

IBM

@server

iSeries

분산 데이터베이스 프로그래밍

버전 5 릴리스 3





@server

iSeries

분산 데이터베이스 프로그래밍

버전 5 릴리스 3

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 주의사항의 정보를 읽으십시오.

제 6 판(2005년 8월)

이 개정판은 새 개정판에서 별도로 명시하지 않는 한, Operating System/400(제품 번호 5722-SS1)의 버전 5, 릴리스 3, 수정 0 및 모든 후속 릴리스와 수정에 적용됩니다. 이 버전은 일부 축약 명령어 세트 컴퓨터(RISC) 모델에서는 실행되지 않으며 CISC 모델에서도 실행되지 않습니다.

이 개정판은 SC41-5702-03을 대체합니다.

© Copyright International Business Machines Corporation 2005. All rights reserved.

목차

분산 데이터베이스 프로그래밍 정보 vii

이 설명서의 사용자 ix

V5R3 분산 데이터베이스 프로그래밍 책의 새로운 사항. xi

인쇄 가능한 PDF. xiii

코드 면책사항 정보. xv

제 1 장 분산 관계형 데이터베이스와 iSeries 서버. . . 1

분산 관계형 데이터베이스 처리 1

 리모트 작업 단위 4

 분산 작업 단위 4

 기타 분산 관계형 데이터베이스 용어 및 개념 . . . 5

분산 관계형 데이터베이스 구조(DRDA) 지원 6

DRDA 및 CDRA 지원 7

 CDRA를 사용한 문자 변환 7

어플리케이션 리퀘스터 드라이버 프로그램 8

iSeries 서버의 분산 관계형 데이터베이스 8

iSeries 분산 관계형 데이터베이스(DRDB) 관리 . . . 10

예: Spiffy사의 분산 관계형 데이터베이스 11

 Spiffy 조직 및 시스템 프로파일 11

 spiffy사의 자동차 서비스 업무 프로세스 12

 spiffy사의 분산 관계형 데이터베이스(DRDB) 관리 13

제 2 장 분산 관계형 데이터베이스의 계획 및 설계 15

분산 관계형 데이터베이스에 대한 요구와 예상 식별 15

 분산 관계형 데이터베이스에 대한 자료 요구 . . . 15

 분산 관계형 데이터베이스 기능 16

 분산 관계형 데이터베이스에 대한 목표 및 방향 16

분산 관계형 데이터베이스에 대한 어플리케이션, 네트워크 및 자료의 설계 18

 추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계. 18

 분산 관계형 데이터베이스에 대한 네트워크 고려사항. 19

 분산 관계형 데이터베이스에 대한 자료 고려사항 20

분산 관계형 데이터베이스에 대한 관리 전략 개발 20

 분산 관계형 데이터베이스에 대한 일반 조작 . . . 20

 분산 관계형 데이터베이스에 대한 보안 고려사항 22

 분산 관계형 데이터베이스 계정 23

분산 관계형 데이터베이스에 대한 문제 분석 . . . 23

분산 관계형 데이터베이스 백업 및 회복 24

제 3 장 iSeries 분산 관계형 데이터베이스의 통신 25

DRDA 구현용 통신 톨 25

 분산 관계형 데이터베이스용 APPC/APPN 25

 DDM 및 분산 관계형 데이터베이스(RDB) 사용 26

분산 관계형 데이터베이스 통신 네트워크 고려사항 26

분산 관계형 데이터베이스(RDB)용 통신 구성 27

 APPC용 통신 네트워크 구성 28

 예: 분산 관계형 데이터베이스에 대한 APPN 구성 30

 TCP/IP용 통신 네트워크 구성 38

 OptiConnect를 통한 통신 구성. 38

제 4 장 iSeries 분산 관계형 데이터베이스에 대한 보안. 41

분산 관계형 데이터베이스(RDB) 보안 요소. 42

 APPC 네트워크의 DRDA 보안 요소. 43

 APPC 네트워크의 DRDA 어플리케이션 서버 (AS) 보안. 46

 TCP/IP를 사용한 DDM/DRDA 보안 요소. 48

 DRDA 서버 액세스 제어 종료 프로그램 60

 DRDA용 오브젝트 관련 보안 63

 분산 관계형 데이터베이스 오브젝트에 관한 . . . 65

 분산 관계형 데이터베이스에 대한 허용 권한 하에서 실행되는 프로그램 66

분산 관계형 데이터베이스(DRDB)에서 보호 전략. . . 67

제 5 장 iSeries 분산 관계형 데이터베이스 설정 . . 69

iSeries 서버에 관한 작업 관리 70

 DRDA용 작업 관리 환경 설정. 70

 APPC용 서브시스템 설정에 대한 고려사항 71

사용자 관계형 데이터베이스에 관한 DRDA 고려사항 74

관계형 데이터베이스 디렉토리 사용 75

 관계형 데이터베이스 디렉토리에 대한 작업 . . . 75

 관계형 데이터베이스 디렉토리 설정 예 81

DRDA 보안 설정 84

DRDA용 TCP/IP 서버 설정 84

대화식 SQL(ISQL)용 SQL 패키지 설정. 85

DDM 파일 설정 85

분산 관계형 데이터베이스에서 자료를 표에 로딩 . . . 86

신규 자료를 분산 관계형 데이터베이스 표에 로딩	86
iSeries 서버 간의 자료 이동	88
비iSeries 서버에서 iSeries 서버로 데이터베이스 이동	94
제 6 장 분산 관계형 데이터베이스(DRDB) 관리 및 조작 TASK	97
관계형 데이터베이스 활동 모니터	97
분산 관계형 데이터베이스에서 작업(Job)에 대한 작업	98
분산 관계형 데이터베이스에서 사용자 작업(Job)에 대한 작업	99
분산 관계형 데이터베이스에서 활동 작업(Job)에 대한 작업	100
분산 관계형 데이터베이스에서 확약 정의에 대한 작업	101
분산 관계형 데이터베이스의 작업 기록부를 사용 한 요구 정보 트래킹	102
분산 관계형 데이터베이스 작업 찾기	103
리모트 iSeries 서버 조작	105
DDM 대화 제어	106
DDM 자원 재생	108
프로그램에 의해 사용된 오브젝트 표시	108
예: 프로그램 참조 표시	110
분산 관계형 데이터베이스에서 콜렉션 드롭(Drop)	111
분산 관계형 데이터베이스에서 작업 계정	112
TCP/IP 서버 관리	113
DRDA TCP/IP 서버 전문 용어	114
DDM용 TCP/IP 통신 지원 개념	114
DRDA/DDM 서버 작업	117
DDM 서버 작업 서브시스템 구성	120
서버 작업 식별	121
관계형 데이터베이스 디렉토리 감사	124
제 7 장 분산 관계형 데이터베이스에 대한 자료 가 용성 및 보호	127
분산 관계형 데이터베이스에 대한 회복 지원	127
분산 관계형 데이터베이스에 대한 디스크 장애 발생 후 자료 회복	128
분산 관계형 데이터베이스에 대한 저널 관리	130
확약 제어를 통한 트랜잭션 회복	133
분산 관계형 데이터베이스에 대한 저장 및 복원 처리	136
분산 관계형 데이터베이스에 대한 네트워크 중복 문 제	141
분산 관계형 데이터베이스 네트워크에서 자료 중복	142

제 8 장 분산 관계형 데이터베이스(DRDB) 성능	145
네트워크를 통한 분산 관계형 데이터베이스 성능 개 선	145
서버를 통한 분산 관계형 데이터베이스 성능 개선	146
데이터베이스를 통한 분산 관계형 데이터베이스 성 능 개선	147
DRDA 자료 위치 결정	147
DRDA블로킹에 영향을 주는 요소	147
DRDA 조회 블록의 크기에 영향을 미치는 요소	150
제 9 장 분산 관계형 데이터베이스(DRDB) 처리	153
iSeries 문제점 처리 개요	153
분리되는 분산 관계형 데이터베이스(DRDB) 문제	154
DRDA의 부정확한 출력 문제	155
어플리케이션이 예상 시간 내에 완료되지 않는 문제	155
분산 관계형 데이터베이스 사용자에게 대한 작업	158
화면 복사	159
메세지	160
APPC에 대한 프로그램 시작 요구 실패 처리	165
TCP/IP에 대한 연결 요구 실패 처리	166
어플리케이션 문제점	168
리스팅	168
SQLCODE 및 SQLSTATE	171
시스템 및 통신 문제점	176
iSeries 문제점 기록부	176
실패 보고를 위한 자료 수집	178
작업 기록부 인쇄	178
TCP/IP 서버 사전시작 작업의 작업 기록부 찾기	178
제품 활동 기록부 인쇄	179
작업 추적	180
통신 추적	181
TRCTCPAPP 추적	183
첫 번째 실패 자료 캡처(FFDC) 자료 찾기	186
어플리케이션 서버 문제점을 진단하기 위한 서비스 작업 시작	187
APPC 서버에 대한 서비스 작업	187
사용자 자신의 TPN 작성 및 QCNTSRVC 설정	188
TCP/IP 서버에 대한 서비스 작업	189
QRWOPTIONS 자료 영역 사용법	190
제 10 장 분산 관계형 데이터베이스(DRDB) 어플 리케이션 쓰기	193
분산 관계형 데이터베이스(DRDB) 어플리케이션에 대한 프로그래밍 고려사항	194
DRDA 프로토콜의 새로운 레벨	194
분산 관계형 데이터베이스 오브젝트 명명	195

