장할 수 없으면, 프로그램은 그 결과를 다시 저장할 수 있는 큰 SQLDA를 갖는 또 다른 DESCRIBE를 실행해야 합니다. 이로 인해 네트워크 통신량을 증가시킵니다.

PREPARE 및 DESCRIBE 시퀀스는 사용하지 마십시오. PREPARE.....INTO 문을 사용하면 더 나은 성능을 얻을 수 있습니다.

- 동적 COMMIT 또는 ROLLBACK문 대신에 정적으로 바인드된 SQL COMMIT 또는 ROLLBACK문을 수행하십시오.
- SELECT, COMMIT 또는 ROLLBACK문이 아니면, PREPARE 시퀀스와 EXECUTE 시퀀스 대신에 명령문을 실행하기 위해 EXECUTE IMMEDIATE를 실행하십시오.
- ODBC 응용프로그램은 동적 SQL을 사용합니다. CLI/ODBC 정적 프로파일링을 사용하여 성능을 개선할 수도 있습니다. 이 기능을 통해 ODBC 호출을 캡처하여 데이터베이스 패키지에 저장된 정적 명령문으로 변환시킬 수 있습니다. 실제적으로 얻게 될 성능은 응용프로그램의 복잡도에 따라 다릅니다.

기타 SQL 고려사항

일반적으로 프로그램에서 명령행 처리기(CLP)를 사용하는 것이 동적 SQL을 사용하는 것보다 더 느린데, CLP가 SQL을 데이터베이스 엔진에 제출하기 전에 입력을 구문 분석해야 하기 때문입니다. 또한, CLP는 수신한 데이터를 형식화하는 데, 이는 응용프로그램에 필요하지 않을 수 있습니다.

같은 SQL문도 REXX™와 같은 인터프리터 언어에서 사용하는 것이 C와 같은 컴파일 언어에서보다 기본적으로 더 느립니다.

CONNECT문에는 유형 1과 유형 2라는 두 가지 유형이 있습니다. 유형 2로 데이터베이스에 연결하는 것은 이전의 연결을 잠복 상태로만 두고 삭제하지는 않습니다. 나중에 잠복 연결로 스위치하면, 로딩 라이브러리와 내부 데이터 구조 설정의 오버헤드를 피합니다. 이러한 이유로, 유형 2 연결을 사용하면 두 개 이상의 데이터베이스에 액세스하는 응용프로그램의 성능을 향상시킬 수 있습니다.

제 17 장 연결 관리

연결 관리는 연결 풀링과 연결 집중기와 같은 두 가지 조작으로 구성됩니다. 연결 풀링은 데이터베이스 연결의 오버헤드를 줄이고 연결 볼륨을 관리합니다. 연결 집중기는 호스트 데이터베이스 서버에서 사용되는 자원을 최적화하여 작업 환경 내의 확장성을 증가시킵니다. 두 조작 모두 여기에 설명되어 있습니다.

연결 풀링

DB2 Connect Enterprise Edition과 같은 DB2 Connect 서버 제품은 흔히, 수 천개의 동시 클라이언트 요청을 위한 데이터베이스 연결을 제공합니다. 데이터베이스 서버에 연결하고 연결을 끊는 것은 데이터베이스 서버 및 DB2 Connect 서버 성능에 불리한 매우 자원 집약적인 프로세스입니다.

이것은 웹 페이지를 방문할 때마다 데이터베이스 서버에 새로 연결하여 쿼리를 수행하고 연결을 종료하는 웹 환경에서 특히 그렇습니다. 이 오버헤드를 줄이기 위해 DB2 Connect 서버 제품은 연결 풀링을 사용하여 쉽게 액세스할 수 있는 풀에서 데이터베이스에 대해 열려 있는 연결을 유지보수합니다.

웹 기술을 기반으로 하는 대부분의 응용프로그램은 짧은 트랜잭션을 매우 많이 실행합니다. 일반적인 웹 트랜잭션은 자체 연결의 일부로 실행됩니다. 즉, 트랜잭션의 실행은 데이터베이스 연결을 설정한 다음, 몇 가지 SQL문을 실행한 후 이 연결을 종료한다는 것을 의미합니다. 이러한 연결 설정 및 단절 프로세스는 자원을 많이 사용하므로 비용이 많이 듭니다. 여기에는 DB2 Connect 에이전트 작성, 이 에이전트와 DB2 서버 간에 네트워크 연결 설정 및 서버에서 DB2 스레드 작성이 포함됩니다. 더 오래 연결하기 위해 필요한 자원은 이 연결에서 실행되는 모든 트랜잭션보다 많지만, 일반적인 웹 트랜잭션의 경우 이러한 자원 사용은 트랜잭션 자체를 실행하는 자원 사용도를 초과합니다.

연결 풀링은 설정된 연결 인프라스트럭처를 다음 연결에 다시 사용할 수 있도록 하는 기술입니다. DB2 Connect 인스턴스가 시작되면 코디네이팅 에이전트 풀이 작성됩니다. 연결 요청이 들어오면 에이전트가 이 요청에 지정됩니다. 에이전트가 DB2 서버에 연결되고 스레드가 DB2에 작성됩니다. 응용프로그램이 연결 끊기 요청을 발행하면, 에이전트가 이 요청을 DB2 서버에 전달하지 않습니다. 대신, 에이전트는 다시 풀에 들어갑니다. 풀의 에이전트는 여전히 DB2 서버 및 해당 DB2 스레드에 대한 연결을 소유합니다. 다른 응용프로그램이 연결 요청을 발행하면, 이 에이전트는 새 응용프로그램에 지정됩니다. 안전한 조작을 위해 사용자 식별 정보가 DB2 스레드로 전달되고 이 스레드는 사용자 인증을 수행합니다.

DB2 Connect의 연결 풀링은 이런 환경에서 중요한 성능 향상을 제공합니다. DB2 Connect는 사용 가능한 풀의 데이터베이스에 대해 열려 있는 연결을 유지보수합니다. 클라이언트가 연결을 요청하면 준비된 연결이 들어 있는 이 풀에서 연결을 제공할 수 있습니다. 연결 풀링은 이 연결을 열고 닫는 데 일반적으로 소요되는 오버헤드를 상당 부분 줄일 수 있습니다.

연결 풀링은 DB2 Connect를 통해 호스트에 연결하는 응용프로그램에 투명합니다. 응용프로그램이 호스트로부터 연결 끊기를 요청하면, DB2 Connect가 응용프로그램과의 인바운드 연결을 제거하지만 풀의 호스트에 대한 아웃바운드 연결은 보존합니다. 새 응용프로그램이 연결을 요청하면, DB2 Connect가 기존 풀의 연결을 사용합니다. 이미 존재하는 연결을 사용하면 호스트에서 높은 CPU 연결 시간 뿐만 아니라 전체적인 연결 시간도 줄일 수 있습니다.

DB2 Connect 에이전트는 유휴 또는 활동 중 상태 중 하나일 수 있습니다. 에이전트가 응용프로그램에 대한 작업을 실행하고 있을 때는 활동 중입니다. 이 작업이 완료되면 에이전트는 동일하거나 다른 응용프로그램으로부터 또 다른 작업을 대기하는 유휴 상태로 들어 갑니다. 모든 유휴 에이전트는 유휴 에이전트 풀이라는 곳에 함께 보존됩니다. **num_poolagents** 구성 매개변수를 사용하여 이 풀의 크기를 구성할 수 있습니다. 이 매개변수는 시스템이 유지보수할 최대 유휴 에이전트 수와 같습니다. 이 매개변수를 0으로 설정하는 것은 연결 풀링 기능을 해제하는 것과 같습니다. 이 구성 매개변수의 디폴트값은 값이 100인 AUTOMATIC으로 설정되어 있습니다. AUTOMATIC으로 설정되면 DB2 Connect는 유휴 에이전트 풀에 있는 유휴 에이전트 수를 자동으로 관리합니다.

DB2 Connect는 첫 번째 클라이언트 요청을 수신하기 전에 데이터베이스에 대한 연결을 설정하지 않습니다. 다른 방법으로, 클라이언트가 요청하기 전에 유휴 에이전트의 풀을 채울 수 있습니다. 이 풀은 **num_initagents** 구성 매개변수를 사용하여 시작할 때 채울 수 있습니다. 이 매개변수는 작성되어야 할 유휴 에이전트의 수를 시동시 결정합니다. 이러한 유휴 에이전트는 초기에 호스트 데이터베이스 서버에 대한 연결이 없습니다.

클라이언트가 호스트에 대한 연결을 요청하면, DB2 Connect가 호스트 데이터베이스 서버에 대해 연결을 갖는 풀 내의 에이전트 중에서 하나를 가져오려고 시도합니다. 그 에이전트에 문제가 있으면 유휴 풀에 있는 사용 가능한 다른 에이전트를 찾으려고 시도합니다. 풀이 비어 있으면, DB2 Connect가 새 에이전트를 작성합니다.

max_coordagents 구성 매개변수를 사용하여 동시에 사용할 수 있는 에이전트의 최대 수를 제어할 수 있습니다. 일단 이 수를 초과하면 오류 sqlcode SQL1226으로 인해 연결을 새로 작성할 수 없게 됩니다. (이 코드는 동시 아웃바운드 연결의 최대수가 초과되었음을 나타냅니다.) 이 구성 매개변수의 기본값은 값이 200인 AUTOMATIC으로 설정되어 있습니다. AUTOMATIC으로 설정되면 DB2 Connect는 코디네이터 에이전트의 수를 자동으로 관리합니다.

DB2 레지스트리 변수 DB2CONNECT_IN_APP_PROCESS를 사용하면 DB2 Connect 서버 제품과 동일한 머신에서 실행되는 응용프로그램이 DB2 Connect를 응용프로그램 프로세스 내에서 실행(디폴트 동작)하거나 응용프로그램이 DB2 Connect 서버 제품에 연결한 후 에이전트 내에서 호스트 연결을 실행할 수 있습니다. 응용프로그램이 연결 풀링을 사용하려면 DB2 Connect 서버 제품 에이전트에서 호스트에 대한 연결을 작성하고 DB2CONNECT_IN_APP_PROCESS를 NO로 설정해야 합니다.

DB2 Connect 연결 풀링 및 응용프로그램 서버(AS) 연결 풀링

연결 풀링은 대량의 트랜잭션을 지원해야 하는 웹 기술을 기본으로 하는 응용프로그램에 있어서는 필수적입니다. 대부분의 웹 응용프로그램 서버는 이제 자체적인 데이터베이스 연결 풀링 방법을 제공합니다. 예를 들어, Microsoft MTS(COM+) 및 IBM WebSphere는 모두 연결 풀링을 제공합니다.

이러한 서버가 구현하는 응용프로그램 풀링 메커니즘은 DB2 Connect 서버가 제공하는 메커니즘과 매우 다릅니다. 응용프로그램 서버는 자체 사용 목적으로만 연결을 풀링하므로 일반적으로 사용자 ID, 암호, 분리 레벨 등이 모든 연결에서 동일합니다. 더욱 중요한 것은 응용프로그램 서버는 같은 프로세스에 의해 시작된 연결만을 풀링합니다. 즉 다른 머신, 사용자 또는 프로세스로부터의 연결은 풀링되지 않습니다. 이러한 응용프로그램 서버 풀링 기술은 동일한 응용프로그램 인스턴스에 의해 설정된 연결을 재사용하는 데 효과적이지만, 다중 사용자, 서버 등에서 연결을 풀링하는 데는 절대로 비효율적입니다.

DB2 Connect 서버가 제공하는 연결 풀링은 응용프로그램, 머신 및 사용자와 완전히 독립적입니다. 서로 다른 사용자 ID를 갖는 여러 클라이언트, 응용프로그램 서버로부터의 연결은 모두 서로의 연결을 재사용할 수 있으므로 풀링된 자원을 더 잘 활용할 수 있습니다.

어떤 유형의 연결 풀링을 사용하는 것이 더 효율적일까요? 모두 사용하는 것입니다. 일반적으로, DB2 Connect 연결 풀링과 응용프로그램 서버 연결 풀링은 서로 방해하지 않으므로 함께 사용하는 것이 좋은 전략입니다. 응용프로그램 서버 연결 풀링이 사용 가능하더라도 DB2 Connect 연결 풀링이 DB2 Connect 서버를 사용하는 다른 클라이언트 뿐만 아니라 여러 응용프로그램 서버에 대해서도 연결을 재사용할 수 있도록 합니다.

연결 집중기(connection concentrator)

연결 집중기(connection concentrator)는 z/OS용 DB2 데이터베이스 서버에 필요한 자원을 줄여 다수의 워크스테이션 및 웹 사용자를 지원할 수 있도록 합니다. 이 기능은 z/OS용 DB2 및 DB2 Connect 솔루션의 확장성을 크게 증가시키는 한편, z/OS용 DB2 데이터 공유 환경에서 실패 없는 조작 및 트랜잭션 레벨 로드 밸런싱을 제공합니다.

연결 집중기(connection concentrator)를 사용하면 응용프로그램이 DB2 호스트 서버에서 자원을 소비하지 않고도 연결 상태를 계속 유지할 수 있습니다. 수 천명의 사용자가 응용프로그램을 사용하고 DB2 호스트 서버에서 몇 가지 스레드만 활동할 수 있습니다.

DB2 Connect의 연결 집중기(connection concentrator) 기술을 사용하면 DB2 Connect 서버 제품(예: DB2 Connect Enterprise Edition)이 동시에 비즈니스 트랜잭션을 실행하는 수 천 명의 사용자를 지원할 수 있으며 System z 호스트 또는 IBM Power Systems 데이터베이스 서버에 필요한 자원을 크게 줄일 수 있습니다. 훨씬 적은 수의 System z 호스트 또는 IBM Power Systems 데이터베이스 서버 연결에 모든 응용프로그램의 워크로드를 집중시킴으로써 이 목표를 이룰 수 있습니다. 이는 위에 설명된 연결 풀링 기능과 유사한 것처럼 보이지만, 사실 매우 많은 볼륨의 OLTP(On-line Transaction Processing) 응용프로그램에 대한 자원 소비를 줄일 수 있는 더 정교한 접근 방법입니다.

연결 집중기(connection concentrator)는 에이전트 개념을 사용하여 이를 다음 두 개의 엔티티로 분할합니다.

- 논리 에이전트: 응용프로그램 연결을 나타냅니다.
- 코디네이팅 에이전트: DB2 연결 및 스레드를 소유하고 응용프로그램 요청을 실행합니다.

새 응용프로그램이 호스트에 연결하려고 하면 응용프로그램에 논리 에이전트가 지정됩니다. SQL을 데이터베이스에 전달하려면 코디네이팅 에이전트가 필요하며 새 트랜잭션이 시작되는 즉시 지정됩니다. 이 아키텍처의 핵심은 코디네이팅 에이전트가 다음과 같다는 점입니다.

- 논리 에이전트와 연관되지 않습니다.
- 커밋 또는 롤백으로 인해 트랜잭션 완료 시 풀로 리턴됩니다.

다른 핵심 기능은 데이터 공유 환경의 새 트랜잭션에 코디네이팅 에이전트를 지정하는 방법입니다. DB2 Connect는 System z WLM(Work Load Manager) 정보를 사용하는 정교한 스케줄링 알고리즘을 구현합니다. 이 정보는 WLM에 설정된 기준에 따라 데이터 공유 그룹의 구성원 간에 워크로드를 분산하는 데 사용됩니다. WLM은 각 구성원의 로드를 인지할 뿐 아니라 각각의 사용 가능성 여부도 알고 있습니다. 그 결과 DB2 Connect는 실패했거나 오버로드된 구성원으로부터 사용 중이며 활용도가 낮은 구성원으로 작업을 쉽게 재배치할 수 있습니다. DB2 Connect 연결 집중기(connection concentrator)는 최대 논리 에이전트 수(max_connections)를 코디네이팅 에이전트 수(max_coordagents)보다 높게 설정할 때 활성화됩니다.

응용프로그램을 종료함으로써 더 이상 연결이 필요하지 않게 될 때 연결 풀링으로 연결 설정 단가를 절약할 수 있습니다. 즉, 한 응용프로그램에서 풀 처리된 연결을 재사용하기 전에 다른 응용프로그램은 연결 해제 되어야 합니다.

대신, DB2 Connect는 연결 집중기(connection concentrator)를 통해 응용프로그램이 트랜잭션을 완료한 후 즉시 다른 응용프로그램이 연결을 사용할 수 있도록 하므로 다른 응용프로그램이 연결을 끊지 않아도 됩니다. 본질적으로, 데이터베이스 서버 연결 및 연관된 호스트와 DB2 Connect 자원은 활동 트랜잭션이 있는 동안에만 응용프로그램에 의해 사용됩니다. 트랜잭션이 완료되자마자, 실행되는 트랜잭션을 필요로 하는 다른 응용프로그램에서 연결 및 연관된 자원을 사용할 수 있습니다.

이전 버전의 DB2 Connect에서는 모든 활성 응용프로그램에 응용프로그램 요청 및 데이터베이스 연결을 관리하는 EDU(Engine Dispatchable Unit)가 있었습니다. 이 EDU를 일반적으로 코디네이터 에이전트라고 합니다. 각 코디네이터 에이전트는 응용프로그램 및 EDU의 상태나 컨텍스트를 추적합니다. 각 EDU는 연결 수가 증가하고 에이전트 간의 컨텍스트 전환으로 인해 추가 오버헤드가 발생할 때 상당 양의 메모리를 사용합니다.

위의 아키텍처에서는 연결과 EDU간에 일대일 대응 관계가 성립됩니다. 그러나 연결 집중기(connection concentrator)는 연결과 EDU간에 다대일 대응 관계를 허용합니다. 즉, 연결 (X)와 EDU (Y)가 현재 $X \geq Y$ 관계에 있습니다.

연결 집중기(connection concentrator)는 에이전트를 두 개의 엔티티 즉, 논리 에이전트와 작업자 에이전트로 분할합니다. 논리 에이전트는 특정 EDU에 대한 참조없이 응용프로그램만을 나타냅니다. 논리 에이전트에는 응용프로그램에 필요한 모든 정보와 제어 블록이 들어 있습니다. 서버에 연결된 n 개의 응용프로그램이 있다면 서버상에 n 개의 논리 에이전트가 있게 됩니다. 작업자 에이전트는 응용프로그램 요청을 실행하는 물리적인 EDU이지만 주어진 응용프로그램에 대해 영구적으로 접속하지 않습니다. 작업자 에이전트는 논리 에이전트와 연관되어 트랜잭션을 수행하고 트랜잭션 바운더리에서 연관을 종료한 후 사용 가능한 풀로 리턴합니다.

디스패처라고 하는 엔티티는 논리 에이전트에 작업자 에이전트를 지정합니다. 특정 컴퓨팅 플랫폼에서 열린 파일 핸들 수를 제한하면 둘 이상의 스케줄러 인스턴스가 생성될 수 있습니다.

연결 집중기(connection concentrator)의 제한사항

DB2 Connect 서버 집중기(concentrator)를 사용하는 데 중요한 여러 제한사항이 있습니다. 시스템에서 연결 집중기(connection concentrator)를 사용하려고 시도하기 전에 다음 내용을 모두 읽어 보십시오.

일반 제한사항:

- 집중기(concentrator)는 로컬 및 리모트 클라이언트로부터 인바운드 연결을 설정하는데 TCP/IP 프로토콜을 따릅니다. TCP/IP 또는 로컬(IPC)을 사용하는 인바운드 연결만이 풀 처리된 아웃바운드 연결을 이용할 수 있습니다. 집중기(concentrator)는

Named Pipes와 같은 다른 통신 프로토콜을 통한 연결도 허용하지만, 그러한 연결에는 XA 집중 기능을 사용할 수 없습니다.

- XA 긴밀 결합 트랜잭션 지원의 경우, 동일한 XA 트랜잭션에 참여하는 모든 응용프로그램이 같은 DB2 Connect 서버 인스턴스를 사용하여 호스트에 연결해야 합니다.
- 트랜잭션 바운더리에서 보류 자원(예: 보류 커서)을 닫는 응용프로그램만 집중기(concentrator)의 이점을 얻을 수 있습니다. 보류 커서를 닫지 않은 트랜잭션은 계속 진행되기는 하지만, 전용 작업자 에이전트를 할당받게 되고 이로 인해 집중기(concentrator)의 전체 기능 세트를 사용할 수 없게 됩니다.
- 임시 테이블을 선언하는 경우, 트랜잭션 또는 분기 바운더리에서 테이블을 명시적으로 삭제(drop)해야 합니다. 테이블을 삭제(drop)하지 못하면 연결 집중이 해제되지만 응용프로그램은 계속 작동합니다.
- 동일한 XA 트랜잭션에 참여하는 모든 응용프로그램은 동일한 CCSID를 가지고 있어야 하며 동일한 사용자 ID를 사용하여 연결해야 합니다.
- 2단계 연결을 지원하기 위해 아웃바운드 연결을 설정한 경우, 해당 연결의 에이전트는 2단계 연결을 지원하는 데만 사용할 수 있습니다. 마찬가지로, 1단계 연결을 지원하기 위해 설정된 에이전트는 1단계 연결만 지원할 수 있습니다.
- 집중기(concentrator)는 콜 레벨 인터페이스(CLI)에서 동적 SQL만 지원합니다. 집중기(concentrator)는 각 트랜잭션 바운더리에서 다시 준비되는 명령문에 따라 달라지므로 CLI 응용프로그램은 KEEP DYNAMIC을 집중기(concentrator)로 사용해서는 안 됩니다.
- Embedded 동적 SQL 응용프로그램으로부터의 동적 PREPARE 요청은 거부됩니다. 정적 SQL을 사용하거나 동적 SQL문에 대해 CLI를 사용하도록 응용프로그램을 변경해야 합니다.
- 연결 집중기(connection concentrator)가 설정된 경우 DB2 Connect 서버에 대한 인바운드 요청에 SSL을 사용할 수 없습니다. 그러나 목표 데이터베이스 서버에 대한 아웃바운드 요청에는 SSL을 사용할 수 있습니다. 연결 집중기(connection concentrator)가 해제된 경우 인바운드 및 아웃바운드 요청에 SSL을 모두 사용할 수 있습니다.

DB2 버전 9 또는 버전 8 FixPak 13(이상)으로 작업하는 경우, DB2 Connect 연결 집중기(connection concentrator) 지원을 사용하려면 IBM Power Systems 버전 5 릴리스 4(PTF SI23726)가 필요합니다. 그렇지 않으면 연결 집중기(connection concentrator)의 XA 부분만 지원됩니다.

연결 집중기(connection concentrator) 활성화

데이터베이스 관리 프로그램 구성 매개변수 *max_coordagents*는 논리 에이전트의 최대 수를 설정합니다. *max_connections* 값을 디폴트값보다 큰 수로 설정하여 집중기(concentrator) 기능을 활성화할 수 있습니다. *max_connections*는 *max_coordagents* 값

과 같습니다. 각 응용프로그램에는 하나의 논리 에이전트가 있기 때문에, *max_connections*가 실제로 데이터베이스 인스턴스에 연결될 수 있는 응용프로그램의 수를 제어하는 반면에, *max_coordagents*는 언제든지 활동할 수 있는 인바운드 연결의 수를 제어합니다. *max_connections*의 범위는 *max_coordagents*에서 최대 64,000까지의 숫자입니다. 논리 에이전트의 최대수는 *max_coordagents*와 같습니다.

max_connections 및 *max_coordagents* 모두 AUTOMATIC으로 설정할 수 있습니다. *max_connections*를 AUTOMATIC으로 설정하면 연결 수가 기본 구성 값을 초과하여 증가할 수 있습니다. *max_connections* 및 *max_coordagents* 모두 AUTOMATIC으로 설정하면 *max_connections*가 기본값을 초과하여 증가하고 *max_coordagents*는 자동으로 증가하여 연결과 코디네이터 에이전트 사이의 집중 비율을 유지합니다.

기존의 여러 구성 매개변수는 에이전트를 구성하는 데 사용됩니다. 이 매개변수는 다음과 같습니다.

max_coordagents

활동 중인 코디네이터 에이전트의 최대수

num_poolagents

에이전트 풀 크기. 에이전트 풀에는 비활동 에이전트 및 유휴 에이전트가 포함됩니다. 성능 향상을 위해 *num_poolagents*를 평균 클라이언트 수로 구성해야 합니다.

num_initagents

풀에 있는 작업자 에이전트의 초기 수. 이것이 유휴 에이전트가 됩니다.

XA 트랜잭션 지원

DB2 Connect는 연결 집중기(connection concentrator)의 아키텍처를 통해 z/OS용 DB2 및 IBM i용 DB2에 밀접하게 결합된 XA 트랜잭션 지원을 제공할 수 있습니다. 집중기(concentrator)는 다른 트랜잭션에 대해 특정 XA 트랜잭션(단일 XID)과 작업자 에이전트를 연관시킵니다. 그러나 XA 트랜잭션이 *xa_end()*(분기 바운더리)로 끝나면, 작업자 에이전트가 그 자신을 일반 풀로 해제시키지 않습니다. 대신, 작업자는 특정 XA 트랜잭션과 연관된 상태로 남아 있습니다. 다른 응용프로그램이 동일한 XA 트랜잭션을 조인할 때, 작업자 에이전트가 해당 응용프로그램에 첨부됩니다.

트랜잭션 바운더리 호출은 모두 에이전트를 풀로 리턴시킵니다. 인스턴스의 경우에 읽기 전용인 *xa_prepare()*, *xa_rollback()*, *xa_recover()*, *xa_forget()*, *xa_commit()* 또는 롤백을 일으키는 XA 오류는 에이전트를 보통 풀로 리턴시킵니다. *xa_end()*는 트랜잭션 분기만을 종료시키며 XID와 연관된 것을 종료시키기에는 충분치 않습니다.

XA 트랜잭션 지원의 예

1. 4,000개 이상의 동시 연결이 필요한 환경을 고려해 보십시오. CGI 응용프로그램을 사용하는 웹 서버 또는 수많은 데스크탑 사용자가 있는 오피스 시스템에서는 둘다 이러한 요구사항을 초과할 수 있습니다. 이러한 경우, 효율성을 위해 대개 DB2 Connect가 독립형 게이트웨이로 작동해야 합니다. 즉, 데이터베이스와 DB2 Connect 시스템이 서로 다른 머신에 있어야 합니다.

DB2 Connect 서버 시스템은 데이터베이스 머신에 대해 동시에 열려 있는 4,000개의 연결을 유지보수할 수 없습니다. 대부분의 경우에, 주어진 순간에 발생하는 트랜잭션 수는 동시 연결 수보다 상당히 적을 것입니다. 그러면, 시스템 관리자는 데이터베이스 구성 매개변수를 다음과 같이 설정하여 시스템의 성능을 최대화할 수 있습니다.

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

집중기(concentrator)는 게이트웨이가 한번에 1,000개의 트랜잭션을 관리만 할지라도 최대 4,000개의 동시 세션을 열어 놓습니다.