분산 관계형 데이터베이스(DRDB)에 연결 . . .	196	WHERE절의 호스트 변수 유형이 성능에 어떤 영향을 미칩니까?	259
분산 관계형 데이터베이스 및 SQL CALL의 특 정 SQL	205	규정되지 않은 표와 뷰 이름을 분석하기 위해 라 이브러리 리스트를 사용할 수 있습니까? . . .	260
DRDA 작업 단위 종료	208	DB2 Connect 사용자가 일반적으로 사용되는 EBCDIC 순서 대신 iSeries 서버의 DRDA 작 업 NLSS 정렬 순서표가 사용되도록 지정할 수 있습니까?	261
저장 프로시저, 사용자 정의 기능(UDF) 및 확 약 제어	209	조회를 실행할 때 행이 리턴되지 않는 이유는 무 엇입니까?	261
코드화 문자 세트 ID(CCSID).	209	iSeries용 DB2 UDB V5R3과 대화하기 위해 필요한 DB2 UDB for Linux/Unix/Windows (LUW) 또는 DB2 Connect LUW의 레벨은 무 엇입니까?	261
기타 DRDA 자료 변환	212	DB2 Connect 버전 8에서 iSeries V5R3까지 사용 가능한 화면이동 커서 지원을 얻는 방법은 무엇입니까?	262
DDM 파일 및 SQL	212	상이한 환경에서의 상호 조작에 대한 추가 정보	262
분산 관계형 데이터베이스 프로그램 작성 . . .	213		
SQL문을 사용한 프로그램 사전컴파일 . . .	214	부록 C. 추적 작업 및 FFDC 자료 해석	267
어플리케이션 프로그램 컴파일	216	추적 작업의 RW 구성요소에 대한 자료 항목 해석	267
어플리케이션 바인딩	217	예: RW 추적 자료 분석	268
테스트 및 디버깅	218	RW 추적점 설명	269
SQL 패키지에 대한 작업	220	첫 번째 실패 자료 캡처(FFDC)	273
SQL 패키지 작성(CRTSQLPKG) 명령 사용	220	FFDC 덤프	274
SQL 패키지 관리	221	FFDC 덤프 출력 설명	277
DLTSQPKG(SQL 패키지 삭제) 명령	221	DDM 코드	282
SQL DROP PACKAGE 명령문	221		
		부록 D. 용어집	287
부록 A. 어플리케이션 프로그래밍 예	223	부록 E. 관련 정보	295
컬렉션 및 표 작성 예	224	iSeries 서버 정보	295
표에 자료 삽입 예	225	분산 관계형 데이터베이스 라이브러리	297
RPG 프로그램 예	230	Other IBM 분산 관계형 데이터베이스 플랫폼 라이 브러리	298
COBOL 프로그램 예	239	구조 설명서	299
C 프로그램 예	246	레드북	300
프로그램 출력 예	252		
		부록 F. 주의사항	301
부록 B. DRDA를 사용하는 공통 플랫폼 액세스	253	상표	302
CCSID 고려사항	253	코드 면책사항 정보	303
iSeries 시스템 값 QCCSID	254	서적의 다운로드 및 인쇄 조건	304
VM 데이터베이스 관리자용 DB2 UDB 서버 및 DB2 UDB for z/OS에 대한 CCSID 변환 고려 사항	255	색인	305
서로 다른 어플리케이션 서버에 대화식 SQL 및 조 회 관리자 설정	256		
비iSeries 어플리케이션 리퀘스터(AR) 사용자 FAQ	256		
DB2 Connect에서 연결하려고 할 때			
SQL5048N 메시지를 받는 이유는 무엇입니까?	257		
iSeries 파일을 저널링해야 합니까?	257		
조회 자료는 성능 향상을 위해 언제 블록됩니까?	258		
DBM SQL0969N 오류 메시지에 보고된 SQLCODE 및 연관된 토큰을 어떻게 해석합니 까?	259		

분산 데이터베이스 프로그래밍 정보

분산 데이터베이스 프로그래밍에서는 Operating System/400(OS/400) 사용권 프로그램의 분산 관계형 데이터베이스 관리 부분에 대해 설명합니다. 분산 관계형 데이터베이스 관리는 어플리케이션 외부에 있으며 일반적으로 컴퓨터의 네트워크에 위치지정된 자료에 액세스하는 어플리케이션을 제공합니다.

이 예에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 이 설명서의 사용자
- V5R3 분산 데이터베이스 프로그래밍 책의 새로운 사항

먼저 iSeries 분산 관계형 데이터베이스를 처리, 지원, 프로그래밍 및 관리하는데 관한 정보는 분산 관계형 데이터베이스 및 iSeries 서버를 참조하십시오.

이 설명서의 사용자

이 설명서는 주로 하나 이상의 iSeries 서버에 있는 분산 관계형 데이터베이스의 개발, 관리 및 지원을 담당하는 어플리케이션 프로그래머를 위한 것입니다. iSeries 데이터베이스에 익숙하지 않은 어플리케이션 프로그래머는 OS/400 오퍼레이팅 시스템이 제공하는 데이터베이스 전체 범위에 대한 개요를 파악할 수 있습니다. 어플리케이션 프로그래머는 이 정보를 사용하여 분산 관계형 데이터베이스 어플리케이션이 실행하는 서버 문맥을 알 수 있습니다.

이 설명서를 사용하려면 먼저 일반 프로그래밍 개념 및 전문용어에 익숙해야 하며 iSeries 서버 및 OS/400 오퍼레이팅 시스템에 대해 전반적으로 이해해야 합니다.

V5R3 분산 데이터베이스 프로그래밍 책의 새로운 사항

이 릴리스에는 다음과 같은 갱신사항이 포함되어 있습니다.

- 서버 인증 항목 표시 방법
- Kerberos DRDA[®] 서비스명 정의
- RDB 디렉토리 항목 예 추가
- STRTRC 사용
- TRCTCPAPP 추적 형식화
- QRWOPTIONS 자료 영역 사용법
- DRDA 프로토콜의 새로운 레벨
- CONNECT문에 대한 새 진단 정보
- 연결하려고 할 때 SQL5048N 메시지를 받는 이유는 무엇입니까?
- 바인드 옵션 예
- 조회 닫기
- 사용자 ID 및 암호 길이
- 저장 프로시저, 사용자 정의 기능(UDF) 및 확약 제어
- 일반 텍스트로 부여하는 특정 암호에 대한 주의

인쇄 가능한 PDF

이 문서의 PDF 버전을 보거나 다운로드하려면 분산 데이터베이스 프로그래밍(약 2722KB)을 선택하십시오.

PDF 파일 저장

워크스테이션에서 PDF 파일을 저장하여 보기 또는 인쇄하려면:

1. 브라우저에서 PDF 파일을 마우스 오른쪽 버튼으로 클릭하십시오(위의 링크를 마우스 오른쪽 버튼으로 클릭).
2. PDF를 저장하는 옵션을 클릭하십시오.
3. PDF 파일을 저장하려는 디렉토리를 탐색하십시오.
4. 저장을 클릭하십시오.

Adobe Reader 다운로드

이러한 PDF를 보거나 인쇄하려면 시스템에 Adobe Reader가 설치되어야 합니다. 사용자는 Adobe 웹 사이트 (www.adobe.com/products/acrobat/readstep.html)  에서 무료로 다운로드할 수 있습니다.

코드 면책사항 정보

이 문서에는 프로그래밍 예제가 들어 있습니다.

IBM®은 사용자의 특정 요구에 맞게 유사한 기능을 생성할 수 있도록 모든 프로그래밍 코드 예제를 사용할 수 있는 비독점적인 저작권 라이선스를 부여합니다.

모든 샘플 예제는 IBM에 의해 예시 목적으로만 제공됩니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다.

여기에 포함된 모든 프로그램은 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 일체의 보증 없이 "현상태대로" 제공됩니다.

제 1 장 분산 관계형 데이터베이스와 iSeries 서버

iSeries 서버*의 분산 관계형 데이터베이스 지원은 IBM* 분산 관계형 데이터베이스 구조(DRDA)*(DRDA*)의 구현과 어플리케이션 리퀘스터 드라이버(ARD) 프로그램을 사용하는 다른 SQL 클라이언트의 통합으로 구성됩니다. Operating System/400®(OS/400), iSeries용 DB2 UDB 조회 관리자 및 SQL 개발 킷이 결합되어 이 지원을 제공합니다.

이 장에서는 분산 관계형 데이터베이스와 이 데이터베이스가 iSeries 서버에서 사용되는 방법을 설명합니다. 다음 주제에 설명되어 있는 분산 관계형 데이터베이스의 일부 일반적인 개념을 정의합니다.

- 분산 관계형 데이터베이스 처리
- 분산 관계형 데이터베이스 구조(DRDA) 지원
- DRDA 및 CDRA 지원
- 어플리케이션 리퀘스터 드라이버 프로그램
- iSeries 서버의 분산 관계형 데이터베이스
- iSeries 분산 관계형 데이터베이스(DRDB) 관리

이 주제 외에도, 예: Spiffy사의 분산 관계형 데이터베이스도 기술되어 있습니다. 이 가상의 회사는 분산 관계형 데이터베이스 어플리케이션 프로그램에서 iSeries 서버를 사용합니다. Spiffy사의 샘플은 이 매뉴얼에서 사용되는 모든 샘플의 백그라운드가 됩니다.

분산 관계형 데이터베이스 처리

관계형 데이터베이스는 컴퓨터에 있는 하나 이상의 표에 저장된 자료의 집합입니다. 표는 표 1에서처럼 수평 행과 수직 열로 구성된 자료의 2차원 배열입니다. 각 행에는 표의 각 열마다 하나씩 순서 값이 있습니다. 열에는 이름과 특정 자료 유형이 있습니다(예를 들어, 문자, 소수 또는 정수).

표 1. 일반적 관계 표

항목	이름	공급자	수량
78476	야구	ACME	650
78477	미식 축구	특대	228
78478	농구	ACME	105
78479	축구	ACME	307

서버에서는 표를 여러 방법으로 정의하고 액세스할 수 있습니다. 서버에서 표를 설명하고 액세스하는 한 가지 방법은 SQL 같은 언어를 사용하는 것입니다. SQL은 표준 IBM 데이터베이스 언어이며 다른 서버에서 분산 자료 처리를 가능하게 하는 데 필요한 일관성을 제공합니다.

서버에서 표를 설명하고 액세스하는 다른 방법은 자료 서술 스펙(DDS)을 사용하여 실제 및 논리 파일을 설명하고 파일 인터페이스(예: 읽고 기록하는 고급 언어(HLL) 명령문)를 사용하여 표에 액세스하는 것입니다.

SQL은 iSeries 서버에서 사용되는 것과는 다른 용어를 사용합니다. 대부분의 SQL 오브젝트의 경우, iSeries 서버에 이에 대응하는 서버 오브젝트가 있습니다. 표 2는 SQL 관계형 데이터베이스 용어와 iSeries 서버 용어 사이의 관계를 보여줍니다.

표 2. SQL 용어와 시스템 용어의 관계

SQL 용어	시스템 용어
<p>관계형 데이터베이스. 표의 집합으로 인식되고 관계형 자료 모델에 따라 조작되는 데이터베이스입니다. 사용자가 iSeries™ 서버에서 액세스할 수 있는 관계형 데이터베이스는 시스템 용어 열 아래 나열된 대로 세가지가 있습니다. 자세한 정보는 iSeries Information Center의 관계형 데이터베이스 주제를 참조하십시오.</p>	<p>시스템 관계형 데이터베이스 또는 시스템 데이터베이스. iSeries 서버에 접근할 수 있는 모든 데이터베이스 오브젝트로 독립형 보조 기억장치 풀에 저장되어 있지 않은 오브젝트.</p>
<p>스키마. 라이브러리, 저널, 저널 리시버, SQL 카탈로그 및 선택형 자료 사전으로 구성됩니다. 스키마는 관련 오브젝트를 그룹화하며 이름으로 오브젝트를 찾을 수 있습니다.</p>	<p>사용자 관계형 데이터베이스 또는 사용자 데이터베이스. 하나의 독립형 보조 기억장치 풀 그룹에 존재하는 모든 데이터베이스 오브젝트와 독립형 보조 기억장치 풀에 저장되어 있지 않은 데이터베이스 오브젝트. 주: V5R2에서 iSeries 서버는 독립형 보조 기억장치 풀이 서버에 구성되어 있을 경우, 다중 관계형 데이터베이스의 호스트가 됩니다. 시스템 관계형 데이터베이스는 항상 하나가 존재하고 사용자 관계형 데이터베이스는 하나 이상 존재할 수 있습니다. 각 사용자 데이터베이스는 시스템 데이터베이스의 모든 오브젝트를 포함합니다.</p>
<p>주: 또 스키마를 콜렉션으로서 공통으로 참조할 수 있습니다.</p>	<p>주: 하지만 사용자는 SQL에서는 시스템 데이터베이스가 사용자 데이터베이스 내에 포함되는 것으로 볼 수 있지만, 확약 제어에서는 시스템 데이터베이스가 별개의 데이터베이스로 다루어진다는 것을 명심해야 합니다. 자세한 정보는 iSeries Information Center의 트랜잭션 및 확약 제어 주제를 참조하십시오.</p>
<p>표 열과 행의 세트. 행. 열의 세트가 있는 표의 수평 부분. 열. 한 자료 유형의 표의 수직 부분. 뷰. 하나 이상의 표에 있는 열과 행의 서브세트. 색인. 표의 열에 있는 자료의 콜렉션을 오름차순이나 내림차순으로 논리적으로 배열한 것. 패키지. 어플리케이션 서버가 사용하는 SQL문에 대한 제어 구조가 들어 있는 오브젝트.</p>	<p>리모트 관계형 데이터베이스 또는 리모트 데이터베이스. 리모트로 액세스할 수 있는 iSeries 서버 또는 다른 서버에 상주하는 데이터베이스 라이브러리. 관련 오브젝트를 그룹화하며 이름으로 오브젝트를 찾을 수 있습니다.</p>
<p>카탈로그. 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가 들어 있는 표와 뷰의 세트. QSYS2의 카탈로그 뷰에는 iSeries 서버의 모든 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가 들어 있습니다. 또한 SQL 스키마에는 스키마의 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보만 들어 있는 뷰의 집합이 있습니다.</p>	<p>실제 파일(PF). 레코드 세트. 레코드. 필드 세트. 필드. 자료 유형 관련 정보의 하나 이상의 바이트. 논리 파일. 최대 32개 실제 파일(PF)의 필드 및/또는 레코드의 서브세트. 논리 파일의 유형</p>
<p>패키지. 어플리케이션 서버가 사용하는 SQL문에 대한 제어 구조가 들어 있는 오브젝트.</p>	<p>SQL 패키지. SQL 용어와 같은 의미.</p>
<p>카탈로그. 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가 들어 있는 표와 뷰의 세트. QSYS2의 카탈로그 뷰에는 iSeries 서버의 모든 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가 들어 있습니다. 또한 SQL 스키마에는 스키마의 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보만 들어 있는 뷰의 집합이 있습니다.</p>	<p>상이한 오브젝트. 그러나 DSPFD(화면 파일 설명)과 DSPFFD(화면 파일 필드 서술) 명령은 SQL 카탈로그가 제공하는 조회와 동일한 정보를 일부를 제공합니다.</p>

분산 관계형 데이터베이스(RDB)는 자료와 자료 자체를 사용하는 어플리케이션 프로그램이 다른 시스템에 위치할 때 또는 프로그램이 한 서버의 다중 데이터베이스에 있는 자료를 사용할 때 존재합니다. 후자의 경우, DRDA 프로토콜이 하나의 서버에서 하나 또는 그 이상의 데이터베이스를 액세스하는데 사용된다는 점에서 데이터베이스가 분산되어 있는 것입니다. 이런 환경에서 데이터베이스 연결은 두 가지 유형 중 하나가 됩니다.

로컬 또는 DRDA입니다. 한 번에 많아야 하나의 로컬 데이터베이스 연결이 존재합니다. 분산 관계형 데이터베이스(RDB)의 간단한 형식중 하나는 그림 1에서 설명되는데 어플리케이션 프로그램은

분산 관계형 데이터베이스(RDB)를 사용할 때 어플리케이션 프로그램이 실행되는 시스템을 어플리케이션 리퀘스터(AR)라고 하며 리모트 자료가 있는 시스템을 어플리케이션 서버(AS)라고 합니다. 종종 ‘클라이언트’라는 용어는 AR과 바꿔 쓰고 ‘서버’는 AS로 바꿔 씁니다.

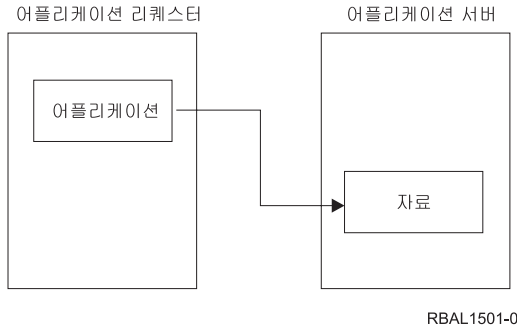


그림 1. A 분산 관계형 데이터베이스(DRDB)

작업 단위는 하나 이상의 데이터베이스 요구이며 그림 2에서처럼 완료된 작업을 구성하는 관련 처리입니다. 간단한 예는 재고 관리 제어 어플리케이션 프로그램의 재고에서 한 부분을 취한 것입니다. 재고 관리 프로그램은 공장 재고 계정표에서 항목을 임시로 제거한 다음 해당 항목을 같은 위치의 부품 기록표에 추가할 수 있습니다. ‘트랜잭션’ 용어는 작업 단위 개념을 설명하는 데 사용되는 또다른 표현식입니다.

위의 예에서 작업 단위는 부품을 공장 재고 계정표에서 삭제하고 기록표에 추가할 때까지 완료되지 않습니다. 요구가 완료되면 어플리케이션 프로그램은 작업 단위를 확약할 수 있습니다. 이는 작업 단위와 관련된 데이터베이스 변경이 영구적으로 이루어졌다는 의미입니다.

작업 단위 지원으로 어플리케이션 프로그램은 작업 단위로 변경사항을 롤백할 수 있습니다. 작업 단위가 롤백 되면 마지막 확약이나 롤백 연산이 적용되지 않았기 때문에 변경이 이루어집니다. 따라서 어플리케이션 프로그램은 데이터베이스에 대한 요구 집합을 단위로 처리합니다.

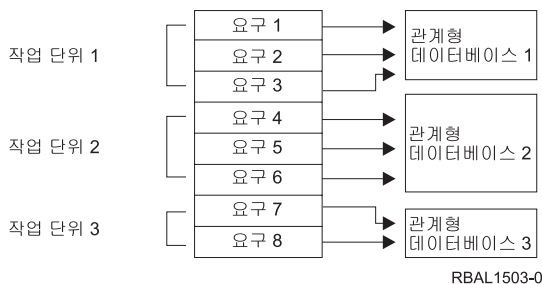


그림 2. 로컬 관계형 데이터베이스의 작업 단위

작업 단위의 자세한 내용은 다음 주제들을 참조하십시오.