2. 위의 예에서 작업자 에이전트는 논리 에이전트에 대한 관계를 끊임없이 연결하고 해제합니다. 유휴 상태가 아닌 에이전트는 데이터베이스에 대한 연결을 유지보수할 수 있지만 특정 트랜잭션에 참여하지 않으므로, 연결을 요청하는 논리 에이전트(응용프로그램)가 사용할 수 있습니다.

XA 트랜잭션의 경우에는 약간 다릅니다. 이 예에서는 TP 모니터가 DB2 Connect 게이트웨이 및 System z 또는 IBM Power Systems 데이터베이스와 함께 사용된다고 가정합니다. 응용프로그램에서 연결을 요청하면, 집중기(concentrator)는 비활동 에이전트를 인수하여 해당 요청을 처리하거나 새 작업자 에이전트를 작성합니다. 응용프로그램이 XA 트랜잭션을 요청한다고 가정하십시오. 이 트랜잭션에 대해 XID가 작성되며 작업자 에이전트가 이에 연관됩니다.

응용프로그램의 요청이 서비스될 때 이는 `xa_end()`를 발행하여 작업자 에이전트로부터 접속 해제합니다. 작업자 에이전트는 트랜잭션의 XID와 연관된 상태로 남아 있습니다. 이제 이는 연관된 XID를 가진 트랜잭션에 대한 요청만 서비스할 수 있습니다.

이때, 다른 응용프로그램이 비XA 트랜잭션을 요청할 수도 있습니다. 사용 가능한 다른 작업자 에이전트가 없을지라도 XID와 연관된 에이전트를 두 번째 응용프로그램에 대해 사용할 수 없습니다. 이 에이전트는 활동 중인 것으로 간주됩니다. 두 번째 응용프로그램은 그에 맞게 작성된 새 작업자 에이전트를 가집니다. 그 두 번째 응용프로그램이 해당 트랜잭션을 완료하고 나면, 작업자 에이전트가 사용 가능한 풀로 해제됩니다.

한편, 첫 번째 에이전트의 XID와 연관된 트랜잭션을 요청하는 다른 응용프로그램은 해당 에이전트를 접속 및 접속 해제할 수 있으며, 이 에이전트는 응용프로그램에 대해 전용 XA 트랜잭션을 실행합니다. 특정 트랜잭션을 요청하는 응용프로그램은 에이전트가 자유 상태인 경우에 이 작업자 에이전트로 전송됩니다.

작업자 에이전트는 응용프로그램에서 트랜잭션 바운더리 호출(xa_end()가 아님)을 발행할 때까지 일반 풀로 다시 되돌려 해제되지 않습니다. 인스턴스 경우에는 응용프로그램이 트랜잭션을 xa_commit()로 끝내게 되며 이 지점에서 작업자 에이전트는 XID와의 관계를 제거하고 사용 가능 풀로 리턴합니다. 이 때, 응용프로그램을 요청하면 또 다른 XA 또는 비XA 트랜잭션용으로 이를 사용할 수 있습니다.

연결 풀링 및 연결 집중기(connection concentrator)

연결 풀링 및 연결 집중기(connection concentrator)가 유사해 보이지만 구현에 있어서 차이가 있으며 서로 다른 문제에 접근합니다. 연결 풀링은 데이터베이스 연결의 오버헤드를 줄이고 연결 볼륨을 처리하는 데 도움이 됩니다. 연결 집중기(connection concentrator)는 호스트 데이터베이스 서버의 사용을 최적화하여 DB2 및 DB2 Connect 솔루션의 확장성을 증가시키는 데 도움이 됩니다.

연결 풀링을 사용할 때, 연결되어 있는 응용프로그램이 연결 끊기 요청을 발행한 후에야 해당 연결을 다시 사용할 수 있습니다. 많은 2 티어 클라이언트/서버 응용프로그램에서 사용자는 작업일 중에 연결을 끊지 않습니다. 마찬가지로 멀티 티어 응용프로그램에서 대부분의 응용프로그램 서버는 서버 시작시에 데이터베이스 연결을 설정하고 응용프로그램 서버가 종료한 후에 연결을 해제합니다.

이러한 환경에서 연결 풀링은 이점이 거의 없습니다. 그러나 연결 및 연결 끊기의 빈도가 더 높은 웹 및 클라이언트 서버 환경에서는 연결 풀링이 성능에 있어 매우 효과적입니다. 연결 집중기(connection concentrator)는 사용자 응용프로그램이 활성화되어 있는 동안 SQL 트랜잭션 중에만 호스트 데이터베이스 자원을 할당합니다. 이를 통해, 모든 응용프로그램 연결이 고유의 스레드를 소유하는 경우 DB2 스레드 수 및 DB2 스레드가 소모하는 자원 수를 훨씬 작게 구성할 수 있습니다.

오류 없는 작동 및 워크로드의 로드 전역 위해서는 연결 집중기(connection concentrator)가 모든 새 트랜잭션마다 작업을 재할당하므로 매우 효과적입니다. 한편, 연결 풀링은 매우 제한된 밸런싱만 제공하며, 연결시에만 가능합니다.

연결 풀링 및 연결 집중기(connection concentrator)가 서로 다른 문제를 다루지만 함께 사용해야 합니다.

WebSphere MQ Transaction Manager 및 DB2에 필요한 연결 집중기 (connection concentrator)

IBM WebSphere MQ(정식 이름은 IBM MQSeries®) 환경에서 응용프로그램을 실행할 때는 WebSphere MQ가 XA 호환 트랜잭션 관리 프로그램 기능을 하여 분산된 2 단계 커밋 트랜잭션을 조정할 수 있습니다. WebSphere MQ가 이러한 방식으로 트랜잭션 관리 프로그램 기능을 하고 데이터 소스가 DB2 제품군에서 오는 경우, 몇 가지 구성 요구사항이 적용됩니다.

이러한 트랜잭션 관리 프로그램 환경에 적용되는 구성 요구사항의 대부분이 문서의 다른 곳에 이미 설명되어 있습니다. 예를 들어, DB2 런타임 클라이언트에서 DB2 구성 매개변수 TP_MON_NAME을 『MQ』로 설정해야 합니다.

그러나, 한 가지 누락된 구성 요구사항이 있습니다. DB2 서버인 데이터 소스에 연결할 때 DB2 Connect 고유의 요구사항이 있습니다. WebSphere MQ를 사용하여 DB2 및 IBM i용 DB2 서버와 관련된 분산 트랜잭션을 조정하는 경우, DB2 Connect 연결 집중기(connection concentrator) 기능이 게이트웨이에서 사용 가능해야 합니다. **max_connections** 구성 매개변수의 값이 **max_coordagents** 구성 매개변수의 값보다 크면 연결 집중기(connection concentrator)가 사용 가능합니다.

연결 집중기(connection concentrator)를 사용하지 않으면 예기치 않은 트랜잭션 동작이 발생합니다.

제 18 장 DB2 Connect 서버 Sysplex 지원

Sysplex는 작업을 처리하기 위해 하드웨어 및 소프트웨어를 사용하여 협력하는 System z 서버의 콜렉션입니다. Sysplex는 서로 작동 중인 프로세서의 수를 증가시켜 협력을 통해 처리할 수 있는 작업의 양을 늘립니다. Sysplex는 처리 성능을 증가시킬 뿐 아니라 하드웨어 및 소프트웨어의 혼합 레벨 및 동적으로 시스템을 추가하는 데 융통성을 제공합니다.

Sysplex를 사용하여 DB2 Connect는 데이터 공유 그룹의 다양한 구성원 간의 연결을 매끄럽게 조정할 수 있습니다. 또한 Sysplex는 구성원과의 연결에서 장애가 발생할 경우, 대체 구성원을 시도하는 수단을 DB2 Connect에 제공합니다. Sysplex의 리라우팅 기능은 DB2 Connect 기능입니다. DB2 Connect는 디폴트로 Sysplex를 사용 가능하도록 하여 Sysplex의 성능을 리라우트합니다. 호스트 데이터베이스에 대한 Sysplex 지원은 해당 DCS 디렉토리 항목에서 SYSPLEX 매개변수를 제거하여 작동을 중지할 수 있으나 DCS 항목 자체는 지정된 다른 매개변수가 없는 경우에도 제거해서는 안됩니다.

Sysplex의 자동 클라이언트 리라우트 성능의 경우, 디폴트 동작은 통신이 실패한 연결에서 Sysplex가 사용 가능한 연결을 재시도하는 것입니다. DB2 Connect가 z/OS용 DB2 서버에 연결된 경우, 마지막으로 성공한 트랜잭션이 자원을 보유하지 않을 때까지 특수 레지스터 값이 재생됩니다.

DB2_MAX_CLIENT_CONNRETRIES 및 DB2_CONNRETRIES_INTERVAL 레지스트리 변수를 사용하여 사용 불가능을 포함한 정확한 자동 클라이언트 리라우트 재시도 동작을 구성할 수 있습니다. 연결 시간종료 레지스트리 변수는 DB2TCP_CLIENT_CONTIMEOUT입니다.

System z SYSPLEX 사용에 대한 고려사항

DB2 Connect는 여러 Sysplex에 연결을 라우팅할 때 로드 밸런스 및 결합 허용을 제공합니다. 데이터 공유 환경에서 실행 중인 z/OS용 DB2 데이터베이스 서버에 연결한 경우, DB2 Connect는 WLM(Workload Manager)이 제공한 시스템 로드 정보에 기초하여 데이터 공유 그룹을 구성하는 다른 DB2 서브시스템으로 워크로드를 분산합니다.

DB2 Connect는 WLM에서 Sysplex 구성원의 우선순위 목록을 수신합니다. 각 Sysplex는 각 연결 주소에 대해 비중이 큰 우선순위 정보를 리턴합니다. 그런 다음, DB2 Connect가 이 목록을 사용하여 들어오는 CONNECT 요청을 우선순위가 가장 높은 Sysplex 구성원에 분배함으로써 이 요청을 처리합니다. 로드 밸런싱의 경우, 각 연결 중

우선 순위 정보의 비중이 큰 Sysplex 목록을 확보합니다. DB2 Connect 연결 집중기(connection concentrator)가 사용 가능한 경우, 각 트랜잭션을 보낼 위치를 결정할 때 이 목록을 사용합니다.

주: DB2 Connect Sysplex를 사용하기 위해 System z Distributed Data Facility(DDF) 구성을 변경할 필요가 없습니다.

또한, DB2 Connect는 연결 실패시 대체 Sysplex 머신에 연결을 시도하여 결합 허용을 제공합니다. 모든 알려진 연결이 실패할 경우에만 오류가 해당 응용프로그램에 리턴됩니다.

DB2 Connect Sysplex는 에이전트 풀링을 고려하여 설계되었습니다. Sysplex가 사용 가능한 경우, DB2 Connect는 참여 구성원에 대한 연결이 끊겼을 때, 다른 DDF 구성원으로 연결을 라우트합니다. 리라우트는 Sysplex 서버 목록을 통해 수행됩니다.

집중기(concentrator)의 추가 이외에, DB2 Connect는 이제 트랜잭션 바운더리에서 워크로드를 조정할 수 있습니다. 이를 수행하려면 DB2 Connect 집중기(concentrator)가 사용 가능해야 합니다.

DB2 Sysplex 사용

일반 시나리오에서 DB2 Connect 서버(서버 A)는 두 개의 z/OS용 DB2 서버(서버 B 및 C)를 포함하는 Sysplex와 통신합니다.

Sysplex 서버 B	Sysplex 서버 C
HOST_NAME=MVSHOST	HOST_NAME=MVSHOST1

이 시나리오에서 이제 응용프로그램이 다음을 실행한다고 가정합니다.

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

MVSHOST 데이터베이스에 연결되었습니다. Sysplex가 DB2 Connect 서버 및 DCS 디렉토리 항목에 대해 사용 가능하므로 z/OS용 DB2가 각 Sysplex 구성원(participant)(MVSHOST 및 MVSHOST1)에 대해 DB2 Connect의 네트워크 주소를 식별합니다. DRDA4 프로토콜 및 메시지 플로우가 이 정보 리턴에 사용됩니다. 초기 연결이 설정되면, 리턴된 주소 목록이 DB2 Connect 워크스테이션에 캐시됩니다. 초기 CONNECT를 TCP/IP 노드에 대해 발행한 경우, IP 주소가 리턴됩니다.

로드 밸런스 및 결합 허용에 사용된 우선순위 정보

z/OS용 DB2가 제공한 주소 목록에는 각 네트워크 주소에 대한 연결 수를 포함하여 우선순위 정보도 포함됩니다. 이 목록은 DB2 Connect에서 새 연결을 성립할 때마다 새로 고칩니다. 이 추가 정보는 로드 균형과 결합 허용을 위해 사용됩니다.

DB2 Connect에서 사용한 캐시된 주소 목록

ALIASB에 대한 데이터베이스 연결이 연속적으로 실패한 경우, 오류 메시지 SQL30081N이 발행되고 연결이 삭제됩니다. 더 많은 연결 요청이 ALIASB에 수신된 경우 DB2 Connect가 다음을 수행합니다.

1. z/OS용 DB2에 의해 리턴된 우선순위 정보에 기초하여 캐시된 주소 목록에서 가장 높은 우선순위의 서버를 시도합니다. DB2 Connect는 언제나 이 정책을 사용하며 이러한 방식으로 로드 균형이 이루어집니다.
2. 이 연결 시도에 실패하는 경우, z/OS용 DB2에 의해 리턴된 대로 우선순위의 내림차순으로 목록의 다른 주소를 시도합니다. 이렇게 DB2 Connect가 Sysplex 정보를 사용하여 결함을 허용합니다.
3. 다른 모든 연결 시도가 실패하는 경우, DB2 Connect는 카탈로그된 노드 디렉토리에 포함된 주소를 사용하여 ALIASB에 연결을 다시 시도합니다.

db2pd 명령을 sysplex 매개변수(db2pd -sysplex)와 함께 사용하여 Sysplex 환경과 연관된 서버에 대한 정보를 검색할 수 있습니다.

Sysplex의 구성 요구사항

데이터베이스의 DCS 디렉토리 항목이 Sysplex(대소문자 구분 안함)를 6번째 위치의 매개변수에 포함하지 않는 경우, Sysplex를 해당 데이터베이스에서 사용할 수 없습니다.

제 19 장 클라이언트 Sysplex 지원

IBM Data Server Client 및 DB2 Connect 라이선스가 있는 비Java Data Server Driver는 z/OS용 DB2 Sysplex에 액세스할 수 있으며 트랜잭션 레벨 로드 밸런싱, 자동 클라이언트 라우트 및 제한된 XA 지원을 포함한 Sysplex 성능을 모두 사용할 수 있습니다.

Sysplex는 데이터 공유 그룹을 형성하는 DB2 시스템(구성원이라고 함)의 컬렉션입니다. 하나 이상의 결합 기능에는 데이터 공유 그룹의 빠른 캐싱 및 잠금 처리가 있습니다. WLM(Workload Manager), DVIPA(Dynamic Virtual IP Address) 및 Sysplex Distributor와 함께 Sysplex를 통해 클라이언트는 네트워크 복원력으로 TCP/IP를 통해 z/OS용 DB2 데이터베이스에 액세스할 수 있으며 데이터 공유 그룹 내의 구성원간에 균형된 방식으로 응용프로그램에 트랜잭션을 분산할 수 있습니다.

이들 성능의 중심은 DB2 데이터 공유 그룹의 각 구성원이 연결 경계 및 선택적으로 트랜잭션 경계에 리턴되는 서버 목록입니다. 이 목록에는 IP 주소 및 각 DB2 구성원의 IP 주소 및 사용 가능한 성능이 포함됩니다. 이 정보와 함께 클라이언트는 균형된 방식으로 트랜잭션을 분산하거나 통신 장애가 있을 경우 사용하기 위하여 DB2 구성원을 식별할 수 있습니다.

서버 목록은 DB2 데이터베이스에 대한 최초의 연결이 완료될 때 리턴됩니다. 따라서 초기의 데이터베이스 연결은 Sysplex Distributor가 소유한 그룹 DVIPA에서 방향이 지정되어야 합니다. 최소한 하나의 DB2 구성원을 사용할 수 있는 경우 Sysplex Distributor가 요청을 데이터베이스로 라우트합니다. 클라이언트가 서버 목록을 수신한 후 서버 목록의 정보를 기반으로 클라이언트가 직접 DB2 구성원에 액세스합니다.

클라이언트 Sysplex 지원을 구성하려면 db2dsdriver 구성 파일에 설정을 지정하십시오.

WLM, DVIPA 및 Sysplex Distributor에 대한 자세한 정보는 z/OS용 DB2 정보 센터(<http://publib.boulder.ibm.com/infocenter/dzichelp>)를 참조하십시오.

트랜잭션 레벨 워크로드 밸런싱(클라이언트 측)

IBM Data Server Client 및 DB2 Connect 라이선스가 있는 비Java Data Server Driver는 DB2 Connect 서버를 통과하지 않고 트랜잭션 레벨에서 워크로드 밸런싱을 수행할 수 있습니다. 트랜잭션 레벨의 워크로드 밸런싱을 통해 트랜잭션을 시작할 때 z/OS용 DB2 Sysplex의 구성원 워크로드를 재조정할 수 있습니다.

워크로드 밸런싱은 응용프로그램마다 단일 또는 다중 연결이 이루어지는 경우 트랜잭션이 동시에 발생할 때 수행됩니다. 트랜잭션이 동시에 실행되지 않고 연속적으로 실행되는 단일 장기 실행 연결만을 호출하는 응용프로그램의 경우 워크로드는 공간이 가장 큰 구성원으로 이동됩니다. 워크로드 밸런싱 알고리즘은 워크로드 관리 중량의 변경을 자동으로 조정합니다.

다음 개요에서는 클라이언트가 z/OS용 DB2 Sysplex에 연결되어 전송 재사용으로 워크로드 밸런싱이 일어나는 경우 발생하는 단계에 대해 설명합니다.

1. 클라이언트가 연결을 설정하면 서버가 구성된 워크로드 분산 정보가 포함된 서버 목록을 리턴합니다.

클라이언트에 의해 서버 목록이 캐시되고 구성 가능한 수명의 기본값이 30초가 됩니다.

2. 새 트랜잭션이 시작될 때 클라이언트가 서버 목록을 읽어 사용되지 않은 공간이 있는 구성원을 식별하고 충분히 이용되지 않은 구성원에 연관된 유휴 전송의 전송 풀을 찾습니다. (유휴 전송은 연관된 논리적 연결이 없는 전송입니다.)
 - 유휴 전송이 사용 가능하면 클라이언트가 논리적 연결을 전송과 연관시킵니다.
 - 사용자 구성 가능 시간종료 후에 전송 풀의 유휴 전송을 사용할 수 없고 전송 풀이 한계에 도달하여 새 전송을 할당할 수 없는 경우 응용프로그램에 오류가 리턴됩니다.
3. 트랜잭션이 실행되면 전송과 연관된 구성원에 액세스합니다.
4. 트랜잭션이 종료되면 클라이언트가 서버를 통해 논리적 연결에 전송 재사용이 계속 허용되는지를 확인합니다.
5. 전송 재사용이 허용되는 경우 서버가 논리적 연결을 위해 실행 환경에 적용하는 특수 레지스터의 SET 명령문의 목록을 리턴합니다.

클라이언트가 이 명령문을 캐시하면 논리적 연결이 새 전송과 연관될 때 실행 환경을 재구성하기 위해 재생됩니다.

6. 전송 재사용이 허용되면 논리적 연결이 정기적으로 전송으로부터 연관 해제되며 클라이언트가 서버 목록의 클라이언트 사본을 새로 고칩니다.
7. 새 트랜잭션에 워크로드 밸런싱이 필요하면 클라이언트가 같은 프로세스를 사용하여 논리적 연결을 전송과 연관시킵니다.

팁: 클라이언트 측 트랜잭션 레벨 워크로드 밸런싱이 단일 응용프로그램 프로세스에서 트랜잭션이 실행될 때 수행됩니다. 클라이언트에 실행되는 하나 이상의 응용프로그램 프로세스가 같은 Sysplex 사용 가능 데이터베이스에 액세스하는 경우 다른 프로세스와 별도로 각 응용프로그램 프로세스마다 워크로드 밸런싱이 수행됩니다. 다중 응용프로그램 또는 다중 리모트 클라이언트에 실행되는 프로세스의 다중 연결에 트랜잭션 레벨 워크로드 밸런싱을 수행하려면 DB2 Connect 서버에 사용 가능한 워크로드 밸런싱 기능을 사용하십시오.

트랜잭션 레벨 워크로드 밸런싱 구성(클라이언트 측)

클라이언트에서 트랜잭션 레벨의 워크로드 밸런싱을 구성하려면 db2dsdriver 구성 파일에 설정을 지정하십시오.

시작하기 전에

트랜잭션 레벨 워크로드 밸런싱을 수행하려면, 클라이언트는 TCP/IP 연결을 사용해야 하며 DB2 Connect 라이선스가 있어야 합니다. 다음 클라이언트는 트랜잭션 레벨 워크로드 밸런싱에 대한 지원을 제공합니다.

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver Package
- IBM Data Server Driver for ODBC and CLI

이 태스크에 대한 정보

이 태스크는 트랜잭션 레벨 워크로드 밸런싱을 구성하여 클라이언트가 DB2 Connect 서버를 통하지 않고 트랜잭션 레벨에서 워크로드 밸런싱을 수행할 수 있도록 합니다.

프로시저

1. db2dsdriver 구성 파일에서 데이터베이스 항목 또는 DSN 항목의 WLB 서브섹션의 enableWLB 매개변수를 "참"으로 설정하여 트랜잭션 레벨 워크로드 밸런싱을 작동하십시오. 예를 들어 db2dsdriver 구성 파일에 다음을 지정하십시오.

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
</database>
```

디폴트로 enableWLB가 거짓이면 워크로드 밸런싱을 사용할 수 없습니다.

2. 옵션: 다음 매개변수의 값을 지정하여 워크로드 밸런싱의 설정을 미세 조정하십시오. 이 매개변수의 디폴트값은 대부분의 응용프로그램에 충분합니다.

표 17. db2dsdriver 구성 파일의 워크로드 밸런싱 설정

매개변수	설명
maxTransports	전송 풀에 최대 전송 수를 지정합니다. 디폴트값은 -1(무제한)입니다. 기타 음수 값은 유효하지 않습니다. 0값은 워크로드 밸런싱을 사용할 수 없습니다.
maxTransportIdleTime	유휴 전송을 삭제하기 전의 최대 경과 시간을 초 단위로 지정합니다. 디폴트값은 600입니다. 최소 지원값은 0입니다.

표 17. db2dsdriver 구성 파일의 워크로드 밸런싱 설정 (계속)

매개변수	설명
maxTransportWaitTime	클라이언트가 전송이 가능하게 될 때까지 기다리는 초 단위 시간을 지정합니다. 디폴트값은 -1(무제한)입니다. 최소 지원 값은 0입니다.
maxRefreshInterval	서버 목록을 새로 고치기 전의 최소 경과 시간을 초 단위로 지정합니다. 디폴트값은 30입니다. 최소 지원 값은 0입니다.

자세한 정보는 DB2 정보 센터의 db2dsdriver 구성 파일에 대한 주제를 참조하십시오.

주: MAX_COORDAGENTS, NUM_POOLAGENTS 및 MAX_CONNECTIONS 과 같은 기존의 데이터베이스 관리 프로그램 구성 설정은 DB2 Connect 서버에서만 워크로드 밸런싱 구성에 사용되므로 클라이언트 측에 수행되는 워크로드 밸런싱에는 영향을 미치지 않습니다.

예

예 1: 기본 설정을 사용하여 데이터베이스 SAMPLE의 트랜잭션 레벨 워크로드 밸런싱을 사용 가능하게 합니다.

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
</database>
```

예 2: 최대 전송을 95로 하여 데이터베이스 SAMPLE의 트랜잭션 레벨 워크로드 밸런싱을 사용 가능하게 합니다.

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
    <!-- maxTransports represents the maximum number of transports;
    defaults to -1 (unlimited) -->
    <parameter name="maxTransports" value="95" />
  </WLB>
</database>
```

예 3: 최대 전송을 80으로 하고 사용 가능한 전송의 최대 대기 시간을 10초로 하여 데이터베이스 SAMPLE의 트랜잭션 레벨 워크로드 밸런싱을 사용 가능하게 합니다.

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
```

```

<parameter name="enableWLB" value="true" />
<!-- maxTransports represents the maximum number of transports;
defaults to -1 (unlimited) -->
<parameter name="maxTransports" value="80" />
<!-- maxTransportWaitTime represents the maximum elapsed time in
number of seconds before a transport becomes available;
defaults to -1 (unlimited)-->
<parameter name="maxTransportWaitTime" value="10" />
</WLB>
</database>

```

예 4: 데이터베이스 SMPLE의 트랜잭션 레벨 워크로드 밸런싱을 사용 불가능하게 합니다.

```

<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is already disabled by default -->
    <parameter name="enableWLB" value="false" />
  </WLB>
</database>

```

자동 클라이언트 리라우트(클라이언트 측)

자동 클라이언트 리라우트(ACR) 기능은 IBM Data Server Client가 z/OS용 DB2 Sysplex의 구성원에 대한 연결을 상실한 경우 장애 지원을 제공합니다. ACR을 사용하면 클라이언트가 Sysplex의 사용 가능한 구성원을 통해 데이터베이스에 재연결을 시도하여 장애를 복구할 수 있습니다.

ACR은 워크로드 밸런싱이 사용 가능하면 디폴트로 사용할 수 있습니다.

클라이언트 측 ACR 지원은 IBM Data Server Client 및 DB2 Connect 라이선스가 있는 비Java Data Server Driver에 사용할 수 있습니다. ACR 수행에 DB2 Connect 서버는 필요하지 않습니다.

ACR이 사용 가능한 경우 클라이언트에게 기존 연결과의 연결 장애가 일어나면 일반적으로 다음 프로세스가 발생합니다.

1. 클라이언트가 기존 연결을 사용하여 SQL문 실행을 시도하는 경우 장애가 발생합니다.
2. 클라이언트가 서버 목록을 사용하여 액세스할 Sysplex를 식별하고 데이터베이스에 재연결을 시도합니다. 이 서버 목록이 서버에 의해 리턴된 목록이 되거나 클라이언트 친화도가 지정된 경우 클라이언트에 의해 지정된 대체 서버 목록이 될 수 있습니다. 자세한 정보는 139 페이지의 『클라이언트 친화도 구성』을 참조하십시오.
3. ACR이 데이터베이스에 재연결할 수 있는 경우 클라이언트가 새로 설정된 연결에 대한 실행 환경을 재구성합니다. 오류 SQL30108N이 응용프로그램에 리턴되어 실행

패한 데이터베이스 연결의 복구 여부 및 트랜잭션 롤백 여부를 표시합니다. 그러면 응용프로그램이 롤백된 작업을 반복하는 것을 포함하여 그 이상의 복구를 담당합니다.