- 리모트 작업 단위

- 분산 작업 단위
- 기타 분산 관계형 데이터베이스 용어 및 개념

리모트 작업 단위

리모트 작업 단위(RUW)는 어플리케이션 프로그램이 작업 단위 내의 리모트 데이터베이스에 있는 자료에 액세스할 수 있는 분산 관계형 데이터베이스 처리의 한 형식입니다. 리모트 작업 단위에 하나 이상의 관계형 데이터베이스 요구가 있지만 모든 요구는 동일한 리모트 데이터베이스에 대한 것이어야 합니다. 관계형 데이터베이스에 대한 모든 요구는 다른 관계형 데이터베이스로 보내기 전에 완료(확약 또는 롤백)되어야 합니다. 리모트 작업 단위는 그림 3에 있습니다.

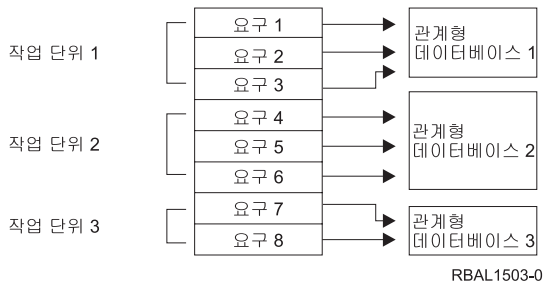


그림 3. 분산 관계형 데이터베이스(DRDB)의 리모트 작업 단위

리모트 작업 단위는 어플리케이션 프로그램이 요구하기 전에 정확한 관계형 데이터베이스 시스템에 연결해야 하기 때문에 어플리케이션 지향 분배입니다. 그러나 어플리케이션 프로그램에는 정확한 연결을 위한 리모트 데이터베이스의 이름만 필요합니다.

리모트 작업 단위를 지원하면 어플리케이션 프로그램이 하나 이상의 위치에 있는 자료를 읽고 갱신할 수 있습니다. 그러나 작업 단위 내에서 프로그램이 액세스하는 모든 자료는 같은 관계형 데이터베이스 관리 시스템에 의해 관리되어야 합니다. 예를 들어, 공장 재고 어플리케이션 프로그램은 다른 위치에 있는 표를 읽거나 갱신하기 전에 재고와 미수금(AR) 관리 작업 단위를 확약해야 합니다.

리모트 작업 단위 처리에서 각 컴퓨터에는 분산 관계형 자료 요구를 처리해 주는 관련 관계형 데이터베이스 관리 시스템과 관련 어플리케이션 리퀘스터가 있습니다. 따라서 사용자나 어플리케이션 프로그램은 로컬 관계 자료를 요구할 때와 거의 같은 방법으로 리모트 관계 자료를 요구할 수 있습니다.

분산 작업 단위

분산 작업 단위(DUW)를 사용하면 사용자나 어플리케이션 프로그램이 그림 4에서처럼 작업 단위 내의 여러 위치에서 자료를 읽거나 갱신할 수 있는 작업 단위 내의 시스템에서 실행되는 어플리케이션은 시스템이 지원하는 SQL을 사용하여 SQL 요구를 복수 리모트 데이터베이스 관리 시스템으로 보냅니다. 예를 들어 공장 재고 관리 프로그램은 한 시스템의 재고표와 작업 단위 내의 다른 시스템에 있는 미수금 표에 대한 갱신을 실행할 수 있습니다.

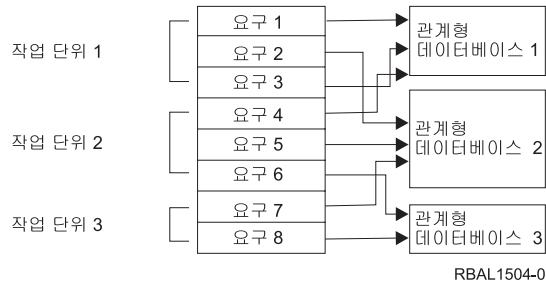


그림 4. 분산 관계형 데이터베이스(DRDB)의 분산 작업 단위

CONNECT TO와 SET CONNECTION과 같은 SQL문을 사용하여 요구의 목표를 사용자나 어플리케이션으로 제어합니다. 각 SQL문은 한 위치에 있는 자료를 참조해야 합니다.

어플리케이션이 작업을 확약할 준비가 되면 확약을 시작합니다. 확약 조정은 동기화(SYN) 점 관리자가 실행합니다.

분산 작업 단위는 다음 사항을 허용합니다.

- 한 작업 단위의 여러 데이터베이스 관리 시스템에 대한 액세스를 갱신하거나
- 한 작업 단위의 다른 데이터베이스 관리 시스템에 대한 읽기 액세스로 하나 이상의 데이터베이스 관리 시스템에 대한 액세스를 갱신합니다.

어플리케이션이 작업 단위의 주어진 데이터베이스 관리 시스템을 갱신할 수 있는지의 여부는 DRDA의 레벨 (DRDA가 리모트 관계형 데이터베이스에 액세스하는 데 사용될 경우)과 연결 및 갱신이 이루어지는 순서에 따라 다릅니다.

기타 분산 관계형 데이터베이스 용어 및 개념

다음 설명은 추가 분산 관계형 데이터베이스(RDB) 개념의 개요입니다. IBM 시스템에서 일부 분산 관계형 데이터베이스 지원이 DB2® Relational Connect 제품 및 DataPropagator™ Relational 제품에 의해 제공됩니다. 또한 AS/400 어플리케이션 프로그램을 쓸 때 이 개념의 일부를 사용할 수도 있습니다.

OS/400® V5R1 시작 시, iSeries용 DB2 UDB는 APPC 및 TCP/IP 통신을 사용하여 리모트 작업 단위와 분산 작업 단위를 지원합니다. 분산 작업 단위를 넘는 복잡한 처리는 분산 요구입니다. 분산 관계형 데이터베이스(RDB) 액세스 요구 유형을 사용하면 사용자나 어플리케이션 프로그램은 여러 위치에서 자료를 읽거나 갱신할 수 있는 단일 SQL문을 제공할 수 있습니다.

분산 관계형 데이터베이스(RDB)의 표는 다른 표와 구분할 필요가 없습니다. 일부 표는 다른 표와 일치하거나 일부 복사되는 경우가 있습니다. 추출, 스냅샷 및 복제는 분산 처리를 사용하는 복사 유형을 설명하는 용어입니다.

추출은 표의 사용자 요구 복사입니다. 복사는 하나의 데이터베이스에서 추출되고 사용자가 지정한 다른 데이터베이스로 로드됩니다. 언로드와 로드 프로세스는 갱신된 자료를 얻기 위해 주기적으로 반복될 수 있습니다. 추출은 거의 변경되지 않는 자료의 읽기 전용 복사와 같이 한 번 또는 자주 발생하지 않는 경우에 유용합니다.

스냅샷은 서버가 자동으로 실행하는 표의 읽기 전용 복사입니다. 서버는 매일, 매주 또는 매달 등 사용자가 지정한 주기에 따라 소스 표로부터 이 사본을 화면정리합니다. 스냅샷은 주기적으로 갱신된 정보를 수신하기 위해 자동 프로세스를 탐색하는 위치에 가장 유용합니다.

자료 복제는 서버가 표의 복사를 자동으로 갱신하는것을 의미합니다. 표의 복사는 여러 위치에 저장되기 때문에 스냅샷과 유사합니다. 자료 복제는 거의 갱신하지 않고 높은 신뢰성과 빠른 자료 검색을 요구하는 상황에 가장 효과적입니다.

표는 네트워크의 컴퓨터 서버에 분할될 수도 있습니다. 이 표를 분산 표라고 합니다. 분산 표는 행이나 열에 의해 수평이나 수직으로 분할되어 쉬운 로컬 참조와 기억장치를 제공합니다. 수직으로 분배된 표의 열은 수평으로 분배된 표의 행과 마찬가지로 여러 위치에 있습니다. 사용자는 표를 어떤 위치에서도 한 위치에 있는 것처럼 볼 수 있습니다. 분배된 표는 특정 부분에 대한 액세스나 갱신을 표의 일부가 있는 같은 위치에서 요구할 경우에 가장 효과적입니다.

추가 용어에 관해서는 용어집을 참조하십시오.

분산 관계형 데이터베이스 구조(DRDA) 지원

분산 관계형 데이터베이스(RDB) 처리를 위한 DRDA 지원은 IBM 관계형 데이터베이스 제품에 의해 사용됩니다. DRDA 지원은 어플리케이션 프로그램과 리모트 관계형 데이터베이스 사이의 통신 프로토콜을 정의합니다.

| DRDA 지원은 IBM과 비IBM 환경에서 분산 관계형 데이터베이스(RDB) 관리를 제공합니다. IBM 환경에서
| 관계형 데이터베이스는 다음 프로그램으로 관리됩니다.

- | • DB2 Universal Database™ for iSeries
- | • DB2 Universal Database for z/OS®
- | • DB2 Universal Database for VSE/VM
- | • DB2 Universal Database for AIX®
- | • DB2 Universal Database for Linux™
- | • DB2 Universal Database for HP-UX
- | • DB2 Universal Database for Sun Solaris
- | • DB2 Universal Database for Windows®

| DRDA 지원은 유사 및 상이한 환경에서 운영되는 관계형 데이터베이스 관리자에게 데이터베이스 정보에 액세스
| 스하는 구조를 제공합니다. 예를 들어, 둘 이상의 DB2 UDB for iSeries 사이의 관계 자료에 대한 액세스는
| 유사 환경에서의 분배입니다. DB2 UDB for iSeries와 OS/400에 내장된 것과 다른 유형의 시스템이나 클라
| 이언트 간의 관계 자료에 대한 액세스는 상이한 환경에서의 분배입니다. DB2 UDB for iSeries와 IBM DB2
| Universal Driver for SQLJ 및 JDBC 간의 관계 자료에 대한 액세스가 그 예입니다. Universal Driver에
| 대한 자세한 정보는 IBM DB2 Universal Driver for SQLJ 및 JDBC 1.0을 참조하십시오.