4. ACR이 데이터베이스에 재연결할 수 없는 경우 오류 SQL30081N이 응용프로그램에 리턴됩니다. 그러면 응용프로그램이 연결 장애 복구를 담당합니다. (예: 스스로 데이터베이스에 연결 시도)

또한 ACR은 클라이언트에게 새 연결과의 연결 장애가 발생한 경우에도 사용됩니다. 그러나 이 경우 오류 SQL30108N이 응용프로그램에 리턴되어 실패한 데이터베이스 연결이 복구되었음을 표시하지 않습니다. 연결이 완료되거나 오류 SQL30081N이 리턴됩니다.

db2dsdriver 구성 파일에 설정을 지정하여 ACR을 사용할 수 없게 하거나 추가 구성을 할 수 있습니다.

CLI 및 .NET 응용프로그램의 무결절성 장애 복구

ACR이 사용 가능하고 트랜잭션의 목표가 z/OS용 DB2이면 CLI 및 .NET 응용프로그램의 무결절성 장애 복구를 디폴트로 사용할 수 있습니다. 무결절성 장애 복구로 트랜잭션의 최초의 SQL 조작 시 응용프로그램에 연결성 장애가 발생하는 경우 드라이버가 자동 클라이언트 리라우트 처리의 일부로 실패한 SQL 조작을 재생합니다. 연결이 완료되면 응용프로그램에 오류가 보고되지 않고 트랜잭션이 롤백되지 않습니다. 연결성 장애 및 연속 복구를 응용프로그램에서 숨깁니다.

db2dsdriver 구성 파일에 설정을 지정하여 무결절성 장애 복구를 사용 불가능하게 하거나 다시 사용 가능하게 할 수 있습니다.

ACR 및 무결절성 장애 복구의 DB2 Connect 서버 고려사항

DB2 Connect 서버를 통해 Sysplex에 액세스하는 경우 ACR 및 무결절성 장애 복구의 동작이 다릅니다.

DB2 Connect 서버와 Sysplex 간에 장애가 발생하는 경우, ACR은 DB2 Connect 서버에서 수행됩니다. DB2 Connect 서버의 레벨이 클라이언트와 같거나 높은 경우 클라이언트가 무결절성 장애 복구를 수행할 수 있습니다. 그렇지 않은 경우 클라이언트가 무결절성 장애 복구를 수행할 수 없으므로 오류 SQL30108N이 응용프로그램에 리턴되어 실패한 데이터베이스 연결이 복구되었음을 표시합니다.

클라이언트와 DB2 Connect 서버 간에 장애가 발생하는 경우, DB2 Connect 서버에 대한 클라이언트에서 ACR을 수행할 수 있습니다. 그러나 무결절성 장애 복구를 항상 사용할 수 없으므로 오류 SQL30108N이 응용프로그램에 리턴됩니다.

자동 클라이언트 리라우트 구성(클라이언트 측)

워크로드 밸런싱이 사용 가능한 경우 자동 클라이언트 리라우트(ACR)를 디폴트로 사용할 수 있습니다. db2dsdriver 구성 파일에 설정을 지정하여 ACR을 사용할 수 없게 하거나 추가 구성을 할 수 있습니다.

시작하기 전에

ACR을 수행하려면, 클라이언트는 TCP/IP 연결을 사용해야 하며 DB2 Connect 라이선스가 있어야 합니다. 다음 클라이언트는 ACR에 대한 지원을 제공합니다.

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver Package
- IBM Data Server Driver for ODBC and CLI

이 태스크에 대한 정보

일부 응용프로그램의 경우 ACR 또는 무결절성 장애 복구를 사용 불가능하게 하거나 ACR을 추가로 구성하고자 할 수 있습니다. 이 태스크는 ACR 구성에 사용할 수 있는 매개변수에 대해 설명합니다.

클라이언트 친화도를 지정하여 ACR의 동작을 변경해서 연결 처리 및 ACR에 사용되는 서버 목록을 제어할 수도 있습니다. 정보는 139 페이지의 『클라이언트 친화도 구성』을 참조하십시오.

프로시저

ACR을 사용 불가능하게 하거나 추가로 구성하려면 다음을 수행하십시오.

- ACR 또는 무결절성 장애 복구가 있는 ACR을 사용 불가능하게 하려면 db2dsdriver 구성 파일의 다음 매개변수의 값을 "거짓"으로 지정하십시오. 데이터베이스 항목의 ACR 서브섹션에 이 설정을 지정하십시오.

표 18. db2dsdriver 구성 파일의 자동 클라이언트 리라우트 설정

매개변수	설명
enableAcr	ACR 기능의 설정(참) 또는 해제(거짓) 여부를 지정합니다. 트랜잭션 레벨 워크로드 밸런싱이 사용 가능한 경우 이 설정 기본값은 참(enableWLB 설정이 참)입니다. ACR이 사용 불가능한 경우 ACR의 기타 설정은 무시됩니다.
enableSeamlessAcr	무결절성 장애 복구 기능의 설정(참) 또는 해제(거짓) 여부를 지정합니다. ACR이 사용 가능하고 목표 서버가 z/OS용 DB2인 경우 무결절성 ACR을 디폴트로 사용할 수 있습니다. 목표 서버가 z/OS용 DB2인 경우 이 설정은 무시되며 무결절성 ACR이 지원되지 않습니다.

자세한 정보는 DB2 정보 센터의 db2dsdriver 구성 파일에 대한 주제를 참조하십시오.

- 최초 연결 시도에 장애가 발생할 경우 사용할 대체 서버 목록을 지정할 수 있으며 연결에 지정된 호스트를 사용할 수 없습니다. 대체 서버 목록을 지정하려면 데이터베이스 항목의 ACR 서브섹션의 enableAlternateServerListFirstConnect의 값을 "참"으로 지정하십시오. enableAlternateServerListFirstConnect의 값이 "참"인 경우 무결절성 장애 복구가 있는 ACR은 db2dsdriver 구성 파일의 ACR에 지정된 기타 설정에 관계없이 내재적으로 사용 가능합니다. 이 기능을 사용하려면 db2dsdriver 구성 파일에 대체 서버 목록도 설정해야 합니다.

대체 서버 목록 설정에 대한 정보는 139 페이지의 『클라이언트 친화도 구성』을 참조하십시오. 대체 서버 목록 설정에 대한 정보가 클라이언트 친화도를 설명하는 주제에 표시되더라도 이 성능을 사용하기 위해 클라이언트 친화도를 사용할 필요가 없습니다. 사실 이 성능은 트랜잭션 레벨 워크로드 밸런싱 및 클라이언트 친화도를 동시에 사용할 수 없지만 트랜잭션 레벨 워크로드 밸런싱이 사용 가능한 경우에도 지원됩니다.

결과

ACR이 사용 가능한 경우 연결성 장애가 발생하는 응용프로그램은 Sysplex의 다른 구성원에게 라우트됩니다. 무결절성 장애 복구를 사용 할 수 없거나 적용할 수 없는 경우 오류 SQL30108N이 응용프로그램에 리턴됩니다.

트랜잭션의 최초 SQL 조작 시 장애가 발생하고 무결절성 장애 복구를 사용(및 적용)할 수 있는 경우 클라이언트가 ACR 처리의 일부로 실패한 SQL 조작에 응답할 수 있습니다. 연결이 완료되면 응용프로그램에 오류가 보고되지 않습니다.

예

예 1: 무결절성 장애 복구를 사용 할 수 없는 데이터베이스 SAMPLE의 ACR을 사용할 수 있음

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
  <ACR>
    <!-- ACR is already enabled by default when WLB is enabled -->
    <parameter name="enableAcr" value="true" />
    <!-- If ACR is enabled and the target server is DB2 for z/OS,
    Seamless ACR is enabled by default; if the target server is
    not DB2 for z/OS, this setting is ignored and seamless ACR
    is not supported. -->
    <parameter name="enableSeamlessAcr" value="false" />
  </ACR>
</database>
```


예 2: 데이터베이스 SAMPLE의 ACR을 사용할 수 없음

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <WLB>
    <!-- Sysplex WLB is disabled by default -->
    <parameter name="enableWLB" value="true" />
  </WLB>
  <ACR>
    <!-- ACR is enabled by default when WLB is enabled -->
    <parameter name="enableAcr" value="false" />
  </ACR>
</database>
```

예 3: 원래 호스트를 사용할 수 없는 경우 최초 연결 시간에 대체 서버 목록을 사용하도록 지정

```
<database name="SAMPLE" host="v33ec065.svl.ibm.com" port="446">
  <ACR>
    <parameter name="enableACR" value="true"/>
    <parameter name="enableAlternateServerListFirstConnect"
      value="true"/>
  </ACR>
</database>
```

XA 지원(클라이언트 측)

IBM 데이터 서버 및 DB2 Connect 라이선스가 있는 비Java 데이터 서버 드라이버가 z/OS용 DB2 Sysplex에 직접 액세스할 수 있으며 미들 티어 DB2 Connect 서버를 통하지 않고 원시 XA 지원을 사용할 수 있습니다.

이 유형의 클라이언트 측 XA 지원은 단일 전송 처리 모델을 사용하는 트랜잭션 관리 프로그램만 사용할 수 있습니다. 단일 전송 모델에서 단일 전송(물리적 연결)을 통한 트랜잭션은 xa_start부터 xa_end까지의 구성원에 연관됩니다. xa_prepare(readonly), xa_prepare 및 xa_commit 또는 xa_rollback, 또는 xa_rollback 직후 트랜잭션이 종료됩니다. 이것은 모두 단일 응용프로그램 프로세스 내에서 발생해야 합니다. 이 모델을 사용하는 트랜잭션 관리 프로그램의 예에는 IBM TXSeries® CICS, IBM WebSphere Application Server 및 Microsoft Distributed Transaction Coordinator가 포함됩니다.

단일 전송 처리 모델의 지원에는 각각의 복구 가능 트랜잭션의 구성원 정보가 xa_recover를 통해 검색되는 인다우트(indoubt) 트랜잭션 복구도 포함되며 이것은 xa_commit 또는 xa_rollback이 지정된 구성원에서 방향지정되도록 합니다.

xa_open 문자열에 SINGLE_PROCESS 매개변수를 사용하거나 db2dsdriver 구성 파일에 XA의 설정을 지정하여 XA 지원을 사용할 수 있습니다.

중요사항: DB2 APAR PK69659는 직접 XA 지원에 설치되어야 합니다. (이것은 Microsoft Distributed Transaction Coordinator와 같은 트랜잭션 관리 프로그램에 필요합니다.) 자세한 정보는 APAR PK69659를 참조하십시오.

XA 지원 사용(클라이언트 측)

z/OS용 DB2 Sysplex에 액세스하는 클라이언트가 XA 지원을 사용하도록 하려면 db2dsdriver 구성 파일에 설정을 지정하거나 xa_open 문자열의 SINGLE_PROCESS 매개변수를 사용합니다.

시작하기 전에

z/OS용 DB2 Sysplex에 액세스할 때 DB2 Connect 라이선스가 필요합니다.

다음 클라이언트는 z/OS용 DB2 Sysplex에 액세스하는 응용프로그램의 XA 지원을 제공합니다.

- IBM Data Server Client
- IBM Data Server Runtime Client
- IBM Data Server Driver Package
- IBM Data Server Driver for ODBC and CLI

중요사항: DB2 APAR PK69659는 직접 XA 지원에 설치되어야 합니다. (이것은 Microsoft Distributed Transaction Coordinator와 같은 트랜잭션 관리 프로그램에 필요합니다.) 자세한 정보는 APAR PK69659를 참조하십시오.

이 태스크에 대한 정보

이 태스크는 IBM Data Server Client 및 비Java Data Server Driver의 XA 지원을 사용 가능하도록 합니다.

제한사항

XA 지원은 단일 전송 처리 모델을 사용하는 트랜잭션 관리 프로그램만 사용할 수 있습니다. 이 제한사항에 대한 자세한 정보는 클라이언트 한계에 대한 주제를 참조하십시오.

프로시저

1. 인스턴스 기반 클라이언트(IBM Data Server Client)의 경우 db2dsdriver 구성 파일에 enableDirectXA 매개변수를 설정하거나 xa_open 문자열의 SINGLE_PROCESS 매개변수를 사용하여 XA 지원 설정(참) 또는 해제(거짓) 여부를 지정하십시오.
2. 인스턴스가 없는 클라이언트의 경우(IBM 데이터 서버 드라이버) XA 지원을 Microsoft Distributed Transaction Coordinator 또는 Microsoft Component Services (COM+)에 디폴트로 사용 할 수 있습니다. 기타 다른 지원된 트랜잭션 관리 프로그램의 경우 xa_open 문자열에 SINGLE_PROCESS 키워드를 설정하여 XA 지원 사용 가능 여부를 지정하십시오. db2dsdriver 구성 파일의 enableDirectXA에 대한 설정은 인스턴스가 없는 클라이언트에 적용할 수 없습니다.

결과

포트를 사용할 수 있는 경우 응용프로그램이 미들 티어 DB2 Connect 서버를 통하지 않고 단일 응용프로그램 프로세스 내의 단일 전송을 통해 분산된 트랜잭션을 실행할 수 있습니다.

예

데이터베이스 SAMPLE의 단일 전송 XA 지원 사용 가능

```
<database name="SAMPLE" host="v33ec065.my.domain.com" port="446">
  <!-- database-specific parameters -->
  <!--directXA is disabled by default -->
    <parameter name="enableDirectXA" value="true" />
  </parameters>
</database>
```

클라이언트 친화도 구성

클라이언트 친화도를 구성하여 초기 연결 처리 및 자동 클라이언트 리라우트(ACR) 처리 중에 시도한 서버의 순서를 제어할 수 있습니다. 트랜잭션 레벨 워크로드 밸런싱을 명확하게 사용할 수 없는 경우 클라이언트 친화도를 사용할 수 있습니다.

이 태스크에 대한 정보

일반적으로 클라이언트 Sysplex 성능을 사용하는 응용프로그램은 데이터베이스 서버에 의해 리턴된 최근 서버 목록을 사용하여 장애 중에 연결할 구성원을 판별합니다. 이 모델에서 클라이언트가 마지막으로 읽은 서버 목록을 기반으로 리라우트를 하므로 재시도할 서버의 순서가 서버에 의해 제어됩니다. 그러나 클라이언트 친화도를 구성하거나 대체 서버 목록을 지정하여 이 동작을 겹쳐써서 초기 연결 처리 중에 그리고 클라이언트에 의해 수행되는 클라이언트 리라우트 처리에 사용할 수 있습니다. 클라이언트 친화도 처리를 사용 중이면 ACR 처리가 시도된 서버의 순서를 클라이언트가 제어할 수 있습니다.

클라이언트 친화도를 구성하려면 db2dsdriver 구성 파일에

ALTERNATE_SERVER_LIST, AFFINITY_LIST 및 CLIENT_AFFINITY의 설정을 지정하십시오. 이 설정이 지정되면 다음이 수행됩니다.

- 이 연결은 응용프로그램 지정 호스트 및 포트에 반대되는 대체 서버 목록을 기반으로 설정됩니다. 예를 들어 응용프로그램이 Host = s1로 지정하고 대체 서버 목록이 Serverlist = (s2, s1, s3)으로 지정하는 경우 초기 연결은 s2에서는 시도되지만 s1에서는 시도되지 않습니다. 연결 시도 순서는 (s2,s2,s2) (s1,s1,s1) (s3,s3,s3)입니다. acrRetryInterval이 개별 연결 시도 간에 적용됩니다. (이 예에서는 총 8회입니다.)
- 무결결성 클라이언트 리라우트가 자동으로 사용 가능하게 됩니다. 클라이언트 리라우트는 ALTERNATE_SERVER_LIST 그룹의 서버 및 포트 매개변수에 지정된 서버에 연결하여 수행됩니다. 대체 목록의 다음 서버를 사용하기 전에 대체 목록의 각 서

버가 maxAcrRetries 회수 만큼 시도되고 이것은 전체 목록이 소모되거나 완료된 연결이 설정될 때까지 계속됩니다. 예를 들어 대체 서버 목록이 (s3,s1,s2)로 정의되고 maxAcrRetries가 3으로 정의되면 리라우트 순서는 (s3,s3,s3) (s1,s1,s1) (s2,s2,s2)가 됩니다. acrRetryInterval이 각각의 개별 연결 시도 간에 적용됩니다. (이 예에서는 총 8회입니다.)

프로시저

db2dsdriver 구성 파일에 클라이언트 친화도 설정을 지정하려면 다음을 수행하십시오.

1. ACR 그룹 밑의 db2dsdriver 구성 파일에서 ALTERNATE_SERVER_LIST의 항목을 지정하여 대체 서버의 목록을 선언하십시오. 이 목록에 각 서버의 포트 번호 및 호스트 이름을 지정하십시오. 예를 들어, 다음과 같습니다.

```
<database name="SAMPLE" host="v33ec065.svl.ibm.com" port="446">
  <acr>
    <parameter name="enableACR" value="true"/>
    <alternate_server_list>
      <server name="server1"
        hostname="v33ec067.svl.ibm.com"
        port="446" >
      </server>
      <server name="server2"
        hostname="v33ec066.svl.ibm.com"
        port="446" >
      </server>
      <server name="server3"
        hostname="v33ec065.svl.ibm.com"
        port="446" >
      </server>
    </alternate_server_list>
    .
    .
    .
```

이 목록의 지정은 단순히 대체 서버를 선언하는 것이며 동작에 변경을 가져오지 않습니다.

2. ACR 그룹 밑에서 AFFINITY_LIST의 항목을 지정하여 서버의 순서를 식별하십시오. 예를 들어, 다음과 같습니다.

```
.
.
.
<affinity_list>
  <list name="list1"
    serverorder="server1,server2,server3" >
  </list>
  <list name="list2"
    serverorder="server3,server2,server1" >
  </list>
</affinity_list>
```

·
·
·

이 목록의 지정은 단순히 서버의 순서를 식별하는 것이며 동작에 변경을 가져오지 않습니다.

3. ACR 그룹 밑에서 다음 클라이언트 친화도 서브그룹 중 하나를 지정하여 클라이언트 친화도를 사용 가능하게 하십시오. 이 서브그룹 중 하나를 지정하면 클라이언트 친화도를 사용 가능하도록 강제 실행됩니다. 이 데이터베이스에 대한 모든 클라이언트 연결은 CLIENT_AFFINITY 서브그룹 중 하나에 지정되어야 합니다. 이 서브그룹 중 하나에서 클라이언트를 찾을 수 없는 경우 연결 시도 중에 오류가 발생합니다. CLIENT_AFFINITY 서브 그룹이 존재하면 ACR을 내재적으로 사용할 수 있습니다.

- CLIENT_AFFINITY_DEFINED

클라이언트 호스트 이름에서 특정 AFFINITY_LIST 요소로의 특정 맵핑을 지정합니다. 클라이언트 호스트 이름은 자동으로 찾을 수 있으며 친화도 목록을 계산할 db2dsdriver 구성 파일 항목과 일치합니다. 예를 들어, 다음과 같습니다.

```
·  
·  
·  
<client_affinity_defined>  
  <!-- this section has specific defined affinities  
  -->  
  <client name="client1"  
    hostname="appsrv1.svl.ibm.com"  
    listname="list2" >  
  </client>  
  <client name="client2"  
    hostname="appsrv2.svl.ibm.com"  
    listname="list1" >  
  </client>  
</client_affinity_defined>  
·  
·  
·
```

- CLIENT_AFFINITY_ROUNDROBIN

ALTERNATE_SERVER_LIST에 라운드 로빈 지정을 지정하십시오. 이 지정에는 CLIENT_AFFINITY_ROUNDROBIN 목록의 클라이언트 인덱스(영(0) 기반)와 같은 시작 서버 세트가 있습니다. 예를 들어,

```
·  
·  
·  
<client_affinity_roundrobin>  
  <!-- roundrobin picks the starting server as  
  the client index number in this section (0 based)  
  modulo the number of servers.
```

```

-->
<client name="client3"
  hostname="appsrv3.svl.ibm.com" >
  <!-- this entry is index 0, modulo 3, thus it gets:
    server1, server2, server3
  -->
</client>
<client name="client4"
  hostname="appsrv4.svl.ibm.com" >
  <!-- this entry is index 1, modulo 3, thus it gets:
    server2, server3, server1
  -->
</client>
</client_affinity_roundrobin>
.
.
.

```

결과

다음 고려사항 및 유효성 확인 규칙은 클라이언트 친화도 처리에 적용됩니다.

- CLIENT_AFFINITY가 사용 가능한 경우 (즉, 하나 이상의 CLIENT_AFFINITY 서브그룹이 db2dsdriver 구성 파일에 존재하는 경우) maxAcrRetries의 디폴트값은 3입니다. 이 값의 유효 범위는 0에서 MAX_INT입니다. 지정된 값이 0이면 maxAcrRetries가 1로 지정된 경우와 마찬가지로 ACR 처리가 발생합니다.
- acrRetryInterval의 디폴트값은 0입니다. 이 값의 유효 범위는 0에서 MAX_INT입니다. 값이 0이면 연결 재시도 간에 또는 드라이버의 ACR 시도 간에 갭이 없습니다.
- 주어진 클라이언트의 대체 서버를 규정하는 총 수는 24를 초과할 수 없습니다. 대체 서버 규정 수가 24보다 크면 SQL1042N이 발생합니다.
- enableWLB 및 CLIENT_AFFINITY 설정은 서로 양립할 수 없습니다. enableWLB 및 CLIENT_AFFINITY가 모두 사용 가능하면 오류 SQL5162N이 발생합니다.
- 필요한 속성은 ALTERNATE_SERVER_LIST, AFFINITY_LIST 및 CLIENT_AFFINITY 그룹에 지정되어야 합니다. 그렇지 않은 경우 SQL5163N이 발생합니다.
- CLIENT_AFFINITY가 사용 가능한 경우 ALTERNATE_SERVER_LIST는 비어 있으면 안됩니다. 그렇지 않은 경우 SQL5164N이 발생합니다.
- CLIENT_AFFINITY가 사용 가능한 경우 클라이언트의 호스트 이름이 유효한 대체 서버 목록에 들어가야 합니다. 그렇지 않은 경우 SQL5164N이 발생합니다. 예를 들어 클라이언트의 호스트 이름이 "xyz.domain.com,"이고 호스트 이름을 CLIENT_AFFINITY 서브그룹 중 하나에서 찾을 수 없는 경우 SQL5164N이 발생합니다. 그리고 호스트 이름이 CLIENT_AFFINITY 그룹 중 하나에만 표시될 수 있습니다.

예

다음 예는 CLIENT_AFFINITY 서브그룹이 지정된 db2dsdriver 구성의 샘플 데이터베이스 섹션을 나타냅니다.

```
<database name="SAMPLE" host="v33ec065.svl.ibm.com" port="446">
  <acr>
    <parameter name="enableACR" value="true"/>
    <alternate_server_list>
      <server name="server1"
        hostname="v33ec067.svl.ibm.com"
        port="446" >
      </server>
      <server name="server2"
        hostname="v33ec066.svl.ibm.com"
        port="446" >
      </server>
      <server name="server3"
        hostname="v33ec065.svl.ibm.com"
        port="446" >
      </server>
    </alternate_server_list>
    <affinity_list>
      <list name="list1"
        serverorder="server1,server2,server3" >
      </list>
      <list name="list2"
        serverorder="server3,server2,server1" >
      </list>
    </affinity_list>
    <client_affinity_defined>
      <!-- this section has specific defined affinities
      -->
      <client name="client1"
        hostname="appsrv1.svl.ibm.com"
        listname="list2" >
      </client>
      <client name="client2"
        hostname="appsrv2.svl.ibm.com"
        listname="list1" >
      </client>
    </client_affinity_defined>
    <client_affinity_roundrobin>
      <!-- roundrobin picks the starting server as
        the client index number in this section (0 based)
        modulo the number of servers.
      -->
      <client name="client3"
        hostname="appsrv3.svl.ibm.com" >
        <!-- this entry is index 0, modulo 3, thus it gets:
          server1, server2, server3
        -->
      </client>
      <client name="client4"
        hostname="appsrv4.svl.ibm.com" >
        <!-- this entry is index 1, modulo 3, thus it gets:
          server2, server3, server1
        -->
    </client_affinity_roundrobin>
  </acr>
</database>
```

```

        </client>
    </client_affinity_roundrobin>
</acr>
</database>

```

클라이언트 Sysplex 지원 사용의 한계

일부 한계를 IBM Data Server Client 및 비Java Data Server Driver에 사용할 수 있는 클라이언트 Sysplex 지원에 적용합니다.

- 트랜잭션 레벨 워크로드 밸런싱의 한계
 - 클라이언트는 TCP/IP 연결을 사용해야 합니다.
 - 클라이언트 Sysplex 지원을 사용하는 IBM Data Server Client 및 Driver의 경우 서버 목록이 응용프로그램 프로세스 내에 캐시됩니다. 따라서 목록은 하나 이상의 스레드에서 해당 워크로드 밸런스 프로세스 내의 연결에 의해서만 공유됩니다.
- 자동 클라이언트 리라우트 한계:
 - 클라이언트는 TCP/IP 연결을 사용해야 합니다.
 - Sysplex 서버가 이전 트랜잭션 종료시 전송 재사용을 허용하지 않는 경우 트랜잭션의 최초 SQL문에 대한 무결절성 장애 복구는 지원되지 않습니다. 전송 재사용이 KEEP DYNAMIC(YES)의 결과로서 부여되지 않는 경우 예외가 있습니다.
 - SQL문에 응답하기 위하여 CLI 드라이버가 이전에 호출한 API의 실행기록을 유지해야 하는 경우 트랜잭션의 최초 SQL문에 대한 무결절성 장애 복구는 지원되지 않습니다. 이것이 발생할 수 있는 경우 시나리오는 실행 시간에 데이터 지정, 복합 SQL 수행 및 배열 입력 사용으로 구성됩니다.
 - 자동 커미트를 사용할 수 있는 경우 무결절성 장애 복구가 발생하면 SQL문을 실행하여 여러 번 커미트할 수 있습니다. 원래 자동 커미트가 성공적으로 완료된 후 자동 클라이언트 리라우트를 트리거하는 이벤트가 발생하는 경우 이러한 결과가 일어날 수 있습니다. 이 동작을 피하려면 자동 커미트 또는 무결절성 장애 복구 중 하나를 사용하지 마십시오.
 - 스토어드 프로시저의 내부에 무결절성 장애 복구를 적용할 수 없습니다.
 - 페더레이티드 서버 환경에 무결절성 장애 복구를 적용할 수 없습니다.
- XA 지원:
 - 다음 트랜잭션 관리 프로그램 프로세싱 모델은 지원되지 않습니다.
 - 이중 트랜스포트 이 모델에서 트랜스포트 A를 통한 트랜잭션은 xa_start부터 xa_end까지의 한 구성원에 관련되거나 xa_prepare(readonly), xa_prepare 및 xa_commit 또는 xa_rollback, 또는 xa_rollback은 다른 응용프로그램 프로세스에서 트랜스포트 B를 통해 전달됩니다. 이 모델을 사용하는 트랜잭션 관리 프로그램의 예는 IBM WebSphere MQ 및 IBM Lotus Domino입니다.