SQL은 표준 IBM 데이터베이스 언어입니다. 이 언어는 유사한 운영 환경이나 상이한 운영 환경에서 분산 자료 처리를 가능하게 하는 데 필요한 일관성을 제공합니다. DRDA 지원 내에서 SQL을 사용하면 DRDA 구현을 지원하는 환경에서 자료를 정의, 검색 및 조작할 수 있습니다.

DRDA 및 CDRA 지원

분산 관계형 데이터베이스가 흥미로운 점 중 하나는 데이터베이스가 다른 유형의 컴퓨터에 스캔될 수는 없지만 다른 유형의 컴퓨터는 다른 국가나 영역에 있을 수 있다는 것입니다. iSeries 서버와 같은 동일한 서버는 시스템에서 사용되는 언어에 따라 자료를 다르게 인코딩할 수 있습니다. 다른 종류의 서버는 자료를 다르게 인코딩합니다. 예를 들어, System/390*, iSeries 서버 및 PS/2* 시스템은 고유한 형식으로 숫자 자료를 인코딩합니다. 또한 System/390®과 iSeries 서버는 문자 자료를 인코딩하는 데 EBCDIC 코드화 체계를 사용하며 PS/2® 시스템은 ASCII 코드화 체계를 사용합니다.

숫자 자료의 경우, 이 차이는 문제가 되지 않습니다. DRDA 지원을 제공하는 상이한 시스템은 한 컴퓨터의 숫자 표현 방식과 다른 컴퓨터의 숫자 표현 방식 간의 차이를 자동으로 변환합니다. 예를 들어, iSeries 어플리케이션 프로그램이 iSeries용 DB2 UDB 데이터베이스로부터 숫자 자료를 읽을 경우, iSeries용 DB2 UDB는 System/390 형식으로 숫자 자료를 보내고 OS/400 데이터베이스 관리 시스템은 이 자료를 iSeries 숫자 형식으로 변환합니다.

문자 자료의 처리는 더 복잡하지만 이 또한 분산 관계형 데이터베이스 내에서 처리할 수 있습니다. 문자 자료 처리에 대한 자세한 정보는 CDRA의 문자 변환을 참조하십시오.

CDRA를 사용한 문자 변환

코드화 체계(EBCDIC(Extended Binary Coded Decimal Interchange Code)와 ASCII(American Standard Code for Information Interchange))의 차이점 뿐 아니라 언어와 관련된 차이점이 있습니다. 예를 들어, 다른 언어로 구성된 시스템은 같은 코드에 다른 문자를 할당하거나 같은 문자에 다른 코드를 할당할 수 있습니다. 예를 들어, 영어로 구성된 시스템은 덴마크어로 구성된 시스템이 å에 할당한 } 문자에 같은 코드를 할당할 수 있습니다. 이 두 시스템이 같은 문자(예: \$)에 다른 코드를 할당할 수 있습니다.

다른 서버 간에 자료를 공유하는 경우, 사용자와 어플리케이션이 같은 방식으로 문자 자료를 볼 수 있어야 합니다. 즉, 뉴욕의 PS/2 사용자와 코펜하겐의 iSeries 서버 사용자는 각자의 서버에서 \$가 다르게 인코딩되더라도 \$를 \$로 볼 수 있어야 합니다. 비록 코드가 덴마크에서는 å와 같을 지라도 뉴욕에서 저장된 문자의 경우, 코펜하겐의 사용자는 }를 볼 수 있어야 합니다. 이렇게 하려면 \$를 PS/2 시스템에 맞는 문자 코드로 변환하고(즉, 영어 문자 세트, ASCII) 뉴욕에서 코펜하겐으로 오면 다시 덴마크 코드로 변환해야 합니다(즉, 덴마크어 문자 EBCDIC). 이런 종류의 문자 변환은 다른 IBM 분산 관계형 데이터베이스 관리자뿐 아니라 iSeries 서버도 제공합니다. 이 변환은 문자 자료 표시 구조(CDRA)에 따라 통일된 방식으로 실행됩니다.

CDRA는 서버가 다른 문자 세트와 코드화 체계를 사용하더라도 서버에서 자료가 이해되도록 문자 자료의 속성을 식별하는 방식을 지정합니다. 서버에서 변환을 실행하려면 각 서버는 다른 서버에서 수신된 문자 자료의 속성을 이해해야 합니다. CDRA는 이 속성을 코드화 문자 세트 ID(CCSID)로 식별하도록 지정합니다. DB2 UDB for z/OS, DB2 UDB for VM 및 OS/400 데이터베이스 관리 시스템의 모든 문자 자료는 코드화 체계, 문자 세트 및 코드 페이지의 특정 조합을 나타내는 CCSID를 갖고 있습니다. 확장 서비스® 환경의 모든

문자 자료에는 코드 페이지만 있습니다(하지만 다른 데이터베이스 관리자는 해당 코드 페이지 ID를 CCSID로 처리합니다).코드 페이지는 문자와 내부 코드 사이의 특정 할당된 세트입니다.

예를 들어, CCSID 37은 코드화 체계 4352(EBCDIC), 문자 세트 697(라틴, 1바이트 문자) 및 코드 페이지 37(미국 및 캐나다 국가 확장 코드 페이지)을 의미합니다. CCSID 5026은 코드 페이지가 290(가다가나(katakana)와 간지(kanji)용 1바이트 문자 세트)인 코드화 체계 4865(확장 EBCDIC) 문자 세트 1172이며, 코드 페이지가 300(가다가나와 간지용 2바이트 문자 세트)인 문자 세트 370을 의미합니다.

DRDA 사용 가능 시스템에는 CCSID-CCSID 쌍과 CCSID-코드 페이지 쌍의 넓은 범위 사이에 문자 자료를 변환하는 메커니즘이 있습니다. 여러 CCSID와 코드 페이지의 문자 변환이 이미 이 제품에 빌드되어 있습니다. iSeries에서 지원되는 CCSID에 대한 자세한 정보는 iSeries Information Center의 OS/400 국제화 주제를 참조하십시오. iSeries 서버에서 CCSID를 사용하는 방법에 대한 설명은 209 페이지의 『코드화 문자 세트 ID(CCSID)』를 참조하십시오.

어플리케이션 리퀘스터 드라이버 프로그램

어플리케이션 리퀘스터 드라이버(ARD) 프로그램은 SQL 어플리케이션이 iSeries용 DB2 UDB 이외의 데이터베이스 관리 시스템이 관리하는 자료에 액세스할 수 있도록 해주는 일종의 종료 프로그램입니다. iSeries 클라이언트는 다음 조작을 하는 동안 ARD 프로그램을 호출합니다.

- 관계형 데이터베이스(RDB) 매개변수가 ARD 프로그램에 대응하는 RDB 이름과 일치할 때 CRTSQLPKG(Create Structured Query Language Package) 명령 또는 CRTSQLxxx 명령을 사용하여 실행되는 SQL 사전컴파일의 패키지 작성 단계.
- 현재 연결이 ARD 프로그램에 대응하는 이름일 때의 SQL문의 처리.

호출을 사용하면 ARD 프로그램이 리모트 관계형 데이터베이스 명령문에 대한 SQL문과 정보를 전달하고 결과를 어플리케이션 리퀘스터(AR)에 리턴할 수 있습니다. 그러면 AR은 결과를 어플리케이션이나 사용자에게 리턴합니다. ARD 프로그램이 액세스한 관계형 데이터베이스에 대한 액세스는 상이한 환경에서의 DRDA 어플리케이션 서버에 대한 액세스와 비슷합니다.

ARD 프로그램은 ADDRDBDIRE(Add Relational Database Directory Entry) 명령을 사용하여 시스템에 등록됩니다. 지정된 매개변수 중 하나는 프로그램이 위치해 있는 라이브러리입니다. 독립형 보조 기억장치 풀로 구성되는 시스템에 대해 ARD 프로그램은 시스템 데이터베이스의 라이브러리에 상주해야 합니다(여기서 라이브러리는 시스템 ASP 또는 구성된 기본 ASP의 일부).

어플리케이션 리퀘스터 드라이버 프로그램에 대한 자세한 정보는 iSeries Information Center의 어플리케이션 프로그래밍 인터페이스(API) 주제를 참조하십시오.

iSeries 서버의 분산 관계형 데이터베이스

iSeries용 DB2 UDB는 iSeries 시스템 관계형 데이터베이스 및 임의의 구성된 사용자 데이터베이스에 모든 데이터베이스 관리 기능을 제공합니다. 시스템의 분산 관계형 데이터베이스 지원은 통신, 작업 관리, 보안 기능 및 다른 기능 지원과 마찬가지로 OS/400 프로그램의 통합 부분입니다.

iSeries 시스템은 DRDA 구현을 지원하는 다른 서버를 갖는 분산 관계형 데이터베이스(RDB) 네트워크의 부분입니다. iSeries 시스템은 유사 또는 상이 환경 둘 중 하나에서 어플리케이션 리퀘스터(AR) 또는 어플리케이션 서버(AS)가 됩니다. iSeries 시스템의 분산 관계형 데이터베이스 구현은 리모트 작업 단위(RUW)와 분산 작업 단위(DUW)를 지원합니다. RUW로 단일 작업 단위 내에서 단일 데이터베이스에 대한 복수 요구를 제출할 수 있으며, DUW로 복수 데이터베이스를 단일 작업 단위 안에 포함시키도록 요구할 수 있습니다.