- 다중 트랜스포트. 이 모델에서는 같은 트랜잭션에 사용되는 다중 응용프로그램 프로세스의 다중 전송을 사용합니다. 이 모델을 사용하는 트랜잭션 관리 프로그램의 예는 BEA Tuxedo 및 IBM TXSeries Encina 모니터입니다.
- 다중 전송 처리 모델을 사용하는 XA 트랜잭션 관리 프로그램의 경우 미들 티어 DB2 Connect 서버가 계속 필요합니다.

제 20 장 DB2 Connect 조정

데이터베이스 관리 프로그램 구성 파일의 다양한 매개변수를 사용하여 DB2 Connect 를 조정할 수 있습니다.

RQRIOBLK

RQRIOBLK 매개변수는 네트워크 입출력 블록의 최대 크기를 설정합니다. 블록 크기를 더 크게 설정하면 큰 요청의 성능이 향상됩니다. 블록 크기는 한 행의 데이터 요청과 같은 작은 요청에 대한 응답 시간에는 일반적으로 영향을 주지 않습니다.

블록 크기가 더 크면 일반적으로 DB2 Connect 서버에서 더 많은 메모리가 필요합니다. 그러면, 작업 세트 크기가 증가되고 작은 워크스테이션에서 페이지징 양이 많아질 수 있습니다.

응용프로그램 실행 시 너무 많은 페이지징이 발생하지 않는 경우, 디폴트 DRDA 블록 크기(32767)를 사용하십시오. 그렇지 않으면, 페이지징이 없어질 때까지 입출력 블록 크기를 줄이십시오. 페이지징이 시작되면, 성능이 급격히 저하됩니다. 성능 모니터링 도구(예: Linux 및 UNIX 운영 체제의 경우 vmstat 도구)를 사용하여 시스템에 페이지징이 발생하는지 여부를 판별하십시오.

DIR_CACHE

DIR_CACHE 매개변수는 디렉토리 정보의 캐시 여부를 판별합니다. 캐싱을 사용하면(DIR_CACHE=YES), 디렉토리 파일을 읽어 메모리에 캐시하므로 연결할 때마다 내부 디렉토리 구조를 작성하고 디렉토리 파일을 읽는 오버헤드를 최소화할 수 있습니다.

캐싱을 사용하지 않으면(DIR_CACHE=NO), 데이터베이스에 연결할 때마다 디스크에서 해당 디렉토리를 읽은 후 검색을 수행합니다. 요청된 항목을 찾으면, 모든 디렉토리 관련 메모리 검색이 해제됩니다.

캐싱할 경우, db2start 처리 중에 공유 디렉토리 캐시가 작성되고 DB2가 중지될 때 비워집니다. 이 캐시는 모든 DB2 서버 프로세스(db2agent)에서 사용됩니다. 또한, 개인용 응용프로그램 디렉토리 캐시는 응용프로그램이 데이터베이스로 첫 번째 연결할 때 구축되고 응용프로그램이 종료하면 해제됩니다.

캐시마다 시스템 데이터베이스 디렉토리, 데이터베이스 연결 서비스 디렉토리, 노드 디렉토리의 이미지를 제공합니다. 캐시를 사용하면 디렉토리 파일 입출력을 제거하고 디렉토리 검색을 최소화하므로 연결에 필요한 자원이 줄어듭니다.

캐시된 디렉토리가 갱신되는 경우, 변경사항이 즉시 캐시로 전달되지 않습니다. 디렉토리 항목이 캐시에 없는 경우, 원래 디렉토리에서 검색됩니다.

캐싱을 사용하면 응용프로그램 라이프에 필요한 개인용 메모리를 증가시킵니다. 캐싱이 없으면, 이 메모리는 디렉토리 찾아보기가 처리 중일 때만 필요합니다. 데이터베이스 에이전트 간에 공유되는 디렉토리 정보가 공유 메모리로 이동하므로 DB2에 의한 공유 메모리의 전체 사용이 약간 증가합니다. 캐시에 필요한 메모리 크기는 디렉토리마다 정의된 항목 수에 따라 결정됩니다.

NUMDB

이전 버전에서는 DB2 Connect의 동작이 NUMDB 구성 매개변수의 영향을 받지 않았지만, 버전 8부터는 변경되었습니다. 이 매개변수는 클라이언트가 DB2 Connect 서버를 통해 연결할 수 있는 최대 데이터베이스 수를 나타냅니다. 더 명확하게 말하면, DB2 Connect 서버에서 카탈로그화할 수 있는 서로 다른 데이터베이스 별명의 최대 수입니다.

기타 DB2 Connect 매개변수

AGENTPRI 및 MAXAGENTS는 버전 9.5에서 사용되지 않습니다.

MAXAGENTS에 대한 값을 갱신하는 명령은 기존 응용프로그램이 중단되지 않도록 계속 작동하지만 그 값은 무시됩니다. 매개변수 이름은 구성 목록에 나타나지 않습니다. 과거에는 지정된 DB2 파티션에 작성할 수 있는 총 에이전트 수가 MAXAGENTS 구성 매개변수를 통해 제어되었습니다. 이제는 에이전트 구성을 자동화할 수 있습니다.

디폴트로 NUM_POOLAGENTS는 디폴트값이 100인 AUTOMATIC으로 설정됩니다. 또한 디폴트로 MAX_COORDAGENTS는 디폴트값이 200인 AUTOMATIC으로 설정됩니다.

클라이언트 응용프로그램에서 DB2 Connect 서버로 어카운팅 문자열을 보내려면 API에 특정한 방법을 사용하여 어카운팅 정보를 설정하십시오. API에 특정한 방법은 DB2ACCOUNT 환경 변수를 설정하는 것보다 더 빠르게 수행됩니다.

IBM Data Server Driver for JDBC and SQLJ

com.ibm.db2.jcc.DB2BaseDataSource.clientAccountingInformation 등록 정보

IBM Data Server Provider for .NET

DB2Connection.ClientAccountingInformation 등록 정보

CLI/ODBC

ClientAcctStr CLI/ODBC 구성 키워드

Embedded SQL(C, C++ 및 COBOL)

sqlsact 함수

조정된 SQLCODE 맵핑 파일이 필요없는 경우, 디폴트 SQLCODE 맵핑을 사용하거나 SQLCODE 맵핑을 작동 중지하여 성능을 향상시킬 수 있습니다. 디폴트 맵핑 파일은 DB2 Connect 라이브러리에 포함되어 있으며, 조정된 맵핑 파일은 디스크에서 읽으므로 성능에 영향을 미칩니다.

호스트 데이터베이스 조정

시스템 성능은 IBM 메인프레임 데이터베이스 서버 성능의 영향을 받습니다. 다른 데이터베이스 관리 시스템에는 서로 다른 성능 기능이 있습니다. 예를 들어, 다른 시스템의 SQL 옵티마이저는 같은 응용프로그램으로 다르게 작동할 수 있습니다. 자세한 정보는 IBM 메인프레임 데이터베이스 서버 시스템 성능 문서를 확인하십시오.

가능할 경우, 저널링을 방지하기 위해 언커미트 읽기(UR) 또는 커미트 안함(NC) 바인드 옵션을 사용하여 성능을 향상시킬 수 있습니다.

주: UR을 사용하면 저널링되지 않은 데이터를 읽을 수만 있고 갱신할 수는 없으며, 이는 블로킹이 ALL로 설정되어 있는 경우에만 가능합니다.

응용프로그램 서버 및 응용프로그램 서버가 제공하는 잠금 세분성에 따라 쿼리 또는 응용프로그램에 사용되는 분리 레벨이 성능에 큰 영향을 미칠 수 있습니다. 데이터베이스에는 적합한 표준화의 레벨, 효율적인 인덱스 사용, 데이터베이스 공간의 적당한 할당이 있어야 합니다. 성능은 다음 절에 설명된 사용하는 데이터 유형의 영향을 받을 수 있습니다.

네트워크 조정 고려사항

분산 데이터베이스 환경에서 전반적인 성능을 향상하는 가장 좋은 방법은 네트워크에서 지연을 제거하는 것입니다. 네트워크 관리자는 보통 전송시 가능한 많은 데이터가 수집될 경우 네트워크가 좀 더 효율적이 되는 것으로 간주합니다. 그러나 이러한 접근 방식은 네트워크의 지연을 가져오는 분산 데이터베이스와 같은 응용프로그램에는 적합하지 않습니다. 일반 사용자는 지연 때문에 네트워크의 효율성을 깨닫지 못합니다.

대부분의 네트워크 디바이스는 지연 매개변수를 가지고 있으며 이들 대부분은 분산 데이터베이스에는 좋지 않은 디폴트값으로 설정되어 있습니다. 성능을 개선하려면, 이들 매개변수를 찾아 가능하면 이들을 0으로 설정해야 합니다. 또한, 디바이스의 버퍼 크기가 데이터 유실로 인한 재전송을 방지할 만큼 충분한 공간이 되도록 해야 합니다. 예를 들어, UNIX 시스템은 대개 전송 또는 수신 큐 깊이에 대해 디폴트값 32를 갖습니다. 결과를 더 좋게 하려면, 큐 깊이를 150으로 설정하십시오. DLC 설정에서 이에 상응하는 매개변수는 수신 깊이로, 이것 또한 150이 되어야 합니다.

대부분의 사이트에서 IOBUF 매개변수가 너무 낮게 설정되어 있습니다. 이 매개변수는 대개 500으로 설정되지만, 경험에 의하면 특히, ESCON® 또는 3172와 같은 채널 연결에서 다량의 데이터를 이동하는 경우에 값 3992가 가장 적합합니다.

LAN 시스템에서, DLC 또는 LLC 전송 및 수신 창의 크기가 성능에 커다란 영향을 미칠 수 있습니다. 전송 값을 7 이상으로 설정해야 하고 대다수 구성에서는 수신 값이 4 이하이면 최적으로 작동합니다.

이더넷을 실행하는 경우 TCP 세그먼트 크기를 1500바이트로 설정해야 합니다. 토큰 링 또는 FDDI 네트워크에서 이 값은 4400바이트이어야 하고, TCP/IP와 함께 ESCON 어댑터를 사용하는 경우 세그먼트 크기는 항상 4096이어야 합니다.

마지막으로 TCP/IP 네트워크의 경우, TCP 전송 및 수신 버퍼 크기는 32768 이상으로 설정해야 합니다. 보통 65536값이 가장 좋습니다.

주: 게이트웨이에서 서버로의 연결(아웃바운드 연결)을 설정하는 것은 클라이언트에서 게이트웨이로의 연결(인바운드 연결)을 설정하는 것보다 자원 사용이 훨씬 더 많습니다. 수천개의 클라이언트가 게이트웨이를 통해 서버와 연결 및 연결 끊기를 반복하는 환경에서는 처리 시간의 상당 부분이 아웃바운드 연결을 설정하는 데 사용됩니다. DB2 Connect는 TCP/IP에서 연결 풀링을 제공합니다. 클라이언트가 서버로부터의 연결 끊기를 요청하면, 게이트웨이가 클라이언트와의 인바운드 연결을 해제하지만 서버에 대한 아웃바운드 연결을 풀에 그대로 유지시킵니다. 새 클라이언트가 게이트웨이에서 연결을 요청하면, 게이트웨이는 풀에서 기존의 연결을 제공하므로 전반적인 연결 시간이 줄어들고 서버에서의 CPU 연결을 위한 자원 사용이 절약됩니다.

네트워크 성능 조정 메소드는 표 19에 요약되어 있습니다.

표 19. 네트워크 성능 조정 메소드

확인 대상	예	설정	주의사항
고의 지연	네트워크 디바이스의 지연 매개변수	0으로 설정	보통 디폴트값이 가장 높습니다.
버퍼	IOBUF 매개변수	최대 3992로 설정	ESCON 또는 기타 채널 어댑터에 특히 유용합니다.
버퍼	RUSIZE	최적의 크기는 4096입니다.	RUSIZE와 RQRIOBLK를 같은 크기로 설정하면 최상의 성능을 얻을 수 있습니다.
버퍼	페이싱	VPACING, PACING 및 모드 프로파일을 63으로 설정해야 합니다.	가능할 경우 적응 페이싱을 사용하십시오.
어댑터 설정	전송/수신 큐 깊이	권장 값은 150입니다.	디폴트값은 보통 32입니다.

표 19. 네트워크 성능 조정 메소드 (계속)

확인 대상	예	설정	주의사항
TCP 설정	세그먼트 크기	이더넷의 경우 1500, 토 큰 링 및 FDDI의 경우 4400	TCP/IP에 사용되는 ESCON 어댑터는 항상 4096으로 설정해야 합니다.
TCP 설정	송신/수신 공간 크기	두 경우 모두 64K여야 합니다.	Windows의 경우, 디폴트 값은 8192뿐입니다. Windows 레지스트리에서 설정할 수 있습니다.

시스템 자원 경합

시스템의 많은 태스크에 의해 시스템 자원에 대한 경합이 일어나면 성능이 저하될 수 있습니다. 다음 질문을 고려하십시오.

- CPU가 포화 상태입니까? 처리 오버헤드를 줄이기 위해 시스템을 업그레이드하고, 시스템 워크로드를 줄이며 시스템을 조정하십시오.
- 메모리가 과잉 커밋 되었습니까? 메모리 작업 세트를 줄이기 위해 메모리를 업그레이드하고 시스템 워크로드를 줄이며, 시스템을 조정하십시오.
- 통신 어댑터/통신 제어기를 너무 많이 사용하고 있습니까? 네트워크를 업그레이드하고 토큰링 카드를 쌍으로 만드는 것을 고려하십시오.
- 서브시스템 중 한 시스템을 너무 많이 사용하는데 이 서브시스템이 데이터 경로에 있습니까?
- 시스템에서 실행 중인 불필요한 프로세스나 태스크가 있습니까? 시스템 자원을 낭비하기 때문에 규칙적으로 사용하지 않을 경우, 서비스를 구성하거나 시작하지 말아야 하는 것이 일반적인 규칙입니다.
- 프로세스나 태스크가 대부분의 자원을 사용합니까? 중단될 수 있습니까? 우선순위를 낮출 수 있습니까? 많은 자원을 사용하지 않도록 세분화될 수 있습니까?

DB2 Connect 성능 문제점 해결

DB2 Connect 사용자가 IBM 메인프레임 서버로부터 대형 쿼리를 수행하는 동안 응답 시간이 길어지면, 다음 사항을 조사하여 가능한 성능 문제점의 원인을 확인해야 합니다.

1. IBM 메인프레임 서버에서 대형 데이터 블록(대개 32K 이상의 데이터)을 리턴하는 쿼리의 경우, 데이터베이스 관리 프로그램 구성 매개변수 RQRIOBLK가 32767로 설정되었는지 확인하십시오. 명령행 처리기(CLP)를 사용하여 다음처럼 실행될 수 있습니다.

```
db2 update database manager configuration using RQRIOBLK 32767
```

2. IBMROB 모드 정의에서 정의된 RU의 최대 크기가 적합한 값으로 설정되었는지 확인하십시오. 토큰링 하드웨어를 사용하는 연결에 대해 크기가 4K 미만으로 설정하지 않도록 권장합니다. 이더넷 하드웨어를 사용하여 연결하는 경우, 이더넷 프레임의 최대 크기는 1536바이트임을 유의하십시오. 이 크기는 제한요인이 됩니다.

z/OS용 DB2 조정

z/OS에서 비활동 스레드 처리를 최적화할 수 있습니다. V5에서는 최대 25,000개의 클라이언트를 동시에 연결할 수 있습니다. 그러나 모든 경우에 동시에 연결할 수 있는 최대 수는 1999입니다. 각 워크스테이션 클라이언트는 비활성일 때에도 연결된 상태로 남아 있을 수 있으며 스레드는 각 커미트의 비활성 체인에 배치됩니다.

DSNZPARM 매개변수 CMTSTAT, CONDBAT 및 MAXDBAT는 스레드 처리에 영향을 미칩니다. 최적의 성능을 발휘하려면, CMTSTAT를 INACTIVE로 설정하고 CONDBAT를 양호한 성능을 발휘하는 연결 DBAT의 최대수로 설정하고 MAXDBAT를 활동 중인 DBAT의 수용 가능한 최대수로 설정하십시오.

DB2 Connect 데이터 전송률 증가

쿼리 결과 세트의 행을 블록으로 만드는 것 이외에, z/OS용 DB2는 OPEN 또는 FETCH 요청에 대한 응답으로 이러한 여러 쿼리 블록을 리모트 클라이언트(예: DB2 Connect)로 리턴할 수도 있습니다. 클라이언트가 한 번에 하나의 행 데이터 블록을 요청하는 DB2 서버로 요청을 반복하여 보내는 대신, 서버가 항상 다시 보내는 것 이외에 일부 쿼리 블록을 다시 보내도록 선택적으로 요청할 수 있습니다. 이러한 추가적인 쿼리 블록을 추가 쿼리 블록이라고 합니다.

이러한 새 기능을 사용하여 클라이언트는 네트워크 성능에서 소요되는 주된 자원이 되는 네트워크 회선 턴어라운드 수를 최소화할 수 있습니다. 쿼리 블록에 대해 클라이언트에 의해 서버로 전송되는 요청 수의 감소는 바로 상당한 성능 향상 효과를 가져옵니다. 이러한 성능 향상은 보내기와 받기 사이의 전환이 성능 면에서 매우 자원이 많이 소모되는 조작이라는 사실에서 기인합니다. DB2 Connect는 이제 디폴트로 DB2 서버로부터 추가 쿼리 블록을 요청함으로써 이 성능 향상을 활용할 수 있습니다.

선호되는 네트워크 프로토콜 TCP/IP에 대해 추가 쿼리 블록(각각 최대 32K 바이트까지 가능)의 리턴을 최대한 활용할 수 있도록 DB2 Connect에서 RFC-1323 하에 설계된 창 스케일 확장이 사용 가능하게 되었습니다. 이 기능을 사용하여 TCP/IP는 보내기 및 받기 창 크기를 동적으로 조정하여 추가 쿼리 블록으로 리턴된 대량의 데이터를 효과적으로 수용할 수 있습니다.

추가 쿼리 블록

z/OS용 DB2 버전 7 이상이 설치된 서버의 추가 쿼리 블록 지원은 DB2 DDF 설치 패널에서 EXTRA BLOCKS SRV 매개변수를 통해 구성됩니다. 이 지원은 DB2가 요청에 대해 클라이언트로 다시 보낼 수 있는 최대 추가 쿼리 블록 수를 제어함으로써 구성됩니다. 이 매개변수 값을 0 - 100의 값으로 설정할 수 있습니다. 매개변수 값을 0으로 설정하면 추가 쿼리 블록 리턴을 사용하지 않는 것입니다. 이 기능을 최대한 활용하려면 디폴트값 100을 사용하여 이 설정의 효과를 감소시키는 네트워크 상의 특이성을 방지합니다.

응용프로그램이 공동 배치된 DB2 Connect 설치를 통해 직접 또는 개별 DB2 Connect 서버 설치를 통해 DB2에 액세스하는 클라이언트 측에서는 여러 가지 방법을 사용하여 커서 또는 명령문 단위로 해당 DB2 Connect 지원을 활성화할 수 있습니다.

- 커서에 대한 쿼리 행 세트 크기 사용
- 커서와 연관된 select문에 'OPTIMIZE for N ROWS'절 사용
- 커서와 연관된 select문에 'FETCH FIRST N ROWS ONLY'절 사용

DB2 Connect는 다른 SQL API를 사용하여 추가 쿼리 블록 지원을 사용할 수 있습니다.

Embedded SQL

- 사용자는 select문 자체에서 'OPTIMIZE for N ROWS'절 또는 'FETCH FIRST N ROWS ONLY'절을 지정하거나 둘다 지정하여 쿼리에 대한 추가 쿼리 블록 지원을 호출할 수 있습니다.
- 'OPTIMIZE for N ROWS'절을 사용하는 경우, DB2는 EXTRA BLOCKS SRV DDF 설치 매개변수 설정에 따라 DB2 Connect로 리턴하려는 행 수를 블록화합니다. DB2는 N으로 설정된 쿼리 결과 세트에 대해 최종적으로 리턴될 수 있는 총 행 수를 제한하지 않으므로 응용프로그램이 N개 행 이후의 행을 폐치하도록 선택할 수 있습니다.
- 'FETCH FIRST N ROWS ONLY'절도 DB2에서 쿼리 결과 세트가 N개 행으로 제한된다는 점을 제외하고는 유사하게 작동합니다. N행 이후를 폐치하면 SQL 코드 +100(데이터 끝)이 발생합니다.

CLI/ODBC

- SQL_MAX_ROWS 명령문 속성을 통해 쿼리에 대한 추가 쿼리 블록 지원을 호출할 수 있습니다.
- 'FETCH FIRST N ROWS ONLY'절은 DB2 7.1 이상의 서버에서 대신 사용됩니다.
 - 버전 7의 경우, 쿼리 결과 세트는 DB2에서 N개 행으로 제한됩니다. N행 이후를 폐치하려 하면 SQL_NO_DATA_FOUND가 발생합니다.

- 버전 8 이상에서는 CLI가 처음 N개 행만 클라이언트 커서 관리 프로그램을 통해 응용프로그램으로 리턴되도록 합니다.

JDBC setMaxRows 메소드를 통해 쿼리에 대한 추가 쿼리 블록 지원을 호출할 수 있습니다. CLI/ODBC 지원과 마찬가지로, DB2 Connect는 DB2 6.x 서버에서 'OPTIMIZE for N ROWS'절에 태그를 붙입니다. 또한, DB2 Connect는 DB2 7.1 이상의 서버에서 'FETCH FIRST N ROWS ONLY'절에 태그를 붙입니다.

RFC-1323 창 스케일

창 크기 조정은 TCP/IP용 RFC-1323 확장을 지원하는 모든 Windows, Linux 및 UNIX 플랫폼에서 지원됩니다. DB2 레지스트리 변수 DB2SORCVBUF를 사용하여 Windows, Linux 또는 UNIX용 DB2에서 이 기능을 사용할 수 있습니다. 창 스케일을 설정하려면 이 레지스트리 변수를 64K 이상의 값으로 설정해야 합니다. 예를 들어, Windows, Linux, 또는 UNIX용 DB2에서 db2set DB2SORCVBUF=65537을 발행할 수 있습니다.

최대 보내기 및 받기 버퍼 크기는 운영 체제에 따라 다릅니다. 구성된 버퍼 크기가 받아들여 졌는지 확인하려면 데이터베이스 관리 프로그램 구성 매개변수 DIAGLEVEL을 4(정보용)로 설정하고 관리 통지 로그 파일에서 메시지를 확인하십시오.

창 스케일이 적용되도록 하려면 연결 양쪽 끝, 즉 워크스테이션과 호스트에서 운영 체제 TCP/IP 스택을 통해 직접 또는 DB2 제품을 통해 간접적으로 사용 가능하게 해야 합니다. 예를 들어, DB2의 경우, 창 스케일은 현재 TCPCVBUFRSIZE를 64K 이상의 값으로 설정함으로써 운영 체제를 통해서만 활성화할 수 있습니다. 리모트 IBM Data Server Client를 사용하여 DB2 Connect 서버 워크스테이션을 통해 IBM 메인프레임 DB2 데이터베이스에 액세스하는 경우, 클라이언트에서도 창 스케일을 사용할 수 있습니다. IBM 메인프레임 DB2 데이터베이스가 관련되어 있지 않은 경우, 동일한 토큰으로 리모트 IBM Data Server Client와 워크스테이션 DB2 서버 간에 창 스케일을 사용할 수도 있습니다.

창 스케일이 네트워크 성능을 향상시키기 위해 설계된 것이지만, 네트워크 성능이 예상대로 가시적으로 향상되지 않는 경우도 있습니다. 창 스케일이 사용 가능하게 된 이후, 이더넷 또는 토큰링 LAN 어댑터에 사용되는 프레임 크기, IP MTU 크기 및 통신 링크 전체의 라우터에서 사용되는 기타 설정과 같은 요소 간의 상호작용으로 인해 성능이 저하될 수 있습니다. 그러므로 디폴트로 창 스케일은 보내기 및 받기 버퍼를 64K로 지정해 놓고 사용 안함으로 설정해야 합니다.

창 스케일을 사용할 때의 효과를 측정하여 그에 따라 네트워크를 조정해야 합니다. 네트워크 성능 향상을 위해 네트워크를 조정하는 것에 대한 내용은 <http://www.networking.ibm.com/>을 참조하십시오.

호스트 데이터 변환

서로 다른 환경(예: Intel [Windows], IEEE [Linux 및 UNIX 운영 체제], System z [VM, VSE, z/OS], IBM Power Systems [IBM i]) 간에 정보가 전송될 때, 숫자 데이터 유형(예: 10진수, 정수, 부동 소수점)을 변환해야 할 수 있습니다. 이 변환은 성능에 영향을 줄 수 있습니다.

1바이트 문자 데이터 변환에 드는 CPU 자원 사용량은 대체적으로 숫자 데이터 변환(데이터 변환이 필요한)에 드는 자원 사용량보다 적습니다.

DATE/TIME/TIMESTAMP의 데이터 변환 자원 사용량은 1바이트 CHAR 변환 자원 사용량과 거의 같습니다. 부동 소수점 수 데이터 변환에 필요한 자원 사용량이 가장 많습니다. 응용프로그램 설계자는 DB2 Connect에 기초하여 응용프로그램을 설계할 때 이러한 사실을 이용하려고 할 수 있습니다.

데이터베이스 테이블에 '비트 데이터용'으로 정의된 컬럼이 있으면, 응용프로그램과 데이터베이스 간에 전송 중인 문자 데이터는 다른 데이터 변환이 필요치 않습니다. IBM 메인프레임 데이터베이스 서버에서 데이터를 아카이브할 때 이 컬럼을 사용할 수 있습니다.

문자 데이터의 데이터 유형

문자 데이터는 CHAR 또는 VARCHAR 데이터 유형을 가질 수 있습니다. 데이터 유형의 효율성은 필드의 일반적인 데이터 길이에 따라 다릅니다.

- 실제 데이터 크기가 크게 변하는 경우, VARCHAR가 더 효율적인 데 그 이유는 CHAR는 필드를 채우기 위해 여분의 공백 문자를 추가하기 때문입니다. 이 공백 문자는 다른 문자처럼 네트워크를 통해 전송되어야 합니다.
- 실제 데이터 크기가 크게 변하지 않는 경우, CHAR가 더 효율적인 데 그 이유는 VARCHAR 필드마다 전송되어야 하는 일정 바이트 길이의 정보가 조금씩 더 있기 때문입니다.

네트워크 하드웨어

다음 고려사항은 하드웨어와 관련됩니다.

- 네트워크 또는 전송 미디어 속도

전송 미디어 속도가 빠를수록 성능이 향상됩니다. 예를 들어, 다음은 일반적인 원시 데이터 전송률입니다.

채널에서 채널로(광섬유)

4.0 MB/s

16Mbps LAN

2.0 MB/s

채널에서 채널로(정규)

1.0 MB/s

4Mbps LAN

0.5 MB/s

고속 T1 반송자(1.544Mbps)

0.193 MB/s

고속 리모트 56Kbps 전화선

0.007 MB/s

19.6Kbps 모뎀

0.002 MB/s

9600bps 모뎀

0.001 MB/s

데이터 전송률은 IBM 메인프레임 데이터베이스 서버의 경로에서 가장 느린 전송 매체에 의해 제한됩니다.

- 네트워크 어댑터 또는 통신 제어기

네트워크 어댑터와 통신 제어기의 메모리 사용을 주의깊게 계획해야 합니다. 추가로, 제어기가 DB2 Connect가 생성한 여분의 트래픽을 조절할 수 있는지를 네트워크 전문가와 확인해야 합니다.

- 네트워크 토폴로지

데이터가 LAN에서 LAN으로, 한 네트워크에서 다른 네트워크로 이동할 경우, 이동 시간을 고려하십시오. 브릿지, 라우터, 게이트웨이는 경과 시간에 추가될 것입니다. 예를 들어, 이동한 브릿지 수를 줄이면 요청마다 필요한 홉 수가 줄어들게 됩니다.

노드 사이의 물리적 거리도 고려하십시오. 메시지가 Satellite로 전송되더라도, 전송 시간은 빛의 속도($3 * 10^{**8}$ m/s)와 송신자와 수신자간의 왕복 거리에 의해 제한됩니다.

- 네트워크 트래픽

네트워크 대역폭 전체를 사용하는 경우, 응용프로그램에 대한 응답 시간과 데이터 전송률은 낮아집니다.

네트워크의 특정 부분, 예를 들어, 아주 작은 버퍼 크기를 가진 오래된 NCP 같은 곳에 데이터를 축적할 때 과다 전송이 발생할 수 있습니다.

- 네트워크 신뢰성

네트워크의 오류율이 높으면, 네트워크의 처리량은 감소하여 데이터 재전송으로 인해 성능이 저하됩니다.

제 21 장 CLI/ODBC 응용프로그램 성능 조정

CLI/ODBC는 사용자의 데이터베이스 응용프로그램에서 호출할 수 있는 SQL API입니다. CLI 함수는 DB2 스토어드 프로시저를 호출하고 스토어드 프로시저는 시스템 카탈로그 테이블에 액세스합니다.

일부 응용프로그램은 ODBC API를 사용하여 추가 처리에 사용되는 메타데이터 정보를 수집합니다. 다음과 같은 10개의 메타데이터 API 호출을 작성할 수 있습니다.

- SQLTables
- SQLColumns
- SQLSpecialcolumns
- SQLStatistics
- SQLPrimarykeys
- SQLForeignkeys
- SQLTablePrivileges
- SQLColumnPrivileges
- SQLProcedures
- SQLProcedureColumns

위에 나열된 메타데이터 API를 사용하는 특정 CLI/ODBC 응용프로그램은 데이터베이스 내의 모든 오브젝트를 쿼리할 수 있습니다. 예를 들어, SQLTables 호출은 데이터베이스의 모든 테이블에 대한 메타데이터를 요청합니다. 대형 시스템의 경우, 이러한 요청으로 인해 많은 네트워크 트래픽이 초래되고 시간이 매우 오래 걸리며 상당 양의 서버 자원이 소모될 수 있습니다.

여러 CLI/ODBC 초기화 키워드를 사용하여 데이터베이스가 처음 연결된 후 "정보 수집" 단계 중에 초기 API 호출에 의해 리턴되는 데이터 양을 제한할 수 있습니다. 이 키워드는 아래의 방법으로 설정할 수 있습니다.

1. db2cli.ini 파일을 직접 편집
2. 클라이언트 구성 지원 프로그램을 사용하는 데이터베이스의 ODBC/CLI 설정값 변경(클라이언트 구성 지원 프로그램을 지원하는 플랫폼에서)
3. DBA 명령행 인터페이스를 사용하는 데이터베이스 CLI 구성 갱신

다음은 키워드입니다.

- DBName
- TableType
- SchemaList
- SysSchema
- GrantorList
- GranteeList

제 5 부 문제점 해결

제 22 장 DB2 Connect 문제점 해결

DB2 Connect 환경에는 많은 소프트웨어, 하드웨어, 통신 제품이 포함되어 있습니다. 문제점 해결은 결론(오류 지점)에 이르기 위해 사용 가능한 데이터를 삭제하고 세분화하는 프로세스를 통해 가장 효과적으로 수행할 수 있습니다.

관련 정보를 수집하고 적용 가능한 주제를 선택한 후 참조 절로 이동하십시오.

관련 정보 수집

문제점 해결은 문제의 범위를 좁히고 가능한 원인을 조사하는 과정을 포함합니다. 관련 정보를 수집하고, 알고 있는 정보, 수집되지 않은 데이터와 삭제 가능한 경로를 판별하는 것이 올바른 시작점입니다. 최소한 다음 문제에 대해 답변할 수 있어야 합니다.

- 초기 연결이 성공적이었습니까?
- 하드웨어가 적절하게 작동하고 있습니까?
- 통신 경로가 작동하고 있습니까?
- 이전의 디렉토리 항목을 유효하지 않게 하는 통신 네트워크의 변경사항이 있습니까?
- 데이터베이스가 시작되었습니까?
- 하나 이상의 클라이언트와 DB2 Connect 서버(게이트웨이) 사이, DB2 Connect 게이트웨이와 IBM 메인프레임 데이터베이스 서버 사이 또는 DB2 Connect Personal Edition과 IBM 메인프레임 데이터베이스 서버 사이의 통신이 끊겼습니까?
- 메시지에서 리턴된 메시지와 토큰 내용으로 무엇을 알 수 있습니까?
- db2trc, db2pd 또는 db2support와 같은 진단 도구 사용이 현재 어떤 지원을 제공합니까?
- 비슷한 태스크를 실행하는 다른 머신이 정상적으로 작동하고 있습니까?
- 리모트 태스크인 경우, 로컬에서 실행해도 성공적입니까?

초기 연결에 실패한 경우

다음 질문을 검토하고 설치 단계를 수행했는지 확인하십시오.

1. 설치 처리가 성공적으로 완료되었습니까?
 - 전제조건 소프트웨어 제품이 모두 사용 가능했습니까?
 - 메모리와 디스크 스페이스가 적당했습니까?
 - 리모트 클라이언트 지원이 설치되었습니까?
 - 다른 오류 조건 없이 통신 소프트웨어 설치가 완료되었습니까?
2. UNIX 운영 체제의 경우, 제품 인스턴스가 작성되었습니까?

- 사용자와 그룹을 인스턴스 소유자와 SYSADM 그룹으로 작성했습니까?
3. 적용 가능하다면, 라이선스에 관한 정보가 성공적으로 처리되었습니까?
 - UNIX 운영 체제의 경우, nodelock 파일을 편집하여 IBM에서 제공한 암호를 입력했습니까?
 4. IBM 메인프레임 데이터베이스 서버 및 워크스테이션 통신이 제대로 구성되어 있습니까?
 - 다음 세 가지 구성은 반드시 고려하십시오.
 - a. IBM 메인프레임 데이터베이스 서버 구성은 서버에 대한 응용프로그램 리퀘스터(AR)를 식별합니다. IBM 메인프레임 서버 데이터베이스 관리 시스템에는 위치, 네트워크 프로토콜 및 보안 측면에서 리퀘스터를 정의하는 시스템 카탈로그 항목이 있습니다.
 - b. DB2 Connect 워크스테이션 구성은 서버에 대해 클라이언트를 정의하고, 클라이언트에 대해 IBM 메인프레임 서버를 정의합니다.
 - c. 클라이언트 워크스테이션 구성에는 워크스테이션 이름과 통신 프로토콜 이름이 반드시 정의되어 있어야 합니다.
 - 초기 연결 실패에 대한 문제점 분석에는 PU(물리 장치) 이름이 완전하고 올바른지 확인하거나, TCP/IP 연결에 대해 올바른 포트 번호 및 호스트 이름이 지정되었는지 확인하는 단계가 포함됩니다.
 - IBM 메인프레임 서버 데이터베이스 관리자 및 네트워크 관리자는 모두 문제점 진단에 사용할 수 있는 유틸리티를 가지고 있습니다.
 5. IBM 메인프레임 서버 데이터베이스를 사용하기 위해 IBM 메인프레임 서버 데이터베이스 관리 시스템에 필요한 권한 레벨을 가지고 있습니까?
 - 사용자 액세스 권한, 테이블 규정자 규칙, 예상되는 결과를 고려하십시오.
 6. 명령행 처리기(CLP)를 사용하여 IBM 메인프레임 데이터베이스 서버에 대해 SQL 문을 발행하려는 시도에 실패했습니까?
 - 프로시저에 따라 CLP를 IBM 메인프레임 데이터베이스 서버에 바인드했습니까?

초기 연결 후 문제점이 발생한 경우

다음 질문들은 문제의 범위를 좁히는 데 도움이 되는 시작점입니다.

1. 특별하거나 이상한 작동 상황이 있습니까?
 - 새 응용프로그램입니까?
 - 새 프로시저가 사용 중입니까?
 - 최근 시스템에 영향을 줄 만한 변경사항이 있었습니까? 예를 들어, 응용프로그램이나 시나리오가 성공적으로 최종 실행된 후에 소프트웨어 제품 또는 응용프로그램이 변경되었습니까?
 - 응용프로그램의 경우, 어떤 API를 사용하여 프로그램을 작성하였습니까?

- 소프트웨어나 통신 API를 사용한 다른 응용프로그램이 사용자의 시스템에서 실행되었습니까?
 - FixPack이 최근에 설치되었습니까? 설치 후에 사용자가 운영 체제에서 사용되지 않은(로드되지 않은) 기능을 사용했을 때 문제가 발생한 경우, IBM의 최신 FixPack을 찾아 그 기능을 설치한 후 로드하십시오.
2. 이 오류가 이전에도 발생했습니까?
 - 이전 오류 상태에 대해 문서로 남아 있는 솔루션이 있습니까?
 - 관련된 구성원(participant)은 누구이며, 그 구성원이 가능한 조치 과정에 대한 힌트를 제공할 수 있습니까?
 3. 네트워크에 대하여 정보를 리턴하는 통신 소프트웨어 명령을 사용하여 조사했습니까?
 - TCP/IP에는 TCP/IP 명령 및 디먼을 사용하여 수신된 중요한 정보가 있을 수 있습니다.
 4. *SQLCA*(*SQL* 통신 영역)에서 리턴된 유용한 정보가 있습니까?
 - 문제점 처리 프로시저에는 *SQLCODE* 및 *SQLSTATE* 필드 내용을 검사하는 단계가 있어야 합니다.
 - 응용프로그램 프로그래머는 *SQLSTATE*를 사용하여 DB2 데이터베이스 제품군에 공통되는 오류 클래스를 테스트할 수 있습니다. 분산 관계형 데이터베이스 네트워크에서 이 필드는 공통되는 기본값을 제공합니다.
 5. *START DBM*이 서버에서 실행되었습니까? 추가로, *DB2COMM* 환경 변수가 서버를 리모트로 액세스하는 클라이언트용으로 올바르게 설정되었는지 확인하십시오.
 6. 같은 태스크를 실행하는 다른 머신에서 서버에 성공적으로 연결할 수 있습니까? 서버에 연결하려는 최대 클라이언트 수에 도달했을 수 있습니다. 다른 클라이언트가 서버로부터 연결을 해제할 경우, 이전에 연결할 수 없었던 클라이언트를 연결할 수 있습니까?
 7. 해당 머신이 올바른 주소를 가지고 있습니까? 머신이 네트워크에서 유일한지 확인하십시오.
 8. 리모트 연결인 경우, 클라이언트에 올바른 권한이 부여되었습니까? 인스턴스에 대한 연결이 성공적일 수 있지만 데이터베이스 또는 테이블 레벨에서 권한이 부여되지 않았을 수 있습니다.
 9. 리모트 데이터베이스로 연결하는 첫 번째 머신입니까? 분산 환경에서 네트워크 간의 라우터 또는 브릿지가 클라이언트와 서버 간의 통신을 차단할 수 있습니다. 예를 들어, TCP/IP를 사용할 경우 리모트 호스트에 PING할 수 있는지 확인하십시오.

진단 도구

문제가 발생하는 경우, 다음을 사용할 수 있습니다.

- 덤프 파일, 트랩 파일, 오류 로그, 통지 파일 및 정보 로그가 포함된 모든 진단 데이터는 데이터베이스 관리 프로그램 구성 매개변수인 진단 데이터 디렉토리 경로 (**diagpath**)로 지정된 경로에 있습니다.

이 구성 매개변수의 값이 널이면 진단 데이터는 다음 디렉토리나 폴더 중 하나에 작성됩니다.

- Linux 및 UNIX 환경에서는 `INSTHOME/sql/lib/db2dump`에 작성되며 여기서 `INSTHOME`은 인스턴스의 홈 디렉토리입니다.
- 지원되는 Windows 환경에서는
 - **DB2INSTPROF** 환경 변수가 설정되지 않으면 `x:\SQLLIB\%DB2INSTANCE%`가 사용되며 여기서 `x:\SQLLIB`는 **DB2PATH** 레지스트리 변수에 지정된 디렉토리 및 디렉토리 참조이고 **DB2INSTANCE**는 인스턴스 이름을 갖습니다.

주: 디렉토리 이름을 `SQLLIB`로 지정할 필요는 없습니다.

- **DB2INSTPROF** 환경 변수가 설정되면 `x:\%DB2INSTPROF%\%DB2INSTANCE%`가 사용되고 여기서 **DB2INSTPROF**는 인스턴스 프로파일 디렉토리의 이름이며 **DB2INSTANCE**는 인스턴스(디폴트로, Windows 32비트 운영 체제에서 **DB2INSTDEF**의 값)입니다.
- Windows 운영 체제에서는 이벤트 보기 프로그램을 사용하여 관리 통지 로그를 볼 수 있습니다.
- 사용할 수 있는 진단 도구에는 **db2trc**, **db2pd**, **db2support** 및 **db2diag**가 포함됩니다.
- Linux 및 UNIX 운영 체제의 경우, 활동 프로세스에 대한 프로세스 상태 정보를 표준 출력으로 리턴하는 **ps** 명령을 사용할 수 있습니다.
- UNIX 운영 체제의 경우, 심각한 오류가 발생할 때 현재 디렉토리에 작성된 코어 파일. 이 파일에는 종료된 프로세스의 메모리 이미지가 있으며 오류를 발생시킨 기능을 판별하는 데 사용될 수 있습니다.

제 23 장 DB2 Connect 내의 DB2 추적

사용자 환경에 나타나는 추적 조치 및 조작은 문제를 해결할 때 유용한 정보를 제공합니다. DB2 데이터베이스 서버 제품에서 취해진 조치를 가져오고 덤프하고 형식화할 수 있습니다. 추적 기능은 DB2 데이터베이스 서버 제품의 일부로 제공됩니다.

db2trc를 사용하여 DB2 추적 확보

db2trc 명령은 DB2와 함께 제공되는 추적 기능을 제어합니다. 추적 기능은 DB2 조작에 대한 정보를 기록하고 이 정보를 읽을 수 있는 양식으로 형식화합니다.

추적 실행 중에는 오버헤드가 추가되므로 추적 기능을 사용하면 시스템 성능에 영향을 줄 수 있음을 기억하십시오.

일반적으로, IBM Software Support 및 개발 팀에서는 문제점 해결을 위해 DB2 추적을 사용합니다. 조사 중인 문제점에 대한 정보를 얻기 위해 추적을 실행할 수 있지만 DB2 소스 코드에 대한 지식이 없으면 사용은 다소 제한됩니다.

그럼에도 불구하고 추적 파일 확보 요청에 대비하여 올바르게 추적을 작동하는 방법과 추적 파일 덤프 방법을 알아두는 것이 중요합니다.

주: db2trc를 사용하려면 SYSADM, SYSCTRL 또는 SYSMAINT 권한 중 하나가 필요합니다.

사용 가능한 옵션에 대한 일반적인 개념을 얻으려면, 매개변수 없이 db2trc 명령을 실행하십시오.

```
C:\#>db2trc
Usage: db2trc (chg|clr|dmp|flw|fmt|inf|off|on) options
```

특정 db2trc 명령 매개변수에 대한 자세한 정보를 보려면 -u 옵션을 사용하십시오. 예를 들어, 추적 작동에 대한 자세한 정보를 보려면 다음 명령을 실행하십시오.

```
db2trc on -u
```

이렇게 하면 DB2 추적을 작동할 때 지정할 수 있는 모든 추가 옵션("기능"으로 레이블됨)에 대한 정보가 제공됩니다.

추적을 작동할 때 가장 중요한 옵션은 -L입니다. 이 옵션은 추적 중인 정보를 저장하는 데 사용될 메모리 버퍼의 크기를 지정합니다. 바이트 또는 MB 단위로 버퍼 크기를 지정할 수 있습니다. (MB를 지정하려면 값 뒤에 "M" 또는 "m"을 추가하십시오.) 추적 버퍼 크기는 2MB여야 합니다. 이 요구사항을 충족시키지 못하는 크기를 지정하면, 버퍼 크기가 2의 거듭제곱에 가장 가깝게 자동으로 반내림됩니다.

버퍼가 너무 작으면, 정보가 유실될 수 있습니다. 디폴트로, 버퍼가 가득차면 가장 최근의 추적 정보만 보존됩니다. 버퍼가 너무 크면, IBM Software Support로 파일을 보내기 어려울 수 있습니다.

비교적 짧은 조각(예: 데이터베이스 연결)을 추적하는 경우, 크기가 약 8MB면 일반적으로 충분합니다.

```
C:\#> db2trc on -l 8M
Trace is turned on
```

그러나 대형 조각을 추적하거나 많은 작업이 동시에 진행되는 경우에는 대형 추적 버퍼가 필요할 수 있습니다.

대부분의 플랫폼에서 추적은 언제든지 설정할 수 있으며 앞서 설명한 대로 작동합니다. 그러나 알아 두어야 할 다음과 같은 특정 상황이 있습니다.

1. 다중 데이터베이스 파티션 시스템에서는 논리적 데이터베이스 파티션과 반대로 각 실제 데이터베이스 파티션에 대해 추적을 실행해야 합니다.
2. HP-UX, Linux 및 Solaris 플랫폼에서는 인스턴스가 시작된 후에 추적이 작동 해제된 경우 지정된 크기에 관계 없이 다음 번에 추적이 시작될 때 아주 작은 버퍼가 사용됩니다. 예를 들어, 어제 db2trc on -l 8m을 사용하여 추적을 작동하고 추적을 수집한 후 추적을 작동 해제했습니다(db2trc off). 오늘, 인스턴스를 줄인 후 재시작하지 않고 메모리 버퍼를 32MB로 설정하여 추적을 실행하려고 합니다(db2trc on -l 32m). 이 경우 추적은 작은 버퍼만 가져옴을 알 수 있습니다. 이러한 플랫폼에서 추적을 효과적으로 실행하려면, 필요한 버퍼 크기로 인스턴스를 시작하기 전에 추적을 작동하고 나중에 필요에 따라 버퍼를 『지우십시오』.

DB2 추적 파일 덤프

ON 옵션을 사용하여 추적 기능을 사용할 수 있게 한 후에는 인스턴스가 완료한 모든 후속 작업이 추적됩니다.

추적이 실행되는 동안 clr 옵션을 사용하여 추적 버퍼를 지울 수 있습니다. 추적 버퍼의 모든 기존 정보가 제거됩니다.

```
C:\#>db2trc clr
Trace has been cleared
```

추적 조각이 완료되었으면, dmp 옵션 다음에 추적 파일 이름을 사용하여 디스크에 메모리 버퍼를 덤프하십시오. 예를 들면, 다음과 같습니다.

```
C:\#>db2trc dmp trace.dmp
Trace has been dumped to file
```

디스크에 추적 버퍼를 덤프한 후에 추적 기능이 계속 실행됩니다. 추적을 작동 해제하려면, 다음과 같이 OFF 옵션을 사용하십시오.


```
C:\#>db2trc off
Trace is turned off
```

DB2 추적 파일 포매팅

db2trc dmp 명령이 작성한 덤프 파일은 2진 형식으로 되어 있으며 읽을 수 없습니다. 추적 파일을 읽을 수 있는지 검증하려면 2진 추적 파일을 형식화하여 순서 제어를 표시하고 형식화된 출력을 널(NULL) 디바이스로 보내십시오.

다음 예는 이 작업을 수행하기 위한 명령을 표시합니다.

```
db2trc flw example.trc nul
```

여기서 example.trc는 dmp 옵션을 사용하여 생성된 2진 파일입니다.

이 명령의 출력은 파일을 읽는 데 문제점이 있는지 여부와 추적이 래핑되었는지 여부를 명시적으로 알려줍니다.

이 시점에서 덤프 파일을 IBM Software Support로 보낼 수 있습니다. 그러면 IBM Software Support에서 DB2 서비스를 기반으로 파일을 형식화합니다. 그러나 덤프 파일을 보내기 전에 파일을 ASCII로 형식화할 것인지 묻는 경우가 있습니다. 이는 flw 및 fmt 옵션을 통해 수행됩니다. 작성할 ASCII 파일의 이름과 함께 2진 덤프 파일의 이름을 제공해야 합니다.

```
C:\#>db2trc flw trace.dmp trace.flw
C:\#Temp>db2trc flw trace.dmp trace.flw
Total number of trace records      : 18854
Trace truncated                    : NO
Trace wrapped                      : NO
Number of trace records formatted  : 1513 (pid: 2196 tid 2148 node: -1)
Number of trace records formatted  : 100 (pid: 1568 tid 1304 node: 0)
...
```

```
C:\#>db2trc fmt trace.dmp trace.fmt
C:\#Temp>db2trc fmt trace.dmp trace.fmt
Trace truncated                    : NO
Trace wrapped                      : NO
Total number of trace records      : 18854
Number of trace records formatted  : 18854
```

출력이 "Trace wrapped"가 "YES"임을 표시하는 경우, 이는 추적 기간 동안 수집된 모든 정보를 포함할 수 있을 만큼 추적 버퍼가 크지 않았음을 의미합니다. 래핑된 추적은 상황에 따라 적절할 수 있습니다. 최신 정보(-i 옵션을 지정하지 않으면 이는 유지보수되는 디폴트 정보임)에 관심이 있으면, 추적 파일의 내용이 충분할 수 있습니다. 그러나 추적 기간 처음에 발생한 것에 관심이 있거나 발생한 모든 것에 관심이 있으면, 대형 추적 버퍼를 사용하여 조작성 재실행할 수 있습니다.

2진 파일을 읽을 수 있는 텍스트 파일로 포매팅하는 경우 사용 가능한 옵션이 있습니다. 예를 들어, db2trc fmt -xml trace.dmp trace.fmt를 사용하여 2진 데이터를

변환하고 결과를 XML 구문 분석 가능한 형식으로 출력할 수 있습니다. 추가 옵션은 추적 명령(db2trc)에 대한 자세한 설명에 표시되어 있습니다.

알아야 할 다른 사항은 Linux 및 UNIX 운영 체제에서는 DB2가 심각한 오류로 인해 인스턴스를 종료하는 경우 추적 버퍼를 디스크에 자동으로 덤프한다는 것입니다. 따라서 인스턴스가 비정상적으로 종료될 때 추적을 사용할 수 있으면 진단 디렉토리에 파일이 작성되며 파일 이름은 db2trdmp.###입니다(여기서 ###는 데이터베이스 파티션 번호임). Windows 플랫폼에서는 이러한 상황이 발생하지 않습니다. 해당 상황에서는 추적을 수동으로 덤프해야 합니다.

요약하면, 다음은 db2trc 명령의 일반 시퀀스 예입니다.

```
db2trc on -l 8M
db2trc clr
<Execute problem recreation commands>
db2trc dump db2trc.dmp
db2trc off
db2trc flw db2trc.dmp <filename>.flw
db2trc fmt db2trc.dmp <filename>.fmt
db2trc fmt -c db2trc.dmp <filename>.fmtc
```

제 24 장 DRDA 추적 파일

DRDA 추적을 분석하기 전에 DRDA가 통신 구조 및 데이터 정의에 대한 개방형 표준 준임을 이해해야 합니다. 예를 들어, DRDA는 전송을 위한 데이터 구성 방법과 해당 정보 통신 발생 방법에 대한 규칙 세트를 포함합니다.

이러한 규칙이 정의된 참조 매뉴얼은 다음과 같습니다.

- DRDA V3 Vol. 1: Distributed Relational Database Architecture™
- DRDA V3 Vol. 2: Formatted Data Object Content Architecture
- DRDA V3 Vol. 3: Distributed Data Management Architecture

이러한 매뉴얼의 PDF 버전은 www.opengroup.org에서 제공됩니다.

db2drdat 유틸리티는 DRDA 응용프로그램 리퀘스터(AR)와 DB2 DRDA 응용프로그램 서버(AS)간(예: DB2 Connect와 호스트 또는 Power Systems 서버 데이터베이스 서버간)에 교환되는 데이터를 기록합니다.

추적 유틸리티

db2drdat 유틸리티는 DB2 Connect 서버(IBM Data Server Client 대신)와 IBM 메인프레임 데이터베이스 서버 간에 교환된 데이터를 기록합니다.

이러한 지식은 특정 문제점의 원인을 판별하는 데 도움이 되므로 데이터베이스 관리자(또는 응용프로그램 개발자)가 데이터 플로우의 작동 방법을 이해하는 것이 유용합니다. IBM 메인프레임 데이터베이스 서버에 대해 CONNECT TO 데이터베이스 명령문을 실행했으나 명령에 실패하고 실패 리턴 코드를 수신한 상황을 인식했다고 가정하십시오. IBM 메인프레임 데이터베이스 서버 관리 시스템에 전달된 정보를 정확히 이해하는 경우, 리턴 코드 정보가 일반적이더라도 실패의 원인을 판별할 수 있습니다. 사용자의 오류로 인한 실패도 많이 발생합니다.

db2drdat의 출력은 DB2 Connect 워크스테이션과 IBM 메인프레임 데이터베이스 서버 관리 시스템 간에 교환된 데이터 스트림을 나열합니다. IBM 메인프레임 데이터베이스 서버로 송신된 데이터는 SEND BUFFER라는 레이블이 붙고 IBM 메인프레임 데이터베이스 서버에서 수신된 데이터는 RECEIVE BUFFER라는 레이블이 붙습니다.

수신 버퍼에 SQLCA 정보가 들어 있으면, 해당 데이터의 포맷된 해석과 레이블된 SQLCA가 다음에 나타납니다. SQLCA의 SQLCODE 필드는 IBM 메인프레임 데이터베이스 서버에 의해 리턴된 맵핑되지 않은 값입니다. 송신 및 수신 버퍼는 파일 내부에 있는 가장 오래된 버퍼부터 가장 최근 버퍼 순으로 배치됩니다. 각 버퍼에는 다음 정보가 있습니다.