예를 들어, DUW 지원을 사용하면 작업 단위 내의 한 서버에서 부품의 재고 수를 하나씩 감소시키거나 다른 서버에서 부품의 재고 수를 하나씩 증가시킨 다음, 2단계 확약 프로세스를 사용하여 단일 작업 단위가 끝날 때 이 리모트 데이터베이스에 대한 변경을 확약할 수 있습니다. iSeries용 DB2 UDB는 분산 요구를 지원하지 않으므로 각 SQL문으로 한 데이터베이스만 액세스할 수 있습니다. 어플리케이션 프로그램에 제공된 지원 레벨은 어플리케이션 서버(AS)에서 사용할 수 있는 지원 레벨과 연결 및 갱신이 이루어진 순서에 따라 다릅니다. 보다 자세한 정보는 196 페이지의 『분산 관계형 데이터베이스(DRDB)에 연결』을 참조하십시오.

DRDA 액세스 외에 DRDA를 지원하지 않는 데이터베이스에 액세스하는 데 ARD 프로그램을 사용할 수 있습니다. ARD 프로그램을 통해 액세스한 관계형 데이터베이스의 연결은 상이한 서버 연결과 유사하게 처리됩니다. 이 연결은 DRDA 어플리케이션 서버 연결, 로컬 관계형 데이터베이스 연결 및 다른 ARD 프로그램에 액세스하는 연결과 함께 존재할 수 있습니다.

5 페이지의 『기타 분산 관계형 데이터베이스 용어 및 개념』에서 소개한 iSeries 서버의 스냅샷과 복제의 분배 기능은 서버에 의해 자동으로 실행되지 않습니다. iSeries 서버에 DataPropagator Relational Capture and Apply 제품을 설치하고 구성하여 이 기능을 실행할 수 있습니다. 또한 사용자가 작성한 어플리케이션 프로그램에서 이 기능을 사용할 수 있습니다. 분산 관계형 데이터베이스(RDB)에서 이 기능을 구성하는 방법에 대한 자세한 정보는 제 7 장 『분산 관계형 데이터베이스에 대한 자료 가용성 및 보호』에서 설명됩니다.

5 페이지의 『기타 분산 관계형 데이터베이스 용어 및 개념』에서 설명된 iSeries 서버의 분산 요구 기능은 직접 지원되지 않습니다. 그러나 DataJoiner[®] 제품은 다양한 자료 소스로부터 표를 결합하여 분산 조회를 실행할 수 있습니다. DataJoiner는 IBM Information Warehouse 제품군에 있는 포괄적인 정보 카탈로그인 DataGuide와 협조하여 작업합니다. DataGuide는 그래픽 사용자 인터페이스를 회사의 자료 자원에 대한 완료 정보 리스트에 제공합니다.

OS/400 프로그램에는 SQL에 대한 실시간 지원이 포함됩니다. iSeries용 DB2 UDB 어플리케이션 리퀘스터(AR)나 어플리케이션 서버(AS)에 iSeries용 DB2 UDB 조회 관리자 및 SQL Development Kit 사용권 프로그램이 설치되어 있지 않아도 분산 관계형 데이터베이스(RDB) 요구를 처리하거나 iSeries 서버에서 SQL 쿼리를 작성할 수 있습니다. 그러나, SQL문으로 프로그램을 컴파일하거나, 대화식 SQL을 실행하거나 iSeries용 DB2 UDB 조회 관리자를 실행하는 데는 iSeries용 DB2 UDB 조회 관리자 및 SQL Development Kit 프로그램이 필요하지 않습니다.

iSeries 분산 관계형 데이터베이스(DRDB) 관리

iSeries 서버에서 분산 관계형 데이터베이스(RDB)를 관리하는 데 OS/400 사용권 프로그램 내의 자원과 톨에 대한 폭 넓은 지식이 필요합니다. 이 설명서는 iSeries 서버에서 분산 관계형 데이터베이스(RDB)를 관리하는 데 도움을 주는 오퍼레이팅 시스템으로 사용할 수 있는 여러 기능에 대한 개요를 제공합니다. 이 안내서는 iSeries 서버(유사 환경) 네트워크의 분산 관계형 데이터베이스(RDB) 기능과 작업을 설명합니다. 유사 및 상이한 환경의 iSeries 분산 관계형 데이터베이스(RDB) 기능 사이의 차이점은 이 안내서에서 일반적인 설명으로만 제시됩니다.

분산 관계형 데이터베이스를 적절히 구현하면 리모트 서버에서 데이터베이스에 쉽게 액세스하고, 상주 위치를 알지 않아도 데이터베이스 파일을 쉽게 처리하고, 어플리케이션 프로그램에 대해 변경없이 하나의 서버를 다른 데이터베이스로 쉽게 이동시킬 수 있습니다.

분산 관계형 데이터베이스를 효과적으로 구현하려면 다음과 같은 주요 영역의 요구사항을 잘 알아야 합니다.

- 분산 관계형 데이터베이스 계획 및 설계에서는 분산 데이터베이스를 계획하고 설계할 때 고려해야 할 중요한 사항에 대해 설명합니다.
- iSeries 분산 관계형 데이터베이스 통신은 네트워크를 설정하거나 분산 관계형 데이터베이스(RDB)에 대한 작업을 위해 기존 네트워크를 변경할 때 사용하는 통신 기능에 대해서 설명합니다.
- iSeries 분산 관계형 데이터베이스 보안은 리모트 관계형 데이터베이스에 대한 통신 및 DRDA 액세스를 비롯한 iSeries 분산 관계형 데이터베이스의 보안 사항에 관한 정보를 제공합니다.
- iSeries 분산 관계형 데이터베이스 설정에서는 자료를 분산 데이터베이스에 입력하는 방법에 대한 자세한 내용과 iSeries 서버의 서브시스템과 관계형 데이터베이스 디렉토리에 대한 설명을 제공합니다.
- 분산 관계형 데이터베이스 관리 및 조작 TASK에서는 사용자가 네트워크 전체를 통해 실행 중인 분산 관계형 데이터베이스 작업을 관리할 수 있는 방법에 대해 다루고 있습니다.
- 분산 관계형 데이터베이스의 자료 가용성 및 보호에서는 iSeries 서버의 프로그램과 자료를 보호하고 문제가 생겼을 때 회복되는 시간을 줄이는 톨과 기술에 대해서 이야기하고 있습니다. 또한 여기서는 네트워크 사용자가 필요할 때에 해당 네트워크 전역에서 관계형 데이터베이스와 표에 액세스하도록 보장하는 대안에 대한 내용을 제공합니다.
- 분산 관계형 데이터베이스 성능은 사용자의 네트워크, 시스템 및 사용자의 데이터베이스 설계를 개선시킬 수 있는 방법을 다룹니다.
- 분산 관계형 데이터베이스 문제점 처리는 분산 관계형 데이터베이스에서 발생할 수 있는 문제점 일부와 이 문제점을 해결하는 방법을 다룹니다.
- 분산 관계형 데이터베이스 어플리케이션 쓰기는 분산 관계형 데이터베이스(RDB)의 프로그래밍 문제의 개요를 설명합니다.

iSeries 분산 관계형 데이터베이스(RDB)로 작업하는 다른 분산 관계형 데이터베이스(RDB) 플랫폼에 대한 고려사항은 253 페이지의 부록 B 『DRDA를 사용하는 공통 플랫폼 액세스』에서 설명됩니다.

DRDA를 지원하는 다른 IBM 시스템에 대한 자세한 정보는 해당 시스템에 제공된 정보나 295 페이지의 부록 E 『관련 정보』의 분산 관계형 데이터베이스 라이브러리 및 Other IBM 분산 관계형 데이터베이스 플랫폼 라이브러리에 있는 설명서를 참조하십시오.

예: Spiffy사의 분산 관계형 데이터베이스

분산 관계형 데이터베이스(RDB) 지원을 설명하는 데 Spiffy사가 여러 IBM 매뉴얼에서 사용되었습니다. 이 매뉴얼에서 가상의 회사는 iSeries 서버 네트워크의 DRDA에 대한 iSeries 서버 지원을 설명하기 위해 다소 변경되었습니다. 이 매뉴얼 전체에 사용된 예는 특정 기능, 연결 및 프로세스를 설명합니다. 이 예는 다른 분산 관계형 데이터베이스(RDB) 설명서에서 사용된 예와 정확하게 일치하지는 않지만 유사하게 작성하였습니다.

spiffy사는 가상 기업이지만 여기에서 설명하는 업무는 구조가 유사한 여러 회사에서 사용하는 업무를 모델로 삼았습니다. 그러나 예로 든 회사조차도 분산 관계형 데이터베이스를 사용하여 할 수 있는 모든 것을 설명하지는 않습니다.

다음 주제는 Spiffy 조직과 분산 관계형 데이터베이스(RDB) 지원 사용에 관한 정보를 담고 있습니다.

- Spiffy 조직 및 시스템 프로파일
- spiffy사의 자동차 서비스 업무 프로세스
- spiffy사의 분산 관계형 데이터베이스(DRDB) 관리

Spiffy 조직 및 시스템 프로파일

spiffy사는 여러 다른 제품 중에서 지역 사무소와 로컬 판매 대리점 네트워크를 통해 고객에게 자동차를 판매하고 서비스하는 국내 제품 대리점입니다. 오늘날 자동차 산업의 경쟁이 치열하다고는 하지만 spiffy사와 같은 회사의 성공은 고품질 서비스와 예비 부품을 고객에게 적시에 전달하는 데 있습니다. 이 경쟁에 대응하기 위해 Spiffy사는 판매 대리점 조직 내에 방대한 서비스망을 빌드하였습니다.