- 프로세스 ID
- SEND BUFFER, RECEIVE BUFFER 또는 SQLCA 레이블. 버퍼의 첫 번째 DDM 명령이나 오브젝트는 DSS TYPE로 레이블됩니다.

송신 및 수신 버퍼에 남아 있는 데이터는 다음 내용이 들어 있는 5개의 컬럼으로 나누어집니다.

- 바이트 수
- 컬럼 2와 컬럼 3은 두 시스템 간에 교환된 DRDA 데이터 스트림을 ASCII 또는 EBCDIC로 나타냅니다.
- 컬럼 2와 컬럼 3의 ASCII 표현
- 컬럼 2와 컬럼 3의 EBCDIC 표현

추적 출력

db2drdat 유틸리티는 다음의 정보를 *tracefile*에 기록합니다.

- -r
 - DRDA 응답/오브젝트의 유형
 - 수신 버퍼
- -s
 - DRDA 요청의 유형
 - 송신 버퍼
- -c
 - SQLCA
- TCP/IP 오류 정보
 - 수신 함수 리턴 코드
 - 심각도
 - 사용된 프로토콜
 - 사용된 API
 - 함수
 - 오류 번호

주:

1. Exit 코드 0 값은 성공적으로 완료된 명령을 나타내고, 0이 아닌 값은 반대의 경우를 나타냅니다.
2. 리턴되는 필드는 사용되는 API에 따라 다릅니다.
3. 리턴되는 필드는 같은 API용이라도 DB2 Connect가 실행 중인 플랫폼에 따라 다릅니다.

4. db2drdat 명령이 이미 존재하는 파일로 출력을 보내면, 파일 권한이 삭제를 허용하는 경우 오래된 파일은 삭제됩니다.

추적 출력 파일 분석

다음 정보는 db2drdat 추적에서 캡처됩니다.

- 클라이언트 응용프로그램의 프로세스 ID(PID)
- 데이터베이스 연결 서비스(DCS) 디렉토리에서 카탈로그된 RDB_NAME
- DB2 Connect CCSID
- IBM 메인프레임 데이터베이스 서버 CCSID
- DB2 Connect 시스템이 통신하는 IBM 메인프레임 데이터베이스 서버 관리 시스템

첫 번째 버퍼에는 IBM 메인프레임 데이터베이스 서버 관리 시스템으로 전송된 EXCSAT(Exchange Server Attributes) 및 ACCRDB(Access RDB) 명령이 포함됩니다. CONNECT TO 데이터베이스 명령의 결과로서 이 속성과 명령을 송신합니다. 다음 버퍼에는 DB2 Connect가 IBM 메인프레임 데이터베이스 서버 관리 시스템으로부터 수신한 응답이 포함됩니다. 이 응답에는 교환 서버 속성 응답 데이터(EXCSATRD)와 액세스 RDB 응답 메시지(ACCRDBRM)가 들어 있습니다.

EXCSAT

EXCSAT 명령에는 서버 이름(SRVNAM) 오브젝트에 의해 지정된 클라이언트의 워크스테이션 이름이 있으며, DDM 스펙에 의한 코드 포인트 X'116D'입니다. EXCSAT 명령은 첫 번째 버퍼에 있습니다. EXCSAT 명령에서 값 X'9481A292'(CCSID 500으로 코드화)는 X'116D'가 제거되면 *mask*로 변환됩니다.

EXCSAT 명령에는 EXTNAM(외부 이름) 오브젝트도 포함됩니다. 이 오브젝트는 흔히 IBM 메인프레임 데이터베이스 관리 시스템에서 진단 정보에 위치합니다. 이것은 8바이트 프로세스 ID(또는 4바이트 프로세스 ID와 4바이트 스프레드 ID)와 20바이트 응용프로그램 ID로 이루어져 있습니다. 코드 포인트 X'115E'로 표시되며, 이 예에서는 값이 공백으로 채워진 db2bp이고 뒤에 000C50CC가 나옵니다. Linux 또는 UNIX IBM Data Server Client에서 이 값은 활동 프로세스에 대한 프로세스 상태 정보를 표준 출력으로 리턴하는 ps 명령과 상관시킬 수 있습니다.

ACCRDB

ACCRDB 명령에서는 RDBNAM 오브젝트에 RDB_NAME이 포함되며, 이는 코드 포인트 X'2110'입니다. ACCRDB 명령은 첫 번째 버퍼의 EXCSAT 명령 다음에 옵니다. ACCRDB 명령에서, X'2110'이 제거되면 값 X'E2E3D3C5C3F1'이 STLEC1로 변환됩니다. DCS 디렉토리의 목표 데이터베이스 이름 필드에 해당합니다.

어카운팅 문자열에는 코드 포인트 X'2104'가 들어 있습니다.

DB2 Connect 워크스테이션용으로 구성된 코드 세트는 ACCRDB 명령의 코드 포인트 X'119C'를 가진 CCSID 오브젝트 CCSIDSBC(1바이트 문자용 CCSID)를 위치지정하여 표시합니다. 이번 예에서, CCSIDSBC는 X'0333'이고 819입니다.

각 코드 포인트 X'119D' 및 X'119E'를 가진 추가 오브젝트 CCSIDDBC(2바이트 문자용 CCSID)와 CCSIDMBC(혼합 바이트 문자용 CCSID)는 ACCRDB 명령에도 표시됩니다. 이 예에서 각각 CCSIDDBC는 X'04B0'이고 1200이며 CCSIDMBC는 X'0333'이고 819입니다.

EXCSATRD 및 ACCRDBRM

또한, CCSID 값은 IBM 메인프레임 데이터베이스 서버로부터 두 번째 버퍼 내의 ACCRDBRM(Access RDB Reply Message)으로 리턴됩니다. 이 버퍼에는 ACCRDBRM 다음에 나오는 EXCSATRD가 들어 있습니다. 출력 파일 예에는 IBM 메인프레임 데이터베이스 서버 시스템에 대해 두 개의 CCSID 값이 들어 있습니다. 값은 1208(단일 바이트 및 혼합 바이트 문자에 사용) 및 1200(2바이트 문자에 사용)입니다.

DB2 Connect가 IBM 메인프레임 데이터베이스 서버로부터 들어오는 코드 페이지를 인식하지 못하면, 소스 및 목표 코드 페이지와 함께 SQLCODE -332가 사용자에게 리턴됩니다. IBM 메인프레임 데이터베이스 서버가 DB2 Connect로부터 전송된 코드 세트를 인식하지 못하는 경우, VALNSPRM(DDM 코드 포인트 X'1252'에서 매개변수 값이 지원되지 않음)이 리턴되며, 이 값은 사용자를 위해 SQLCODE -332로 변환됩니다.

ACCRDBRM은 또한 매개변수 PRDID(코드 포인트 X'112E'를 가진 특정 제품 ID)가 들어 있습니다. 값은 EBCDIC의 DSN08015로 표현되는 X'C4E2D5F0F8F0F1F5'입니다. 표준에 따르면 DSN은 z/OS용 DB2입니다. 버전 번호도 표시됩니다. ARI는 VSE 및 VM용 DB2 서버이고, SQL은 DB2 데이터베이스 또는 DB2 Connect이며, QSQ는 IBM i용 DB2입니다.

출력 파일 샘플 추적

다음 그림은 DB2 Connect 워크스테이션과 호스트 또는 System i 데이터베이스 서버 간에 교환된 일부 DRDA 데이터 스트림을 보여주는 샘플 출력입니다. 사용자의 관점에서 볼 때, CONNECT TO 데이터베이스 명령은 명령행 처리기(CLP)를 사용하여 발행되었습니다.

175 페이지의 그림 13은 TCP/IP 연결을 통해 DB2 Connect Enterprise Edition 버전 9.1 및 z/OS용 DB2 버전 8을 사용합니다.

1 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 0 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 233

2 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 19532 probe 1177
 bytes 250

SEND BUFFER(AR):

	EXCSAT RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	00C3D041000100BD 1041007F115E8482	...A....A...^..	.C}.....";db
0010	F282974040404040 4040404040404040	...@@@@@@@@@@@@	2bp
0020	4040F0F0F0C3F5F0 C3C3F0F0F0000000	@@.....	000C50CC000...
0030	0000000000000000 0000000000000000
0040	0000000000000000 000000000060F0F0<-00
0050	F0F1A2A495404040 4040404040404040@@@@@@@@@@	01sun
0060	4040404040404040 4040404040404040	@@@@@@@@@@@@@@	
0070	C4C5C3E5F8404040 F0A2A49540404040@@@...@@@	DECV8 0sun
0080	4040404040404040 4000181404140300	@@@@@@@@@.....
0090	0724070008147400 05240F0008144000	\$....t.\$...@.
00A0	08000E1147D8C4C2 F261C1C9E7F6F400	...G....a.....QDB2/AIX64.
00B0	08116D9481A29200 0C115AE2D8D3F0F9	..m.....Z.....	.._mask...]SQL09
00C0	F0F0F0	...	000

	ACCSEC RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0026D00100020020 106D000611A20003	.&.... .m.....	..}....._...s..
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@@@@@@STLECI
0020	404040404040	@@@@@	

3 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110546200 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 105

4 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110549755 probe 1178
 bytes 122

RECEIVE BUFFER(AR):

	EXCSATRD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0059D04300010053 1443000F115EE5F8	.Y.C...S.C...^..	..}.....;V8
0010	F1C14BE2E3D3C5C3 F100181404140300	..K.....	1A.STLECI.....
0020	0724070007147400 05240F0007144000	\$....t.\$...@.
0030	0700081147D8C4C2 F20014116DE2E3D3	...G.....m...QDB2..._STL
0040	C5C3F14040404040 4040404040000C11	...@@@@@@@@@...	EC1 ...
0050	5AC4E2D5F0F8F0F1 F5	Z.....]DSN08015

	ACCSECRD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0010D0030002000A 14AC000611A20003}.....s..

그림 13. 추적 출력의 예(TCP/IP 연결)

5 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110656806 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 233

6 data DB2 UDB DRDA Communication Manager sqljcsend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 110659711 probe 1177
 bytes 250

SEND BUFFER(AR):

	SECCHK RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	003CD04100010036	106E000611A20003	.<.A...6.n..... ..}>.....>...s..
0010	00162110E2E3D3C5	C3F1404040404040	..!.....@@@...
0020	404040404040000C	11A1D9858799F485	@@@@..... ..Regr4e
0030	A599000A11A09585	A6A39695 vr....newton

	ACCRDB RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	00ADD001000200A7	20010006210F2407!.\$..}....x.....
0010	00172135C7F9F1C1	F0C4F3C14BD7C1F8	..!5.....K... ..G91A0D3A.PA8
0020	F806030221064600	162110E2E3D3C5C3!F..!..... 8.....STLEC
0030	F140404040404040	404040404000C11	.@@@@@@@@@... 1 ...
0040	2EE2D8D3F0F9F0F0	F0000D002FD8E3C4/... .SQL09000....QTD
0050	E2D8D3C1E2C30016	00350006119C03335.....3 SQLASC.....
0060	0006119D04B00006	119E0333003C21043.

7 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259908001 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 176

8 data DB2 UDB DRDA Communication Manager sqljcreceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 0 nsec 259911584 probe 1178
 bytes 193

RECEIVE BUFFER(AR):

	SECCHKRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0015D0420001000F	1219000611490000	...B.....I.. ..}
0010	000511A400	u.

	ACCRDBRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	009BD00200020095	2201000611490000"....I.. ..}....n.....
0010	000D002FD8E3C4E2	D8D3F3F7F000C11	.../..... ..QTDSQL370...
0020	2EC4E2D5F0F8F0F1	F5001600350006115... .DSN08015.....
0030	9C04B80006119E04	B80006119D04B000}
0040	0C11A0D5C5E6E3D6	D540400006212524@.!.!%\$..NEWTON
0050	34001E244E000624	4C00010014244D00	4..\$.N..\$.L...\$.M.+...<.....(.
0060	06244FFFFF000A11	E8091E768301BE00	.\$O.....v..... .!.....Y...c...
0070	2221030000000005	68B3B8C7F9F1C1F0	"!.....h..... ..G91A0
0080	C4F3C1D7C1F8F840	4040400603022106@@@...!.! D3APA88
0090	46000A11E8091E76	831389	F.....v....Y....c.i

그림 14. 추적 출력의 예(TCP/IP 연결) 계속됨

9 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364420503 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 10

10 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 364440751 probe 1177
 bytes 27

SEND BUFFER(AR):

	RDBCMM RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000AD00100010004 200E}..... ..}.....

11 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475009631 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 54

12 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 2 nsec 475014579 probe 1178
 bytes 71

RECEIVE BUFFER(AR):

	ENDUOWRM RPYDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD05200010025 220C000611490004	+.R...%"....I..	..}.....
0010	00162110E2E3D3C5 C3F1404040404040	..!.....@STLEC1
0020	4040404040400005 211501	@

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	000BD00300010005 2408FF\$..	..}.....

13 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721710319 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 126

그림 15. 추적 출력의 예(TCP/IP 연결) 계속됨

14 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 721727276 probe 1177
 bytes 143

SEND BUFFER(AR):

	EXCSQLIMM RQSDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0053D0510001004D 200A00442113E2E3	.S.Q...M ..D!...	..}....(.....ST
0010	D3C5C3F140404040 4040404040404040	...@@@@@@@@@@@@	LEC1
0020	D5E4D3D3C9C44040 4040404040404040@@@@@@@@	NULLID
0030	4040E2D8D3C3F2C6 F0C1404040404040	@@.....@@@@@@	SQLC2F0A
0040	4040404041414141 41484C5600CB0005	@@@AAAAAHLV....<.....
0050	2105F1	!..	..1

	SQLSTT OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	002BD00300010025 2414000000001B64	.+....%\$......d	..}.....
0010	656C657465206672 6F6D206464637375	elete from ddcsu	%.....?_.....
0020	73312E6D79746162 6C65FF	s1.mytable.	.._`./.%..

15 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832901261 probe 100
 bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
 102

16 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 832906528 probe 1178
 bytes 119

RECEIVE BUFFER(AR):

	SQLCARD OBJDSS	(ASCII)	(EBCDIC)
	0 1 2 3 4 5 6 7 8 9 A B C D E F	0123456789ABCDEF	0123456789ABCDEF
0000	0066D00300010060 240800FFFFFFF3434	.f.....`\$.44	..}....-.....
0010	3237303444534E58 4F544C2000FFFFFFE	2704DSNXOTL+!.<.....
0020	0C00000000000000 00FFFFFFF000000
0030	0000000000572020 2057202020202020W W
0040	001053544C454331 2020202020202020	..STLEC1<.....
0050	2020000F44444353 5553312E4D595441	..DDCSUS1.MYTA(...
0060	424C450000FF	BLE...	..<....

17 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.100)
 pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833156953 probe 100
 bytes 16

Data1 (PD_TYPE_UINT,8) unsigned integer:
 10

그림 16. 추적 출력의 예(TCP/IP 연결) 계속됨

```

18 data DB2 UDB DRDA Communication Manager sqljcSend fnc (3.3.54.5.0.1177)
pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 833159843 probe 1177
bytes 27

SEND BUFFER(AR):

          RDBRLLBCK RQSDSS                (ASCII)          (EBCDIC)
          0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF 0123456789ABCDEF
0000 000AD00100010004 200F                ..... .          ..}.....          ..}.....

19 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.100)
pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943302832 probe 100
bytes 12

Data1 (PD_TYPE_UINT,4) unsigned integer:
54

20 data DB2 UDB DRDA Communication Manager sqljcReceive fnc (3.3.54.3.0.1178)
pid 807116 tid 1 cpid -1 node 0 sec 5 nsec 943306288 probe 1178
bytes 71

RECEIVE BUFFER(AR):

          ENDUOWRM RPYDSS                (ASCII)          (EBCDIC)
          0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF 0123456789ABCDEF
0000 002BD05200010025 220C000611490004  .+.R...%"....I..  ..}.....
0010 00162110E2E3D3C5 C3F1404040404040  ..!.....@@@@@  ....STLEC1
0020 4040404040400005 211502                @@@@@@!..          ....

          SQLCARD OBJDSS                (ASCII)          (EBCDIC)
          0 1 2 3 4 5 6 7 8 9 A B C D E F 0123456789ABCDEF 0123456789ABCDEF
0000 000BD00300010005 2408FF                .....$..          ..}.....

```

그림 17. 추적 출력의 예(TCP/IP 연결) 계속됨

DRDA 추적을 위한 후속 버퍼 정보

추가 정보를 얻기 위해 후속 송신 및 수신 버퍼를 분석할 수 있습니다. 다음 요청은 커미트를 포함합니다. commit 명령은 IBM 메인프레임 데이터베이스 서버 관리 시스템이 현재 작업 단위(UOW)를 커미트하도록 지시합니다. 네 번째 버퍼는 커미트 또는 롤백의 결과로 IBM 메인프레임 데이터베이스 서버 데이터베이스 관리 시스템으로부터 수신됩니다. 이 버퍼에는 종료 작업 단위(UOW) 응답메시지(ENDUOWRM)가 들어 있고, 현재 작업 단위(UOW)가 종료된 것을 나타냅니다.

이 예에서, 추적 항목 12에는 DDM 코드 포인트 X'2408'과 그 뒤에 X'FF'로 표시되는 널(Null) SQLCA가 포함됩니다. 널(NULL) SQLCA(X'2408FF')는 성공(SQLCODE 0)을 나타냅니다.

175 페이지의 그림 13에서는 추적 항목 16에 오류 SQLCA가 있는 수신 버퍼의 예를 보여줍니다.

제 6 부 메시지

제 25 장 DB2 Connect의 일반적인 문제점

이 주제에서는 DB2 Connect 사용시 발생하는 가장 일반적인 연결 문제점의 증상을 설명합니다. 각 경우마다 다음 항목이 제공됩니다.

- 해당 메시지, 관련된 메시지 번호와 리턴 코드(또는 프로토콜 별 리턴 코드)의 조합. 각 메시지와 리턴 코드 조합은 별도의 표제를 가지며 표제는 메시지 번호와 리턴 코드순으로 정렬됩니다.
- 보통 샘플 메시지 목록의 양식으로 제공되는 증상
- 오류의 가능한 원인을 가리키는 솔루션 제안. 일부 경우, 둘 이상의 솔루션이 제안될 수 있습니다.

SQL0965 또는 SQL0969

증상 메시지 SQL0965 및 SQL0969는 IBM i용 DB2, z/OS용 DB2 및 DB2 서버로부터 서로 다른 여러 리턴 코드와 함께 발행될 수 있습니다.

둘 중에서 하나의 메시지가 표시될 때, 그 메시지를 제시한 데이터베이스 서버 제품을 알아내려면 문서에서 원래 SQL 코드를 찾아보아야 합니다.

솔루션 IBM 메인프레임 데이터베이스로부터 수신되는 SQL 코드는 변환할 수 없습니다. 오류 코드를 기초로 문제를 정정한 다음 실패 명령을 다시 제출하십시오.

SQL5043N

증상 하나 이상의 통신 프로토콜 지원이 성공적으로 시작되지 않았습니다. 그러나 핵심 데이터베이스 관리 프로그램 기능은 성공적으로 시작되었습니다.

TCP/IP 프로토콜이 DB2 Connect 서버에서 시작되지 않았을 수 있습니다. 이전에 클라이언트가 성공적으로 연결되었을 수 있습니다.

diaglevel = 4일 경우, db2diag 로그 파일에 유사한 항목이 포함될 수 있습니다. 예를 들면, 다음과 같습니다.

```
2001-05-30-14.09.55.321092 Instance:svtdbm5 Node:000
PID:10296(db2tcpm) Appid:none
common_communication sqlcctcpconnmgr_child Probe:46
DIA3205E TCP/IP 서비스 파일에 구성되어 있고 TCT/IP 서버 지원에
필요한 소켓 주소 "30090"을
다른 프로세스에서 사용 중입니다.
```

솔루션 이 경고는 리모트 클라이언트의 서버 역할을 하는 DB2 Connect가 하나 이상의 클라이언트 통신 프로토콜을 처리하는 데 문제점이 있음을 알리는 증상입니다. 이러한 프로토콜은 TCP/IP 등이 될 수 있으며, 대개 이 메시지는 DB2 Connect에 정의된 통신 프로토콜 중 하나가 제대로 구성되어 있지 않음을 나타냅니다.

흔히, DB2COMM 프로파일 변수가 정의되지 않았거나 올바르게 정의되지 않았기 때문일 수 있습니다. 일반적으로, 문제점은 DB2COMM 변수와 데이터베이스 관리 프로그램 구성에 정의된 이름(예: svcename 또는 nname) 간의 불일치입니다.

가능한 시나리오 중 하나는 이전에는 연결이 성공적이었다가 구성이 아무 것도 변경되지 않은 상태에서 SQL5043 오류 메시지가 나타난 것입니다. 이것은 TCP/IP 프로토콜 사용시 리모트 시스템이 어떤 이유로 연결을 비정상적으로 종료할 때, 발생할 수 있습니다. 이 경우, 연결이 여전히 클라이언트에 존재하는 것으로 나타날 수 있으며, 아래 표시된 명령을 실행함으로써 추가 개입없이 연결을 리스토어할 수 있게 됩니다.

가장 가능성이 있는 시나리오는 DB2 Connect 서버에 연결되어 있는 클라이언트 중 하나가 여전히 TCP/IP 포트에 핸들을 가지고 있는 경우입니다. DB2 Connect 서버에 연결된 각 클라이언트 머신에서 다음 명령을 입력하십시오.

```
db2 terminate
db2stop
```

SQL30020

증상 SQL30020N 후속 명령과 SQL문의 성공적인 실행에 영향을 줄 분산 프로토콜 오류 때문에 실행하지 못했습니다.

솔루션 이 오류가 발생하면 서비스를 받아야 합니다. 서비스에 문의하기 전에 db2support 명령을 실행하십시오.

SQL30060

증상 SQL30060N "<authorization-ID>"는 조작 "<operation>"을 수행할 수 있는 권한이 없습니다.

솔루션 DB2에 연결할 때, 통신 데이터베이스(CDB) 테이블이 제대로 갱신되지 않았습니다.

SQL30061

증상 잘못된 IBM 메인프레임 데이터베이스 서버에 연결합니다. 목표 데이터베이스를 찾을 수 없습니다.

솔루션 DCS 디렉토리 항목에 잘못된 서버 데이터베이스 이름을 지정했을 수 있습니다. 이 오류가 발생하면 SQLCODE -30061이 응용프로그램에 리턴됩니다.

DB2 노드, 데이터베이스 및 DCS 디렉토리 항목을 점검하십시오. DCS 디렉토리 항목의 목표 데이터베이스 이름 필드가 플랫폼상의 데이터베이스 이름과 일치해야 합니다. 예를 들어 DB2 데이터베이스의 경우, 사용될 이름이 부트 스트랩 데이터 세트(BSDS) "LOCATION=locname" 필드에 사용된 이름과 같

아야 하는 데, 이 필드에 사용된 이름은 분산 데이터 기능(DDF)이 시작될 때 DSNL004I 메시지(LOCATION=*location*)에도 제공됩니다.

TCP/IP 노드에 대한 올바른 명령은 다음과 같습니다.

```
db2 catalog tcpip node <node_name> remote <host_name_or_address>
      server <port_no_or_service_name>
db2 catalog dcs database <local_name> as <real_db_name>
db2 catalog database <local_name> as <alias> at <node node_name>
      authentication server
```

그런 다음 데이터베이스에 연결하려면 다음 명령을 실행하십시오.

```
db2 connect to <alias> user <user_name> using <password>
```

리턴 코드 79를 수반한 SQL30081N

증상

```
SQL30081N 통신 오류가 발견되었습니다.
사용 중인 통신
프로토콜: "TCP/IP". 사용 중인 통신 API: "SOCKETS".
오류가
발견된 위치: "". 오류를 발견한
통신 함수:
"connect". 프로토콜 별 오류 함수: "79", "*", "*".
SQLSTATE=08001
```

솔루션 이 오류는 리모트 클라이언트가 DB2 Connect 서버에 연결하지 못하는 경우에 발생할 수 있습니다. DB2 Connect 서버에서 IBM 메인프레임 데이터베이스 서버에 연결할 때도 발생할 수 있습니다.

1. DB2 Connect 서버에서 DB2COMM 프로파일 변수가 올바르게 설정되었을 수 있습니다. 이를 확인하십시오. 예를 들어, AIX에서 DB2 Enterprise Server Edition을 실행할 때 db2set db2comm=tcpip 명령이 sqllib/db2profile에 있어야 합니다.
2. IBM Data Server Client 및 DB2 Connect 서버에서 TCP/IP 서비스 이름과 포트 스펙 간에 불일치가 있을 수 있습니다. 두 머신의 TCP/IP 서비스 파일에 있는 항목을 확인하십시오.
3. DB2가 DB2 Connect 서버에서 시작되었는지 점검하십시오. 다음 명령을 사용하여 데이터베이스 관리 프로그램 구성 diaglevel을 4로 설정하십시오.

```
db2 update dbm cfg using diaglevel 4
```

DB2를 중지하고 재시작한 후, db2diag 로그 파일을 검토하여 DB2 TCP/IP 통신이 시작되었는지 확인하십시오. 다음과 유사한 출력이 표시되어야 합니다.

```
2001-02-03-12.41.04.861119 Instance:svtdbm2 Node:00
PID:86496(db2sysc) Appid:none
common_communication sqlcctcp_start_listen Probe:80
DIA3000I "TCPIP" protocol support was successfully started.
```

프로토콜 별 오류 코드 10032를 수반한 SQL30081N

증상

SQL30081N 통신 오류가 발견되었습니다.
사용 중인 통신
프로토콜: "TCP/IP". 사용 중인 통신 API: "SOCKETS".
오류가
발견된 위치: "9.21.85.159". 오류를
발견한
통신 함수: "send". 프로토콜 별 오류 코드: "10032",
"*. *.",
SQLSTATE=08001

솔루션 이 오류 메시지는 TCP/IP 통신이 이미 실패한 머신에서 연결을 끊으려고 할 때 수신될 수 있습니다. TCP/IP 서브시스템의 문제를 정정하십시오.

대부분의 머신에서 TCP/IP 프로토콜을 재시작하면 문제가 정정됩니다. 전체 머신을 재시작해야 할 수도 있습니다.

CONNECT를 수행하는 동안 SQL30082 RC=24

증상 SQLCODE -30082 제공한 사용자 이름 또는 암호가 올바르지 않습니다.

솔루션 필요할 경우 CONNECT문에 올바른 암호가 제공되었는지 확인하십시오. 암호를 목표 서버 데이터베이스로 보낼 수 없습니다. IBM Data Server Client에서 목표 서버 데이터베이스로 암호를 보내야 합니다. 예를 들어, AIX와 같은 특정 플랫폼에서는 CONNECT문에 제공된 경우에만 암호를 획득할 수 있습니다.

제 7 부 부록

부록 A. DB2 기술 정보 개요

DB2 기술 정보는 다음 도구 및 메소드를 통해 사용할 수 있습니다.

- DB2 정보 센터
 - 주제 항목(태스크, 개념 및 참조 항목)
 - DB2 도구에 대한 도움말
 - 샘플 프로그램
 - 자습서
- DB2 서적
 - PDF 파일(다운로드)
 - PDF 파일(DB2 PDF DVD)
 - 인쇄된 서적
- 명령행 도움말
 - 명령 도움말
 - 메시지 도움말

주: DB2 정보 센터의 주제는 PDF 또는 하드카피 서적보다 더 자주 갱신됩니다. 최신 정보를 보려면 사용 가능한 문서 갱신사항을 설치하거나 ibm.com에서 DB2 정보 센터를 참조하십시오.

[ibm.com](http://www.ibm.com)에서 추가 DB2 기술 정보(예: 기술 노트, 백서 및 IBM Redbooks 서적)를 온라인으로 액세스할 수 있습니다. 다음은 DB2 정보 관리 라이브러리 소프트웨어 사이트의 주소입니다. <http://www.ibm.com/software/data/sw-library/>

문서 피드백

DB2 문서에 대한 피드백을 환영합니다. DB2 문서를 향상시키는 방법에 대해서 제안 사항이 있는 경우 db2docs@ca.ibm.com으로 전자 우편을 보내십시오. DB2 문서 팀에서는 고객의 모든 피드백을 읽지만 직접 응답할 수는 없습니다. 고객의 문제를 더 잘 이해할 수 있도록 가능한 한 구체적인 예를 제공하십시오. 특정 주제 또는 도움말 파일에 대한 피드백을 보내실 경우, 제목 및 URL을 알려주십시오.

DB2 고객 지원에 문의할 때는 이 전자 우편 주소를 사용하지 마십시오. 문서에서 해결할 수 없는 DB2 기술 문제점이 있는 경우, 해당 지역의 IBM 서비스 센터에 도움을 요청하십시오.

DB2 기술 라이브러리(하드카피 또는 PDF 형식)

다음 표는 IBM Publications Center(www.ibm.com/shop/publications/order)에서 사용할 수 있는 DB2 라이브러리에 대한 설명입니다. PDF 형식의 영문 DB2 버전 9.7 매뉴얼 및 번역된 버전은 www.ibm.com/support/docview.wss?rs=71&uid=swg2700947에서 다운로드할 수 있습니다.

표에 인쇄할 수 있는 책으로 설명된 경우라도, 사용 국가 또는 지역에 따라 해당 책을 사용할 수 없을 수도 있습니다.

매뉴얼이 갱신될 때마다 문서 번호가 증가합니다. 다음 사항을 참조하여 읽고 있는 매뉴얼이 최신 버전인지 확인하십시오.

주: DB2 정보 센터는 PDF 또는 하드카피 서적보다 자주 갱신됩니다.

표 20. DB2 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
관리 API 참조서	SA30-3958-00	예	2009년 8월
관리 루틴 및 뷰	SA30-3955-00	아니오	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 1</i>	SC27-2437-00	예	2009년 8월
<i>Call Level Interface Guide and Reference, Volume 2</i>	SC27-2438-00	예	2009년 8월
명령어 참조서	SA30-3959-00	예	2009년 8월
데이터 이동 유틸리티 안내서 및 참조서	SA30-3969-00	예	2009년 8월
데이터 복구 및 고가용성 안내서 및 참조서	SA30-3970-00	예	2009년 8월
데이터베이스 관리 개념 및 구성 참조서	SA30-3951-00	예	2009년 8월
데이터베이스 모니터링 안내서 및 참조서	SA30-3953-00	예	2009년 8월
데이터베이스 보안 안내서	SA30-3971-00	예	2009년 8월
<i>DB2 Text Search Guide</i>	SC27-2459-00	예	2009년 8월
<i>Developing ADO.NET and OLE DB Applications</i>	SC27-2444-00	예	2009년 8월
<i>Developing Embedded SQL Applications</i>	SC27-2445-00	예	2009년 8월
<i>Developing Java Applications</i>	SC27-2446-00	예	2009년 8월

표 20. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
<i>Developing Perl, PHP, Python, and Ruby on Rails Applications</i>	SC27-2447-00	아니오	2009년 8월
<i>Developing User-defined Routines(SQL and External)</i>	SC27-2448-00	예	2009년 8월
<i>Getting Started with Database Application Development</i>	GI11-9410-00	예	2009년 8월
<i>Linux 및 Windows에서 DB2 설치 및 관리 시작하기</i>	GA30-3960-00	예	2009년 8월
<i>자국어 안내서</i>	SA30-3972-00	예	2009년 8월
<i>DB2 Server 설치</i>	GA30-3962-00	예	2009년 8월
<i>IBM Data Server Client 설치</i>	GA30-3963-00	아니오	2009년 8월
<i>Message Reference Volume 1</i>	SC27-2450-00	아니오	2009년 8월
<i>Message Reference Volume 2</i>	SC27-2451-00	아니오	2009년 8월
<i>Net Search Extender Administration and User's Guide</i>	SC27-2469-00	아니오	2009년 8월
<i>파티셔닝 및 클러스터링 안내서</i>	SA30-3973-00	예	2009년 8월
<i>pureXML Guide</i>	SC27-2465-00	예	2009년 8월
<i>Query Patroller 관리 및 사용자 안내서</i>	SA30-3974-00	아니오	2009년 8월
<i>Spatial Extender and Geodetic Data Management Feature User's Guide and Reference</i>	SC27-2468-00	아니오	2009년 8월
<i>SQL Procedural Languages: Application Enablement and Support</i>	SC27-2470-00	예	2009년 8월
<i>SQL 참조서, 볼륨 1</i>	SA30-3956-00	예	2009년 8월
<i>SQL 참조서, 볼륨 2</i>	SA30-3957-00	예	2009년 8월
<i>문제점 해결 및 데이터베이스 성능 조정</i>	SA30-3952-00	예	2009년 8월
<i>DB2 버전 9.7로 업그레이드</i>	SA30-3961-00	예	2009년 8월
<i>Visual Explain 자습서</i>	SA30-3968-00	아니오	2009년 8월

표 20. DB2 기술 정보 (계속)

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
DB2 버전 9.7의 새로운 내용	SA30-3967-00	예	2009년 8월
<i>Workload Manager Guide and Reference</i>	SC27-2464-00	예	2009년 8월
<i>XQuery Reference</i>	SC27-2466-00	아니오	2009년 8월

표 21. DB2 Connect 특정 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
DB2 Connect Personal Edition 설치 및 구성	SA30-3965-00	예	2009년 8월
DB2 Connect Server 설치 및 구성	SA30-3966-00	예	2009년 8월
DB2 Connect 사용자 안내서	SA30-3964-00	예	2009년 8월

표 22. Information Integration 기술 정보

이름	문서 번호	인쇄 가능	마지막 갱신 날짜
Information Integration: Administration Guide for Federated Systems	SC19-1020-02	예	2009년 8월
Information Integration: ASNCLP Program Reference for Replication and Event Publishing	SC19-1018-04	예	2009년 8월
Information Integration: Configuration Guide for Federated Data Sources	SC19-1034-02	아니오	2009년 8월
Information Integration: SQL Replication Guide and Reference	SC19-1030-02	예	2009년 8월
Information Integration: Introduction to Replication and Event Publishing	GC19-1028-02	예	2009년 8월

인쇄된 DB2 서적 주문

인쇄된 DB2 서적이 필요한 경우, 대부분 온라인으로 구매할 수 있으나 모든 국가 또는 지역에서 가능한 것은 아닙니다. 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 일부 소프트웨어 서적은 인쇄할 수 없다는 점에 유의하십시오. 예를 들어, DB2 메시지 참조서의 볼륨은 인쇄된 서적으로 사용할 수 없습니다.

DB2 PDF 문서 DVD에서 사용할 수 있는 다수의 DB2 서적의 인쇄된 버전은 IBM에서 유료로 주문할 수 있습니다. 주문하는 위치에 따라 IBM Publications Center에서 온라인으로 서적을 주문할 수도 있습니다. 해당 국가 또는 지역에서 온라인 주문이 불가능하면, 언제든지 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문할 수 있습니다. DB2 PDF 문서 DVD의 모든 서적을 인쇄할 수는 없다는 점에 유의하십시오.

주: 가장 최신의 완전한 DB2 문서는 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7>의 DB2 정보 센터에서 유지보수됩니다.

인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.

- 해당 국가 또는 지역에서 인쇄된 DB2 서적을 온라인으로 주문할 수 있는지 여부를 확인하려면 <http://www.ibm.com/shop/publications/order>의 IBM Publications Center를 확인하십시오. 서적 주문 정보를 액세스하려면 국가/지역/언어를 선택한 다음 해당 위치에서 주문 지시사항을 따르십시오.
- 해당 지역의 IBM 담당자로부터 인쇄된 DB2 서적을 주문하려면 다음을 수행하십시오.
 1. 다음 웹 사이트 중 하나에서 해당 지역 담당자에 대한 문의처 정보를 찾으십시오.
 - www.ibm.com/planetwide에 있는 IBM 전세계 문의처 디렉토리
 - <http://www.ibm.com/shop/publications/order>의 IBM Publications 웹 사이트. 사용 지역의 해당 서적 홈 페이지에 액세스하려면 해당 국가, 지역 또는 언어를 선택해야 합니다. 이 페이지에서 "이 제품의 정보" 링크를 수행하십시오.
 2. 전화로 주문할 경우, 주문할 DB2 서적을 지정하십시오.
 3. 담당자에게 주문하려는 서적의 제목 및 문서 번호를 제공하십시오. 서적의 제목 및 문서 번호는 190 페이지의 『DB2 기술 라이브러리(하드카피 또는 PDF 형식)』를 참조하십시오.

명령행 처리기에서 SQL 상태 도움말 표시

DB2 제품은 SQL문의 결과로 나타나는 상태에 대한 SQLSTATE 값을 리턴합니다. SQLSTATE 도움말은 SQL 상태 및 SQL 상태 클래스 코드의 의미를 설명합니다.

SQL 상태 도움말을 시작하려면 명령행 처리기를 열고 다음을 입력하십시오.

```
? sqlstate or ? class code
```

여기서, *sqlstate*는 유효한 5자리 숫자로 된 SQL 상태이고 *class code*는 SQL 상태의 처음 2자리 숫자를 나타냅니다.

예를 들어, ? 08003은 08003 SQL 상태에 대한 도움말을 표시하고, ? 08은 08 클래스 코드에 대한 도움말을 표시합니다.

DB2 정보 센터의 다른 버전에 액세스

DB2 버전 9.7 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r7/>입니다.

DB2 버전 9.5 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9r5/>입니다.

DB2 버전 9 주제에 대한 DB2 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v9/>입니다.

DB2 버전 8 주제에 대한 버전 8 정보 센터 URL은 <http://publib.boulder.ibm.com/infocenter/db2luw/v8/>입니다.

DB2 정보 센터에서 원하는 언어로 항목 표시

DB2 정보 센터는 브라우저 환경 설정에 지정된 언어로 주제 항목을 표시합니다. 주제가 원하는 언어로 변환되지 않은 경우, DB2 정보 센터는 해당 주제 항목을 영어로 표시합니다.

- Internet Explorer 브라우저에서 원하는 언어로 항목을 표시하려면 다음을 수행하십시오.

1. Internet Explorer에서 도구 → 인터넷 옵션 → 언어 단추를 누르십시오. 언어 환경 설정 창이 열립니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
 - 목록에 새 언어를 추가하려면 추가... 단추를 누르십시오.

주: 언어를 추가하더라도 원하는 언어로 항목을 표시하는 데 필요한 글꼴이 컴퓨터에 설치되지 않습니다.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

- Firefox 또는 Mozilla 브라우저에서 원하는 언어로 주제 항목을 표시하려면 다음을 수행하십시오.

1. 도구 → 설정 → 내용 대화 상자의 언어 섹션에서 단추를 선택하십시오. 환경 설정 창에 언어 패널이 표시됩니다.
2. 원하는 언어가 언어 목록의 첫 번째 항목으로 지정되었는지 확인하십시오.
 - 목록에 새 언어를 추가하려면 언어 선택 창에서 원하는 언어를 선택한 다음 추가... 단추를 누르십시오.

- 언어를 목록 맨위로 이동하려면, 언어를 선택한 후 해당 언어가 언어 목록의 첫 번째 항목이 될 때까지 위로 이동 단추를 누르십시오.
3. 브라우저 캐시를 지운 후 페이지를 새로 고치면 원하는 언어로 DB2 정보 센터가 표시됩니다.

일부 브라우저 및 운영 체제 조합에서는 운영 체제의 국가별 설정을 선택한 로케일 및 언어로 변경해야 합니다.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 갱신

로컬로 설치된 DB2 정보 센터는 주기적으로 갱신해야 합니다.

시작하기 전에

DB2 버전 9.7 정보 센터는 미리 설치된 상태여야 합니다. 자세한 내용은 *DB2 Server* 설치의 『DB2 설치 마법사를 사용하여 DB2 정보 센터 설치』 주제를 참조하십시오. 정보 센터 설치에 적용되는 모든 전제조건 및 제한사항은 정보 센터 갱신에도 적용됩니다.

이 태스크에 대한 정보

기존의 DB2 정보 센터는 자동 또는 수동으로 갱신할 수 있습니다.

- 자동 갱신 - 기존 정보 센터 기능 및 언어를 갱신합니다. 자동 갱신의 또 다른 이점으로는 갱신 동안 정보 센터를 사용할 수 없는 시간이 매우 짧다는 점입니다. 또한 자동 갱신은 주기적으로 실행되는 기타 일괄처리 작업의 일부로 실행되도록 설정할 수도 있습니다.
- 수동 갱신 - 갱신 프로세스 중에 기능이나 언어를 추가하려는 경우 사용하십시오. 예를 들어, 로컬 정보 센터는 기본적으로 영어와 프랑스로 설치되어 있으며, 수동 갱신을 통해 기존 정보 센터의 기능 및 언어 갱신뿐만 아니라 독일어도 설치할 수 있습니다. 단, 수동 갱신을 수행하려면 정보 센터를 중지한 다음 갱신하고 재시작해야 합니다. 정보 센터는 갱신 프로세스 동안에는 사용할 수 없습니다.

프로시저

이 주제는 자동 갱신 프로세스에 대한 설명입니다. 수동 갱신에 대한 지시사항은 『컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신』 주제를 참조하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 자동으로 갱신하려면 다음을 수행하십시오.

1. Linux 운영 체제의 경우
 - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 `/opt/ibm/db2ic/V9.7` 디렉토리에 디폴트로 설치됩니다.

- b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
- c. 다음과 같이 ic-update 스크립트를 실행하십시오.

```
ic-update
```

2. Windows 운영 체제의 경우

- a. 명령 창을 여십시오.
- b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>\IBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
- c. 설치 디렉토리에서 doc\bin 디렉토리로 이동하십시오.
- d. 다음과 같이 ic-update.bat 파일을 실행하십시오.

```
ic-update.bat
```

결과

DB2 정보 센터가 자동으로 재시작됩니다. 갱신사항이 사용 가능한 경우, 정보 센터에는 새로 갱신된 주제가 표시됩니다. 정보 센터 갱신을 사용할 수 없는 경우, 메시지가 로그에 추가됩니다. 로그 파일은 doc\weclipse\configuration 디렉토리에 있습니다. 이 로그 파일 이름은 임의로 생성된 번호입니다.

예: 1239053440785.log

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터 수동 갱신

DB2 정보 센터를 로컬로 설치한 경우, IBM으로부터 문서 갱신사항을 받아 설치할 수 있습니다.

로컬로 설치된 DB2 정보 센터를 수동으로 갱신하려면 다음을 수행하십시오.

1. 컴퓨터에서 DB2 정보 센터를 중지한 후 독립형 모드에서 다시 시작하십시오. 독립형 모드에서 정보 센터를 실행하면 사용자의 네트워크와 연결된 다른 사용자는 정보 센터에 액세스할 수 없으므로 갱신사항을 적용할 수 있습니다. DB2 정보 센터의 워크스테이션 버전은 항상 독립형 모드에서 실행됩니다.
2. 사용 가능한 갱신사항을 확인하려면 갱신 기능을 사용하십시오. 설치해야 할 갱신사항이 있는 경우, 갱신 기능을 사용하여 이를 가져온 후 설치할 수 있습니다.

주: 인터넷에 연결되지 않은 머신에 DB2 정보 센터 갱신사항을 설치해야 할 경우, 인터넷에 연결되고 DB2 정보 센터가 설치된 머신을 사용하여 갱신 사이트를 로컬 파일 시스템으로 미리하십시오. 네트워크 상에 문서 갱신사항을 설치하려는 사용자가 많을 경우에는 갱신 사이트를 로컬로 미리링하거나 갱신 사이트의 프록시를 작성하여 갱신을 수행하면 각 개인에게 필요한 시간을 줄일 수 있습니다.

갱신 패키지가 사용 가능하면 갱신 기능을 사용하여 패키지를 가져오십시오. 그러나 갱신 기능은 독립형 모드에서만 사용할 수 있습니다.

3. 독립형 정보 센터를 중지한 후 컴퓨터에서 DB2 정보 센터를 재시작하십시오.

주: Windows 2008, Windows Vista 이상의 경우 이 절 다음에 나오는 명령은 관리자로 실행해야 합니다. 전체 관리자 권한으로 명령 프롬프트 또는 그래픽 도구를 열려면 단축 아이콘을 마우스 오른쪽 단추로 누른 후 관리자로 실행을 선택하십시오.

컴퓨터 또는 인트라넷 서버에 설치된 DB2 정보 센터를 갱신하려면 다음을 수행하십시오.

1. DB2 정보 센터를 중지하십시오.

- Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 중지를 선택하십시오.
- Linux의 경우, 다음 명령을 입력하십시오.
`/etc/init.d/db2icdv97 stop`

2. 독립형 모드에서 정보 센터를 시작하십시오.

- Windows의 경우:
 - a. 명령 창을 여십시오.
 - b. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 <Program Files>WIBM\DB2 Information Center\Version 9.7 디렉토리에 디폴트로 설치됩니다. 여기서 <Program Files>는 프로그램 파일 디렉토리의 위치를 나타냅니다.
 - c. 설치 디렉토리에서 doc\bin 디렉토리로 이동하십시오.
 - d. 다음과 같이 help_start.bat 파일을 실행하십시오.
`help_start.bat`
- Linux의 경우:
 - a. 정보 센터가 설치된 경로를 찾아가십시오. DB2 정보 센터는 /opt/ibm/db2ic/V9.7 디렉토리에 디폴트로 설치됩니다.
 - b. 설치 디렉토리에서 doc/bin 디렉토리로 이동하십시오.
 - c. 다음과 같이 help_start 스크립트를 실행하십시오.
`help_start`

시스템의 기본 웹 브라우저가 열리고 독립형 정보 센터가 표시됩니다.

3. 갱신 단추(🔄)를 누르십시오. (JavaScript™가 브라우저에서 사용 가능해야 합니다.) 정보 센터의 오른쪽 패널에서 갱신사항 찾기를 누르십시오. 기존 문서의 갱신사항 목록이 표시됩니다.

4. 설치 프로세스를 시작하려면 설치할 선택란을 체크한 후 갱신사항 설치를 누르십시오.
5. 설치 프로세스가 완료되면 완료를 누르십시오.
6. 독립형 정보 센터를 중지하십시오.
 - Windows의 경우, 설치 디렉토리의 doc\bin 디렉토리로 이동한 후 다음과 같이 help_end.bat 파일을 실행하십시오.
help_end.bat

주: help_end 일괄처리 파일에는 help_start 일괄처리 파일로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start.bat 를 중지할 때 Ctrl+C 또는 다른 메소드를 사용하지 마십시오.
 - Linux의 경우, 설치 디렉토리의 doc/bin 디렉토리로 이동한 후 다음과 같이 help_end 스크립트를 실행하십시오.
help_end

주: help_end 스크립트에는 help_start 스크립트로 시작된 프로세스를 안전하게 중지하는 데 필요한 명령이 포함되어 있습니다. help_start 스크립트를 중지할 때 다른 메소드를 사용하지 마십시오.
7. DB2 정보 센터를 재시작하십시오.
 - Windows의 경우, 시작 → 제어판 → 관리 도구 → 서비스를 누르십시오. 그런 다음 **DB2 Information Center** 서비스를 마우스 오른쪽 단추로 누른 후 시작을 선택하십시오.
 - Linux의 경우, 다음 명령을 입력하십시오.
/etc/init.d/db2icdv97 start

갱신된 DB2 정보 센터에는 새로 갱신된 주제가 표시됩니다.

DB2 자습서

DB2 자습서는 DB2 제품의 여러가지 측면을 학습하는 데 유용합니다. 각 레슨은 단계별 지시사항을 제공합니다.

시작하기 전에

정보 센터(<http://publib.boulder.ibm.com/infocenter/db2help/>)에서 XHTML 버전의 자습서를 볼 수 있습니다.

일부 레슨에서는 샘플 데이터나 코드를 사용합니다. 특정 태스크에 필요한 전제조건 설명은 자습서를 참조하십시오.

DB2 자습서

자습서를 보려면 제목을 누르십시오.

『pureXML®』(*pureXML Guide*)

DB2 데이터베이스를 설정하여 XML 데이터를 저장하고 원시 XML 데이터 스토어로 기본 조작을 수행할 수 있습니다.

Visual Explain 자습서의 『Visual Explain』

더 나은 성능을 위해 Visual Explain을 사용하여 SQL문을 분석, 최적화 및 조정할 수 있습니다.

DB2 문제점 해결 정보

DB2 데이터베이스 제품 사용 시 발생하는 광범위한 문제점을 판별하고 해결하는 데 도움이 되는 정보를 사용할 수 있습니다.

DB2 문서

문제점 해결 정보는 *DB2 문제점 해결 안내서* 또는 *DB2 정보 센터*의 데이터베이스 기본 절을 참조하십시오. DB2 진단 도구 및 유틸리티를 사용하여 문제점을 찾아내고 식별하는 방법, 가장 일반적인 문제점에 대한 솔루션 및 DB2 데이터베이스 제품에서 발생할 수 있는 문제점을 해결하는 방법 등에 관한 정보가 있습니다.

DB2 기술 지원 웹 사이트

문제점이 발생한 경우 해당 원인 및 솔루션을 찾으려면 DB2 기술 지원 웹 사이트를 참조하십시오. 기술 지원 사이트에는 최신 DB2 서적, 기술 노트, APAR(Authorized Program Analysis Report 또는 버그 수정), FixPack 및 기타 자원에 대한 링크가 있습니다. 이러한 기술 자료를 검색하여 문제에 대해 사용 가능한 솔루션을 찾을 수 있습니다.

다음은 DB2 기술 지원 웹 사이트의 주소입니다. http://www.ibm.com/software/data/db2/support/db2_9/

이용약관

다음 조건에 따라 이 책을 사용할 수 있습니다.

개인적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 개인적, 비상업적 용도로 복제할 수 있습니다. IBM의 명시적인 동의 없이는 이 책 또는 그 일부를 배포 또는 전시하거나 2차적 저작물을 만들 수 없습니다.

상업적 사용: 모든 소유권 사항을 표시하는 경우에 한하여 귀하는 이 책을 귀하 기업 집단 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적 동의 없이 이 책의 2차적 저작물을 만들거나 이 책 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 이 책이나 이 책에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대한 어떠한 허가나 라이선스 또는 권한도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 이 책의 사용이 IBM의 이익을 해친다고 판단되거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 이 사이트에서 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 경우에만 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 이 책의 내용에 대해 어떠한 보증도 제공하지 않습니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증 없이 현 상태대로 제공합니다.

부록 B. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. 비IBM 제품에 대한 정보는 이 책을 처음 발행할 때의 정보에 기초하고 있으며 변경될 수 있습니다.

IBM은 다른 국가에서 이 책에 기술된 제품, 서비스 또는 기능을 제공하지 않을 수도 있습니다. 현재 사용할 수 있는 제품 및 서비스에 대한 정보는 한국 IBM 담당자에게 문의하십시오. 이 책에서 IBM 제품, 프로그램 또는 서비스를 언급했다고 해서 해당 IBM 제품, 프로그램 또는 서비스만을 사용할 수 있다는 것을 의미하지는 않습니다. IBM의 지적 재산을 침해하지 않는 한, 기능상으로 동등한 제품, 프로그램 또는 서비스를 대신 사용할 수도 있습니다. 그러나 비IBM 제품, 프로그램 또는 서비스의 운영에 대한 평가 및 검증은 사용자의 책임입니다.

IBM은 이 책에서 다루고 있는 특정 내용에 대해 특허를 보유하고 있거나 현재 특허 출원 중일 수 있습니다. 이 책을 제공한다고 해서 특허에 대한 라이선스까지 부여하는 것은 아닙니다. 라이선스에 대한 의문사항은 다음으로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠 주식회사

고객만족센터

전화번호: 080-023-8080

2바이트 문자 세트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

Intellectual Property Licensing

Legal and Intellectual Property Law

IBM Japan, Ltd.

3-2-12, Roppongi, Minato-ku, Tokyo 106-8711 Japan

다음 단락은 현지법과 상충하는 영국이나 기타 국가에서는 적용되지 않습니다. IBM은 타인의 권리 비침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 (단, 이에 한하지 않음) 묵시적이든 명시적이든 어떠한 종류의 보증없이 이 책을 『현상 그대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책 사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지 없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

IBM은 귀하의 권리를 침해하지 않는 범위 내에서 적절하다고 생각하는 방식으로 귀하가 제공한 정보를 사용하거나 배포할 수 있습니다.

(i) 독자적으로 작성된 프로그램과 다른 프로그램(본 프로그램 포함) 간의 정보 교환 및
(ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-700

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩

한국 아이.비.엠. 주식회사

고객만족센터

이러한 정보는 해당 조건(예를 들면, 사용료 지불 등) 하에서 사용될 수 있습니다.

이 정보에 기술된 라이선스가 부여된 프로그램 및 프로그램에 대해 사용 가능한 모든 라이선스가 부여된 자료는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA) 또는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 측정치는 개발 레벨 시스템에서 작성되었을 수 있으며, 따라서 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 본인의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 제품들을 테스트하지 않았으므로, 비IBM 제품과 관련된 성능의 정확성, 호환성 또는 기타 청구에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지 없이 변경될 수 있습니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이들 예제에는 개념을 가능한 완벽하게 설명하기 위하여 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

저작권 라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 이러한 샘플 프로그램의 작성 기준이 되는 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 IBM에 추가 비용을 지불하지 않고 이들 샘플 프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 샘플 프로그램은 모든 조건하에서 완전히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 진술하지 않습니다. 샘플 프로그램은 어떠한 보증없이 "있는 그대로" 제공됩니다. IBM은 샘플 프로그램의 사용으로 인해 발생하는 모든 손해에 대해 책임을 지지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. *_enter 연도_*. All rights reserved.