판매 대리점 조직은 일리노이 주 시카고에 있는 중앙 차량 대리점이 이끌고 있습니다. 북아메리카에는 여러 개의 지역 유통 센터가 있습니다. 이 중 둘은 미네소타 주 미니애폴리스와 미조리 주 캔자스에 있습니다. 이 센터는 지역별 재고 목록을 만들어 차량과 부품의 유통 비용을 최소화하고 있습니다. 미니애폴리스 지역 센터에는 약 15개의 판매 대리점이 있고 캔자스에는 30개의 대리점이 있습니다.

『예: Spiffy사의 분산 관계형 데이터베이스』는 spiffy사에 대한 시스템 구성도입니다.

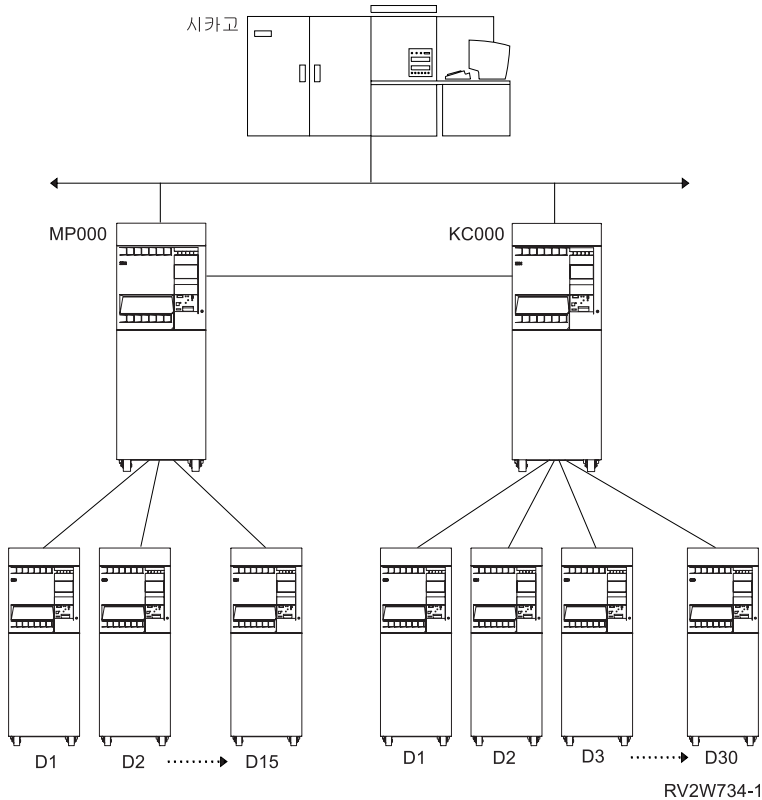


그림 5. spiffy사의 시스템 구성도

spiffy사는 전국적인 통합 텔레커뮤니케이션을 빌드중입니다. 자동차 부문의 경우, 지역 유통 센터와 판매 대리점에 iSeries 서버의 네트워크를 설정하고 있습니다. 이 네트워크는 중앙 차량 대리점에서 System/390에 연결됩니다. 이 네트워크는 경쟁에서의 우위를 유지하기 위해 중요한 업무 자산을 고려하였습니다.

중앙 차량 대리점은 해당 의사결정 지원 소프트웨어와 함께 System/390에서 z/OS용 DB2 UDB를 실행합니다. 이 시스템은 많은 어플리케이션 프로그램에서 한 번에 처리해야 하는 막대한 양의 자료 때문에 사용됩니다. 중앙 차량 대리점 시스템이 자동차 분야 자료 처리에만 쓰이지는 않습니다. 이 시스템은 아직 분산 데이터베이스 환경에서 운영하지 않는 기업의 작업과 프로세스를 처리해야 합니다. 지역 센터는 iSeries 시스템을 실행합니다. 여기서는 SDLC 프로토콜을 사용하여 SNADS와 5250 표시 장치 Pass-through를 갖춘 APPC/APPN을 사용합니다.

모든 판매 대리점에서는 크기가 다른 iSeries 서버를 사용하고 있습니다. 이 시스템은 SDLC 프로토콜을 사용하여 지역 오피스에 연결됩니다. 가장 큰 판매 대리점에는 기업의 자료 처리 기능을 계속하는 시간제 프로그래머와 시스템 오퍼레이터가 있습니다. 설치를 할 때 대부분의 경우에는 프로그래밍 전문가를 고용하지 않고 일부 작은 지점에는 일반적인 컴퓨터 지식을 갖춘 정도 이상의 직원은 고용하지 않습니다.

spiffy사의 자동차 서비스 업무 프로세스

spiffy사의 자동차 부문에는 분산 관계형 데이터베이스 환경에서 자동화되는 업무가 있습니다. 예가 필요 이상 복잡해지지 않도록 차량 서비스에 속하는 회사의 기능만 참고합니다.

판매 대리점에는 2,000명에서 20,000명까지의 고객 리스트가 있습니다. 소규모 대리점의 경우에는 하루 5번, 대규모 대리점의 경우에는 하루 50번 정도의 서비스 주문을 가정할 수 있습니다. 서비스 주문에는 예정된 유지 보수, 보증 수리, 일반 수리 및 부품 주문이 포함됩니다.

대리점은 자주 필요한 예비 부품만 갖고 있으며 재고 관리 데이터베이스를 유지보수합니다. 두 지역 센터 모두 필요할 때 부품을 제공합니다. 대리점은 예상 모델 제어 일괄처리로 주기적으로 재고를 들여 놓습니다.

spiffy사의 분산 관계형 데이터베이스(DRDB) 관리

각 대리점은 독립 기업으로 자료 처리 자원과 프로시저어를 관리합니다. spiffy사는 각 대리점이 하나 이상의 iSeries 서버를 가지고 서버가 특정 시간에 네트워크를 사용할 것을 요구합니다. 그러나, 서버의 크기와 자동화된 업무 프로세스의 수는 각 대리점의 요구와 사용할 수 있는 자원에 의해 결정됩니다.

spiffy사는 모든 대리점이 재고 관리 분산 관계형 데이터베이스를 사용할 것을 요구합니다. 회사는 자체 대리점을 운영하기 때문에 분산 관계형 데이터베이스 환경에 액세스하거나 액세스하지 않는 완전한 대리점 소프트웨어를 갖고 있습니다. spiffy사 대리점은 소프트웨어 툴 세트를 모두 사용합니다. 대부분의 개인 가맹점은 특별히 spiffy사의 업무 방식에 맞게 만들어졌기 때문에 이 소프트웨어를 사용합니다.

지역 유통 센터에서 해당 지역의 재고를 관리합니다. 또한 지역에서 사용되는 모든 분산 데이터베이스 자원의 데이터베이스 관리자로서의 기능을 합니다. 맡고 있는 책임은 각 대리점의 자료 처리 능력 수준에 따라 다릅니다. 지역 센터는 항상 지역 대리점을 돕기 위한 우선 연락처입니다.

미니애폴리스 지역 유통 센터에는 서버와 네트워크에 대한 폭 넓은 경험과 지식을 갖춘 iSeries 프로그래머가 있습니다. 대리점 로드는 다른 지역 센터의 약 절반 정도이므로 이 센터는 네트워크 iSeries 지원 기능에 집중할 수 있습니다. 이 기능에는 어플리케이션 프로그램 개발, 프로그램 유지보수 및 문제 처리가 포함합니다.

다음은 네트워크의 각 활동 레벨의 데이터베이스 책임 사항입니다.

대리점:

- 서버의 기본 조작과 관리 실행
- 로컬 사용자 등록

지역 유통 센터:

- 새 대리점의 자료 처리 설정
- 업무가 중단된 대리점에 대한 데이터베이스 자원 분배
- 지역의 네트워크 사용자 등록
- 지역의 재고 유지보수
- 대리점의 서비스 계획 개발
- 대리점의 헬프 데스크 운영

위의 지역 유통 센터 활동 이외에도 미니애폴리스 iSeries 서버 센터는 다음 활동을 실행합니다.

- iSeries 서버 어플리케이션 개발

- 지역 센터의 헬프 데스크 운영
- 데이터베이스 성능 조정
- 데이터베이스 문제 해결

이 매뉴얼 전체에 사용된 예는 하나 이상의 활동과 관련이 있습니다. 많은 예가 고객 서비스나 수리 스케줄로 인해 재고로부터 부품을 얻는 프로세스입니다. 다른 예는 spiffy사 분산 관계형 데이터베이스(RDB) 네트워크의 서버의 설정, 보안, 모니터 및 문제 해결에 사용되는 분산 관계형 데이터베이스(RDB) 관리 작업입니다.

제 2 장 분산 관계형 데이터베이스의 계획 및 설계

분산 관계형 데이터베이스의 성공적인 작업을 위한 첫 번째 요구사항은 계획을 통해 이루어집니다. 분산 관계형 데이터베이스의 사용 여부를 결정할 때는 기업의 요구와 목표를 고려해야 합니다. 어플리케이션 프로그램을 코딩하는 방법, 자료를 저장하는 장소, 어플리케이션 프로그램과 자료를 연결하는 네트워크 설계는 모두 설계의 중요 고려사항입니다.

분산 관계형 데이터베이스의 데이터베이스 설계는 하나의 iSeries 관계형 데이터베이스만 처리할 때 더 중요합니다. 하나 이상의 iSeries 서버를 고려할 때는 네트워크에 대한 일관성 있는 관리 전략을 개발해야 합니다. 전략을 세울 때 특별한 주의를 요구하는 조작용 다음과 같습니다.

- 일반 작업
- 네트워크 프로토콜
- 시스템 보안
- 계정
- 문제점 분석
- 백업 및 회복 프로세스

분산 관계형 데이터베이스를 준비하려면 업무의 요구와 관계형 데이터베이스 기술을 모두 이해해야 합니다.