상표

IBM, IBM 로고 및 ibm.com[®]은 여러 국가에 등록된 IBM Corp.의 상표 또는 등록 상표입니다. 기타 제품 및 서비스 이름은 IBM 또는 기타 회사의 상표입니다. 현재 IBM 상표 목록은 웹 “저작권 및 상표 정보”(www.ibm.com/legal/copytrade.shtml)에 있습니다.

다음 용어는 기타 회사의 상표 또는 등록상표입니다.

- Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 등록상표입니다.
- Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.
- UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.
- Intel, Intel 로고, Intel Inside[®], Intel Inside 로고, Intel[®] Centrino[®], Intel Centrino 로고, Celeron[®], Intel[®] Xeon[®], Intel SpeedStep[®], Itanium[®] 및 Pentium[®]은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표 또는 등록상표입니다.
- Microsoft, Windows, Windows NT[®] 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

색인

[가]

- 가상 리모트 통신 액세스 방식(VTAM) 54
- 갱신사항
 - 데이터베이스 디렉토리 33
 - DB2 정보 센터 195, 196
- 경보 오브젝트
 - 보기 93
- 경보 요약
 - 보기 91
- 경합
 - 시스템 자원 151
- 고가용성
 - DB2 Connect 95
- 관리 유틸리티
 - DB2 Connect 6
- 교환 서버 속성 명령 173
- 구성
 - 고려사항, 암호 변경 52
 - 호스트 연결 15
- 구성 매개변수
 - agentpri 147
 - DIRCACHE 147
 - MAXDARI 147
 - max_coordagents 115, 117
 - numdb 147
 - num_initagents 115, 117
 - num_poolagents 115, 117
 - rqrioblk 147
 - TCP_KEEPALIVE 101
- 권장 조치
 - 보기, 제출 및 저장 88
- 권한
 - 바인딩 57
- 권한 부여 ID 81
- 규정 평가 로그
 - z/OS용 DB2 Health Monitor 86
- 기호 목적지 이름 40
 - 대문자와 소문자 구별 34

[나]

- 날짜
 - 시간대 지원 35

- 네트워크
 - 데이터 전송률 155
 - 성능 도구 105
 - 조정 149
- 노드
 - 디렉토리
 - 값 34
 - 갱신 33
 - 이름
 - 노드 디렉토리 값 34
 - 디렉토리 사용자 정의 워크시트 40
 - 시스템 데이터베이스 값 33

[다]

- 다중 사이트 갱신
 - 동기점 관리 프로그램 63
 - 분산 작업 단위(DUOW) 61
 - 사용 61
 - 제어 센터 62
 - 테스트 62
- 다중 사이트 갱신 구성 마법사 62
- 데이터
 - 블로킹 111
 - 소스
 - 분산 요청(DR) 12
 - 전송
 - 비용 105, 155
 - 성능 155
 - 호스트 및 워크스테이션 간 67
 - 플로우
 - DB2 Connect 10, 105
- 데이터 변환
 - 호스트 155
- 데이터 유형
 - 문자 155
 - 변환
 - 성능 효과 155
 - 부동 소수점
 - 호스트 데이터 변환 155
 - 압축 10진수 155
 - 존 10진수 155
 - CHAR 155

- 데이터 유형 (계속)
 - INTEGER
 - 호스트 데이터 변환 155
 - VARCHAR
 - 개요 155
- 데이터 이동
 - DB2 Connect 사용 67
- 데이터베이스
 - 별명
 - 디렉토리 사용자 정의 워크시트 40
 - 시스템 데이터베이스 디렉토리 33
 - 성능 도구 105
 - 요청 그룹화 111
 - 이름
 - 디렉토리 사용자 정의 워크시트 40
 - 시스템 데이터베이스 디렉토리 33
 - DCS 디렉토리 35
 - RDBNAM 오브젝트 173
 - 조정 149
 - 호스트 데이터베이스 5
- 데이터베이스 디렉토리
 - 갱신 33
 - 노드 33
 - 다중 항목 41
 - 데이터베이스 연결 서비스(DCS) 33
- 데이터베이스 시스템 모니터
 - 개요 6
 - 리모트 클라이언트 77
- 데이터베이스 연결 서비스(DCS) 디렉토리
 - 값 35
 - 항목 갱신 33
- 데이터베이스 요청
 - 성능 그룹화 111
- 도구
 - 메모리 사용 105
 - 성능 105
 - CPU 사용 105
- 도움말
 - 언어 구성 194
 - SQL문 193
- 동기점 관리 프로그램(SPM)
 - 디폴트 매개변수 64
 - 시나리오 63
- 동적 SQL
 - 성능
 - 기술 111
 - 효과 처리 5
 - CURRENTPACKAGESET CLI/ODBC 구성 매개변수 52

- 디렉토리
 - 사용자 정의 워크시트 40
 - 시스템 데이터베이스
 - 값 33
 - 갱신 33
- 디렉토리 캐시 지원 구성 매개변수
 - DB2 Connect 조정 147

[라]

- 로그
 - 규정 평가 86
- 리모트 작업 단위
 - 개요 11
 - 예 11
 - 특성 11
- 릴리스 향상된 사항
 - 이전 릴리스
 - DB2 Connect 버전 8 3

[마]

- 마법사
 - 다중 사이트 갱신 62
- 매개변수
 - 디렉토리 40
 - 문자열 41
 - MAXAGENTS 147
 - PRDID 173
 - SYSPLEX 35
- 매개변수 문자열
 - 쉽표 35
 - 이중 쉽표 35
- 맵핑
 - SQLCODE 71
 - NOMAP 매개변수 71
- 메모리 사용 도구 105
- 명령
 - ACCRDB 173
 - ACCRDBRM 173
 - ACCSEC 173
 - commit 173
 - db2drdat 171
 - db2trc 167, 169
 - EXCSAT 173
 - EXCSATRD 173
 - GET SNAPSHOT 78
 - SECCHK 173

- 명령문
 - COMMIT 111
 - DESCRIBE 111
 - EXECUTE IMMEDIATE 111
 - FOR FETCH ONLY 111
 - GRANT 54
 - PREPARE 111
 - REVOKE 54
 - ROLLBACK
 - 응용프로그램 설계 111
 - SELECT 111
- 명령행 처리기(CLP)
 - 성능 111
 - SQL문 6
- 모니터
 - 성능 77
 - 연결 77
- 모니터링
 - 연결
 - DB2 Connect 서버 77
 - Windows 성능 모니터 77
- 목표 데이터베이스
 - 이름 35, 40
- 문서
 - 개요 189
 - 이용약관 199
 - 인쇄됨 190
 - PDF 190
- 문자 데이터 유형 155
- 문자 데이터 표현 아키텍처(CDRA) 10
- 문제점 판별
 - 사용 가능 정보 199
 - 사후 연결 164
 - 연결 163
 - 자습서 199
 - 진단 도구
 - 개요 166
- 문제점 해결
 - 개요
 - DB2 Connect 163
 - 성능 151
 - 연결 163, 164
 - 온라인 정보 199
 - 자습서 199
 - 정보 수집 163
 - 추적 가능 167
 - DRDA 174, 179
 - DB2 Connect 183

[바]

- 바인드 목록 57
- 바인딩
 - 권한
 - 오프셋이 있는 대개변수 표시문자 57
 - 유틸리티와 응용프로그램 57
 - 패키지 57
 - 패키지 이름 57
- 벤치마킹
 - 성능 105
- 변환
 - 호스트 데이터 155
- 병목 현상
 - 성능 105
 - 트랜잭션 105
- 보안
 - 노드 디렉토리 값 34
 - 소개 45
 - 유형 40
 - 추가 정보 52
 - 확장 코드
 - OS/390 및 z/OS 52
 - 힌트 52
 - GRANT 명령문 54
 - Kerberos 52
 - TCP/IP 54
- 복합 SQL
 - NOT ATOMIC 111
- 부동 소수점 수
 - 데이터 유형 155
- 분산 데이터 관리(DDM)
 - db2drdat 출력 171
 - DRDA(Distributed Relational Database Architecture) 10
- 분산 요청
 - 개요 12
 - 보상 12
 - 위치 투명성 12
 - 페더레이티드 데이터베이스 12
 - DB2 Connect 지원 12
- 분산 작업 단위
 - 개요 9
 - 다중 사이트 갱신 61
 - 지원되는 서버 61
 - 2단계 커미트 61
- 분산 트랜잭션
 - XA에 대한 클라이언트 지원 137
- 블로킹 데이터 111

블록 크기 147

[사]

사용자 정의

디렉토리, 워크시트 40

새로 고침

z/OS용 DB2 Health Monitor 87

서버

응용프로그램

DB2 Connect EE 24

서적

인쇄됨

주문 192

성능

네트워크

하드웨어 155

명령행 처리기(CLP) 영향 111

시스템 자원 151

연결 집중기(connection concentrator) 123

연결 풀링 123

응용프로그램 111

DB2 Connect

개요 105

문제점 해결 151

전송률 증가 152

조정 103

ODBC 액세스 109

OS/390 152

z/OS 152

송신 버퍼

데이터 추적 171

수신 버퍼 171

술어

논리 성능 111

스토어드 프로시저

개요 23

시간대

개요 35

시나리오

TCP/IP 보안 54

시스템 데이터베이스 디렉토리

값 33

갱신 33

시스템 상태

GET SNAPSHOT 명령 78

시스템 자원

경합 151

시작

z/OS용 DB2 Health Monitor 87

[아]

아웃바운드 시퀀스 번호

DB2 LIST DCS APPLICATIONS 명령 81

암호

변경

OS/390 및 z/OS 52

압축 10진수 데이터 유형 155

앰퍼샌드(이중 &)

SQLCODE 맵핑 파일 71

양방향 CCSID 지원

BIDI 매개변수 35

연결

재설정

호스트에 직접 15

DB2 Connect Enterprise Edition 19

집중기, 연결 집중기(connection concentrator) 참조 117

풀링

개요 115

연결 집중기(connection concentrator) 117

이점 117

호스트에 직접 15

DB2 Connect Enterprise Edition 19

IBM 메인프레임에 직접 15

System z 호스트 또는 IBM i 데이터베이스에 직접 17

연결 관리 115

연결 실패

자동 클라이언트 리라우트 101

연결 집중기(connection concentrator) 115

개요 115

구성 매개변수 117

구현 117

논리 에이전트 117

디스패처 117

연결 풀링과 비교 123

예 117

오버헤드 117

작업자 에이전트 117

제한사항 117

풀링 117

DB2 Connect 124

MAX_COORDAGENTS 구성 매개변수 117

NUM_INITAGENTS 구성 매개변수 117

NUM_POOLAGENTS 구성 매개변수 117

XA 트랜잭션 지원 117

- 연결 풀링 115
 - 개요 115
 - 연결 집중기(connection concentrator)와 비교 123
- 연결성
 - 서버, DB2 Connect Enterprise Edition 19
- 예
 - 연결 집중기(connection concentrator) 117
 - XA 집중기(concentrator) 117
- 오류
 - 문제점 해결 163
- 오류 메시지
 - DB2 Connect 183
- 워크로드 밸런싱
 - 클라이언트 측 130
 - 클라이언트에 구성 131
- 워크시트
 - 디렉토리 사용자 정의 40
- 웹 서버
 - DB2 Connect 23
- 웹 응용프로그램
 - 스토어드 프로시저 23
 - DB2 Connect 20
- 유틸리티
 - 관리, DB2 Connect 6
 - 데이터베이스 시스템 모니터 6
 - 바인딩 57
 - 추적 171
 - 프로세스 상태 173
 - db2drdat 171
 - ddcspkgn 57
 - ps (프로세스 상태) 166, 173
- 응답 시간 105
- 응용프로그램
 - 바인딩 57
 - 복합 SQL 111
 - 설계 111
 - 성능 111
 - 스토어드 프로시저 111
- 웹
 - DB2 Connect 사용 20
- 응용프로그램 개발 111
 - DB2 AD 클라이언트 15
 - ODBC 15
- 응용프로그램 리퀘스터(AR)
 - 매개변수 40
 - DRDA 정의 10
- 응용프로그램 서버(AS)
 - 개요 24

- 응용프로그램 서버(AS) (계속)
 - 구성 24
 - 배치 24
 - 2티어 및 3티어 모델 24
 - DB2 Connect 지원 24
 - DRDA 정의 10
 - FAT 클라이언트 24
- 응용프로그램 이름 모니터 요소 81
- 의사 결정 지원 시스템(DSS) 171
- 이 책에 대한 정보 v
- 이용약관
 - 서적 사용 199
- 익스포트 유틸리티
 - 호스트 및 워크스테이션 간 데이터 전송 67
- 인증 40
 - 개요 50
 - 유형
 - 다폴트 50
 - CLIENT 50, 52
 - DATA_ENCRYPT 50
 - KERBEROS 50
 - SERVER 50
 - SERVER_ENCRYPT 50
 - SERVER_ENCRYPT_AES 50
 - 유효성 확인 50
 - DB2 Connect
 - 지원 54
 - REVOKE 명령문 54
- 인증 값 33
- 임포트 유틸리티
 - 호스트 및 워크스테이션 간 데이터 전송 67

[자]

- 자동 클라이언트 리라우트
 - 설명 99
 - 연결 실패 101
 - 클라이언트 측 133
 - 클라이언트에 구성 135
 - DB2 Connect에 대한 설정 99
- 자습서
 - 문제점 판별 199
 - 문제점 해결 199
 - Visual Explain 198
- 자원 액세스 제어 기능(RACF)
 - 인증 54
- 작업 단위(UOW)
 - 리모트 11

- 작업 단위(UOW) (계속)
 - 분산 61
 - 정의 9
- 전송 130
- 전송 풀 130
- 정적 SQL
 - 성능 111
 - 효과 처리 5
- 제어 센터
 - 다중 사이트 갱신 62
- 제한사항
 - 연결 집중기(connection concentrator) 117
- 조정
 - 매개변수
 - AGENTPRI 147
 - DIRCACHE 147
 - MAXAGENTS 147
 - MAXDARI 147
 - NUMDB 147
 - RQRIOLBK 147
 - 성능
 - 네트워크 149
 - 데이터베이스 149
 - DB2 Connect 103
 - OS/390 및 z/OS용 DB2 152
- 존 10진수
 - 데이터 유형 155
- 종료 작업 단위(UOW) 응답 메시지(ENDUOWRM) 173
- 주의사항 201
- 중지
 - z/OS용 DB2 Health Monitor 87
- 진단 정보
 - 개요 166

[차]

- 참조
 - 다중 데이터베이스 항목 정의 41
- 창 스케일
 - RFC-1323 확장 154
- 처리량
 - 트랜잭션 105
- 추가 쿼리 블록
 - 개요 153
 - EXTRA BLOCKS SRV 매개변수 153
- 추적
 - 출력 파일 171, 172
 - 출력 파일 샘플 174

- 추적 (계속)
 - DB2 Connect와 서버 간 데이터 171
 - DRDA
 - 해석 171
 - DRDA 추적의 버퍼 정보 179
- 추적 가능
 - 개요 167
 - DB2 추적 167, 168, 169
 - DRDA 추적 174, 179
 - 추적 유틸리티(db2drdat) 171

[카]

- 캐시된 주소 목록 126
- 코드 페이지 ID 81
- 콜 레벨 인터페이스(CLI)
 - 개요 159
 - 응용프로그램
 - CURRENTPACKAGESET 52
 - 트러스트된 연결 45
- 쿼리 블록
 - DB2 Connect 데이터 전송률 증가 152
- 클라이언트
 - 워크로드 밸런스 130
 - 자동 클라이언트 리라우트 133
 - XA 지원 사용 138
 - 클라이언트 시퀀스 번호 81
 - 클라이언트 응용프로그램
 - 통신 복구 99
 - ID 81
 - 클라이언트 제품 ID 81
 - 클라이언트 친화도
 - 구성 139
 - 클라이언트 DB 별명 81
 - 클라이언트 NNAME 81
 - 클라이언트 Sysplex 지원
 - 한계 144

[타]

- 테스트
 - 다중 사이트 갱신 62
- 토큰
 - SQLCODE 71
- 통신 복구
 - 클라이언트 응용프로그램 99
- 트랜잭션
 - 다중 사이트 갱신 9, 61

트랜잭션 (계속)

분산

지원되는 서버 61

작업 단위(UOW) 9

지원 65

처리량 105

트랜잭션 처리 모니터 27

2단계 커미트 9

DB2 Connect Enterprise Edition 27

XA 분산 응용프로그램 65

트랜잭션 처리 모니터

다중 사이트 갱신 61

사용 특성 27

예 27

트랜잭션 27

OLTP 27

Tuxedo 27

트러스트 관계

트러스트된 컨텍스트 및 트러스트된 연결 45

트러스트된 연결 45

CLI/ODBC를 통해 사용자 전환 48

CLI/ODBC를 통해 연결 46

트러스트된 컨텍스트

CLI/ODBC를 통해 연결 46

DB2 Connect 지원 45

[파]

파일에 추적 덤프

개요 168

패키지

호스트 데이터베이스 서버 57

System i 데이터베이스 서버 57

페더레이티드 데이터베이스

분산 요청(DR) 12

페이징 블록 크기 147

프로세스 상태 유틸리티 166, 173

[하]

하드웨어

네트워크 성능 155

핵심 파일

문제점 판별 166

형식화된 데이터 오브젝트 내용 아키텍처(FDOCA) 10

호스트 데이터베이스

연결성

고가용성 97

호스트 데이터베이스 (계속)

연결성 (계속)

로드 밸런스 97

DB2 Connect Personal Edition을 사용하여 액세스 15

호스트 응용프로그램 ID 81

[숫자]

2단계 커미트

사용 61

TCP/IP 연결로 사용된 재동기화 포트 34

A

ACCRDB 명령 173

ACCRDBRM 명령 173

ACCSEC 명령 173

AFFINITY_LIST db2dsdriver 설정 139

agentpri 데이터베이스 관리 프로그램 구성 매개변수 147

ALTERNATE_SERVER_LIST db2dsdriver 설정 139

AS 목표 데이터베이스 이름 35

ATOMIC 복합 SQL

DB2 Connect에서 지원되지 않음 111

B

BINDADD 특권

바인딩 권한 57

BSDS(bootstrap data set) 매개변수

Z/OS 및 OS/390 34

C

CCSID(코드화된 문자 세트 ID)

양방향 지원

설명 35

CGI(Common Gateway Interface) 프로그래밍

이점 20

한계 20

CHAR 데이터 유형

설명 155

CLIENT 인증 유형

DB2 Connect 고려사항 50

CLIENT_AFFINITY db2dsdriver 설정 139

commit 명령

추적 출력 버퍼 173

COMMIT 명령문

정적 바인드 111

CPU

사용 도구 105

CREATE IN COLLECTION NULLID 권한 57

CURRENTPACKAGESET CLI/ODBC 키워드 52

D

DATA_ENCRYPT 인증 유형 50

DB2 Connect

개선된 기능

기능 3

개요 3

구성

IBM Power Systems 64

IBM 메인프레임 64

System z 64

데이터 이동 67

보안 45

연결 집중기(connection concentrator) 124

연결성 서버 시나리오 15

제품 3

클라이언트에 대한 Sysplex 지원 129

호스트 지원 15

Enterprise Edition

연결 서버 19

웹 서버 23

웹 응용프로그램 20

트랜잭션 처리 모니터 27

API 22

JDBC 22

SQLJ 22

XA 호환 트랜잭션 관리 프로그램 64

IBM 메인프레임 지원 15

Sysplex 지원 125

System i 지원 15

DB2 서적 주문 192

DB2 정보 센터

갱신 195, 196

다른 언어로 보기 194

버전 194

언어 194

db2drdat 명령

출력 파일 171

db2dsdriver 구성 파일

워크로드 밸런싱 매개변수 131

db2trc 명령

개요 167

추적 결과 덤프 168

db2trc 명령 (계속)

추적 결과 포매팅 169

DCS 디렉토리 값 35

dcs1ari.map 파일 71

dcs1dsn.map 파일 71

dcs1qsq.map 파일 71

DCS(데이터베이스 연결 서비스) 디렉토리

데이터베이스 연결 서비스(DCS) 디렉토리 참조 35

ddcs400.lst 파일 57

ddcsmvs.lst 파일 57

ddcstrc 유틸리티 172

ddcsvm.lst 파일 57

ddcsvse.lst 파일 57

DDM(분산 데이터 관리)

db2drdat 출력 171

DRDA(Distributed Relational Database Architecture) 아키텍처

10

DESCRIBE 명령문

복합 SQL문 111

PREPARE문이 있는 성능 111

DIRCACHE 매개변수 147

DRDA(Distributed Relational Database Architecture)

개요 9, 10

데이터 액세스 9

D(연결 끊기) 매개변수 35

E

enableAcr 매개변수 135

enableDirectXA 138

enableSeamlessAcr 매개변수 135

enableWLB 매개변수 131

EXCSAT 명령 173

EXCSATRD 명령 173

EXECUTE IMMEDIATE 문

응용프로그램 설계 111

EXTNAM 오브젝트 173

F

FOR FETCH ONLY절

SELECT 명령문 111

FORCE 명령

에이전트 ID 대상 81

G

GET SNAPSHOT 명령 78

GRANT 명령문
보안 54

H

Health Monitor
z/OS용 DB2 85
Health 경보 85

I

IBM Data Server Client
워크로드 밸런싱 130
자동 클라이언트 리라우트 지원 133
IBM Data Server Driver
워크로드 밸런싱 130
자동 클라이언트 리라우트 지원 133
IBM i
DRDA 10
IBM WebSphere 21
InfoSphere Federation Server
개요 7
INTEGER 데이터 유형 155
INTERRUPT_ENABLED(연결 끊기) 매개변수 35

J

Java
응용프로그램 서버
API 22
DB2 Connect 22
JDBC 22
SQLJ 22

K

Kerberos 인증 프로토콜
DB2 Connect 50
OS/390 및 z/OS 52

L

LIST DCS APPLICATIONS 명령 81
LOCALDATE 매개변수 35

M

maxagents 데이터베이스 관리 프로그램 구성 매개변수 147
MAXDARI 구성 매개변수 147
maxRefreshInterval 매개변수 131
maxTransportIdleTime 매개변수 131
maxTransports 매개변수 131
maxTransportWaitTime 매개변수 131
max_coordagents 데이터베이스 관리 프로그램 구성 매개변수 115,
117
Microsoft Windows
응용프로그램 15

N

NOMAP 매개변수
DCS 디렉토리 매개변수 71
SQL 맵핑 해제 71
SQL 코드 맵핑 35
NON AATOMIC 복합 SQL
응용프로그램 설계 111
NONE 인증 유형 54
NULLID
OS/400 57
NUMDB
매개변수 147
num_initagents 데이터베이스 관리 프로그램 구성 매개변수 115, 117
num_poolagents 데이터베이스 관리 프로그램 구성 매개변수 115,
117

O

ODBC(Open Database Connectivity)
개요 159
액세스 최적화 109
응용프로그램
CURRENTPACKAGESET 52
인터페이스 15
OS/390
DRDA 10
OS/390 및 z/OS용 DB2 Universal Database
노드 디렉토리 값 34
보안 52
부트스트랩 데이터 세트 34
BSDS 매개변수 34
DOMAIN 34
DYNAMICRULES (BIND) 옵션 52
RESPORT 34

OS/390 및 z/OS용 DB2 Universal Database (계속)

TCPPORT 34

P

Power Systems

DRDA 10

PRDID 매개변수 173

PREPARE 명령문

성능 효과 111

응용프로그램 설계 111

PROGRAM 인증 유형 54

ps 명령

개요 166

EXTNAM 오브젝트 173

R

Relational Connect

제품 설명 7

REVOKE 명령문

인증 54

ROLLBACK 명령문

정적 바인드 111

RQRIOLBK 매개변수

조정 147

S

SAME 인증 유형 54

SECCHK 명령 173

SELECT 명령문

갱신 가능 111

응용프로그램 설계 111

FOR FETCH ONLY 111

SERVER 인증 유형 50

SERVER_ENCRYPT 인증 유형 50

SERVER_ENCRYPT_AES 인증 유형 50

SET CURRENT PACKAGESET 문 52

SHOW DETAIL 모니터 옵션 81

SOCKS

노드

필수 환경 변수 34

SQL0965 오류 코드 183

SQL0969 오류 코드 183

SQL1338 오류 코드 34, 183

SQL30020 오류 코드 183

SQL30060 오류 코드 183

SQL30061 오류 코드 183

SQL30073 오류 코드 183

SQL30081N 오류 코드 183

SQL30082 오류 코드 183

SQL5043N 오류 코드 183

SQLCA(SQL communication area)

데이터 버퍼 171

SQLCODE 필드 171

SQLCODE

맵핑 71

SQLCA의 필드 171

SQLCODE 맵핑

조정 71

NOMAP 매개변수 71

SQLCODE 맵핑 파일 71

SQLDA(SQL 디스크립터 영역)

할당 크기 111

SQLSTATE

클래스 코드 71

SQL(Structured Query Language)

동적 111

정적 111

SQL문

도움말 표시 193

DB2 Connect 5

SQL/DS

DRDA 10

SQL_ATTR_

TRUSTED_CONTEXT_PASSWORD

사용 48

TRUSTED_CONTEXT_USERID

사용 48

USE_TRUSTED_CONTEXT

사용 46

SRVNAM 오브젝트 173

START MVS 시스템 명령어 86

STOP MVS 시스템 명령어 86

Sysplex

결함 허용 126

구성 요구사항 127

로드 밸런스 126

매개변수 35

사용 126

우선순위 정보 126

클라이언트 지원 129

클라이언트 측 지원의 한계 144

DB2 Connect 지원 125

System z에 대한 고려사항 125

System i
DRDA 10

T

TCP/IP

구성

호스트 연결 17

리모트 호스트 이름 34, 40

보안

검증 52

서비스 이름 34

인증

시나리오 54

재동기화 포트 34

포트 이름 40

호스트 이름 40

ACCSEC 명령 173

DOMAIN 34

RESPORT 34

RFC-1323 확장

창 스케일 154

SECCHK 명령 173

TCPPORT 34

TCP_KEEPALIVE

운영 체제 구성 매개변수 101

Tuxedo

DB2 Connect Enterprise Edition 27

V

VARCHAR 데이터 유형

설명 155

Visual Explain

지습서 198

VM

DRDA

10

VSE

DRDA 10

W

WebSphere

개요 21

WebSphere MQ

DB2 Connect 124

Windows 운영 체제

성능 모니터

DB2 응용프로그램 모니터링 77

X

XA

자원 관리자 27

집중기(concentrator) 예 117

클라이언트 측 지원 137

트러스트된 연결 45

XA 지원

클라이언트가 사용 가능 138

XA 트랜잭션 관리 프로그램

설명 27

연결 집중기(connection concentrator) 117

X/Open 분산 트랜잭션 프로세싱(DTP) 모델 27

Z

z/OS

DRDA 10

z/OS용 DB2 Health Monitor

개요 86

경보 오브젝트 보기 93

경보 요약 보기 91

권장 조치 보기, 제출 및 저장 88

시작, 중지 및 새로 고침 87



SA30-3964-00



Spine information:

IBM DB2 Connect 9.7

DB2 Connect 사용자 안내서