분산 관계형 데이터베이스의 계획과 설계는 서로 밀접한 관련이 있기 때문에 이 장에서는 다음의 관련 타스크를 설명할 때 이 주제를 함께 설명합니다.

- 요구와 예상 식별
- 어플리케이션, 네트워크 및 자료의 설계
- 관리 전략 개발

분산 관계형 데이터베이스에 대한 요구와 예상 식별

분산 관계형 데이터베이스의 요구와 예상을 분석할 때 다음 사항을 고려하십시오.

- 분산 관계형 데이터베이스에 대한 자료 요구. 계획에 적합한 자료는 무엇이고, 필요로 하는 사용자는 누구이며, 필요로 하는 이유와 얼마나 자주 필요로 합니까?
- 분산 관계형 데이터베이스 기능.요구사항이 스스로 분산 관계형 데이터베이스 솔루션을 제공합니까?
- 분산 관계형 데이터베이스에 대한 목표 및 방향. 분산 관계형 데이터베이스가 필요한 솔루션일 경우, 어떤 단기 및 장기 목표가 만족될 수 있습니까?

분산 관계형 데이터베이스에 대한 자료 요구

분석의 첫 번째 단계는 자료에 영향을 주는 요소와 어떤 영향을 주는지 판별하는 것입니다. 자신에게 다음 질문을 해 보십시오.

- 관련된 위치는 무엇인가?
- 계획하고 있는 트랜잭션의 종류는 무엇인가?
- 각 트랜잭션에 필요한 자료는 무엇인가?
- 자료 항목이 서로에게, 특히 참조 제한에 의존하는 것은 무엇인가? 예를 들어, 한 표의 정보를 다른 표의 정보와 검사해야 하는가?(그렇다면 두 표를 같은 위치에 보유해야 합니다.)
- 현재 자료가 존재하는가? 존재한다면 위치는 어디인가? 누가 "소유"하고 있는가(즉, 자료의 정확도를 유지 보수하는 담당자는 누구인가)?
- 필요한 자료의 가용성에 어떤 우선순위를 두고 있는가? 위치에 따른 자료의 무결성은? 권한이 없는 액세스로부터 자료를 보호하는가?
- 자료에 대해 계획하고 있는 액세스 패턴은 무엇인가? 예를 들어, 자료를 읽거나 갱신할 것인지, 아니면 둘 다 할 것인가? 빈도는? 일반적인 액세스로 많은 자료를 리턴할 것인가, 적은 자료를 리턴할 것인가?
- 각 트랜잭션에서 예상하는 성능 레벨은? 허용할 수 있는 응답 시간은?

분산 관계형 데이터베이스 기능

분석의 두 번째 단계는 사용자의 자료 요구를 분산 관계형 데이터베이스 솔루션에 제공할 것인지의 여부를 결정하는 것입니다.

대부분의 데이터베이스 처리를 로컬로 실행하고 리모트 자료에 대한 액세스가 가끔만 필요한 어플리케이션이 보통 분산 관계형 데이터베이스의 훌륭한 후보자입니다.

다음 요구사항이 있는 어플리케이션은 일반적으로 분산 관계형 데이터베이스의 부적합한 후보자입니다.

- 자료를 중심부에 유지하고 그리고 리모트 사용자가 해야 할 대부분의 작업이 중심부에 있습니다.
- 지속적으로 높은 성능, 특히 빠른 응답 시간을 요구합니다. 네트워크에서 자료를 이동시키는 시간은 오래 걸립니다.
- 지속적으로 높은 가용성, 특히 24시간, 일주일 내내 가용성을 요구합니다. 네트워크는 많은 시스템과 중간에 통신 회선 및 통신 제어기와 같은 많은 구성요소를 포함하고 있어 고장날 기회가 증가합니다.
- 사용자가 필요로 하는 분산 관계형 데이터베이스는 현재 사용될 수 없으며 발표되지 않았습니다.

분산 관계형 데이터베이스에 대한 목표 및 방향

분석의 세 번째 단계는 단기 및 장기 목표를 평가하는 것입니다.

SQL은 표준 IBM 데이터베이스 언어입니다. 사용자의 목표와 방향이 상이한 시스템의 이식성이나 리모트 자료 액세스를 포함하고 있으면 iSeries 서버에서 분산 관계형 데이터베이스를 사용해야 합니다.

DataPropagator Relational Capture and Apply가 제공하는 추가 자료 복사 기능은 물론 분산 작업 단위의 분산 데이터베이스 기능은 사용자가 iSeries 서버에서 실행할 수 있는 활동 범위를 넓혀 줍니다. 그러나, 분산 데이터베이스 어플리케이션이 현재 iSeries 서버에서 사용할 수 없는 기능을 요구하면 오퍼레이팅 시스템에서 기능을 사용할 수 있을 때까지 다른 옵션을 사용할 수 있습니다. 예를 들면, 다음 중 하나를 실행할 수 있습니다.

- 필요한 기능을 스스로 제공합니다.
- 새 기능을 사용할 수 있도록 분산 관계형 데이터베이스의 계획을 마련합니다.
- 현재 사용할 수 있거나 발표된 기능으로 요구사항을 만족시킬 수 있는지 확인하기 위해 사용자의 목표와 요구사항을 재평가합니다. 일부 대체 솔루션이 표 3에 나열되어 있습니다. 이 대안은 사용할 수 있는 기능을 보충하거나 대체하는 데 사용될 수 있습니다.

표 3. 분산 관계형 데이터베이스에 대한 대안 솔루션

솔루션	설명	장점	단점
분산 자료 관리(DDM)	한 시스템의 어플리케이션 프로그램이나 사용자가 리모트 시스템에 저장된 데이터베이스 파일을 사용할 수 있도록 해주는 오퍼레이팅 시스템의 기능. 시스템은 통신 네트워크에 연결되어 있어야 하며 리모트 시스템은 DDM을 사용해야 합니다.	<ul style="list-style-type: none"> • 간단히 읽고 갱신하는 액세스의 경우, SQL보다 성능이 빠릅니다. • 기존 어플리케이션을 재작성할 필요가 없습니다. • S/38, S/36 및 CICS*를 액세스하는 데 사용될 수 있습니다. 	<ul style="list-style-type: none"> • SQL은 복잡한 기능에 보다 효율적입니다. • 다른 분산 관계형 데이터베이스 플랫폼에 액세스할 수 없는 경우도 있습니다. • CCSID와 숫자 자료 변환을 실행하지 못합니다.
시스템간 통신 기능/공통 프로그래밍 인터페이스 (ICF/CPI 통신)	ICF는 프로그램이 다른 프로그램이나 시스템과 대화식으로 통신할 수 있도록 해주는 오퍼레이팅 시스템의 한 기능입니다. CPI 통신은 프로그램간 통신을 사용하는 어플리케이션에 일관된 어플리케이션 인터페이스를 제공하는 호출 레벨 인터페이스입니다. 인터페이스는 SNA 논리 장치(LU) 6.2 구조를 사용하여 리모트 시스템의 프로그램과 대화를 설정하고, 자료를 송수신하고, 제어 정보를 교환하고, 대화를 종료하고, 상대 프로그램에 오류를 통지합니다.	<ul style="list-style-type: none"> • 사용자의 요구를 만족시키기 위해 어플리케이션을 사용자 정의할 수 있습니다. • 나은 성능을 제공할 수 있습니다. 	분산 관계형 데이터베이스와 DDM과 비교해 볼 때 통신과 자료 변환 요구사항을 지원하려면 보다 복잡한 프로그램이 필요합니다.
표시장치 pass-through	사용자가 다른 iSeries 서버에서 한 iSeries 서버를 사인 온하고 서버의 프로그램과 자료를 사용할 수 있는 통신 기능.	<ul style="list-style-type: none"> • 리모트 시스템의 어플리케이션과 자료는 로컬 시스템에서 액세스할 수 있습니다. • 자료가 휘발성이고 여러 서버의 사용자가 한 서버에서 많은 양의 자료를 필요로 할 때 빨리 액세스할 수 있습니다. 	화면 갱신의 응답 시간이 로컬로 접속된 장치보다 느립니다.

분산 관계형 데이터베이스는 업무에 변화가 필요하고 새 제품이 생산되면서 항상 간단한 것에서 복잡한 것으로 전개됩니다. 사용자의 요구와 예상을 분석할 때 다음 사항을 고려하십시오.

분산 관계형 데이터베이스에 대한 어플리케이션, 네트워크 및 자료의 설계

분산 관계형 데이터베이스 설계에는 다음과 같은 선택사항이 들어 있습니다.

- 어플리케이션
- 네트워크 고려사항
- 자료 고려사항

추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계

분산 관계형 데이터베이스는 로컬 데이터베이스에서 전용으로 사용하기 위해 개발한 어플리케이션과는 다른 요구사항을 갖고 있습니다. 이런 차이점을 적절히 계획하려면 어플리케이션을 설계할 때 다음 사항을 기억하십시오.

- 가능하면 분산 작업 단위(DUW)를 사용하십시오.

주: OS/400 버전 5 릴리스 1 이전에는 2단계 확약 지원이 iSeries 서버의 TCP/IP에서는 사용할 수 없었습니다.


- 공통 인터페이스를 사용하는 프로그램을 코딩하십시오.
- 복잡한 어플리케이션을 작은 부분으로 분할하고 각 어플리케이션 부분을 처리하기에 가장 적합한 위치에 두십시오. 어플리케이션에서 처리를 분산하는 좋은 방법은 SQL CALL 명령문을 사용하여 처리할 자료가 있는 리모트 위치에서 저장 프로시저를 실행하는 것입니다. 저장 프로시저는 iSeries용 DB2 Universal Database 어플리케이션 서버에서 실행될 때 SQL 작업에만 제한되는 것이 아니라 통합 데이터베이스 입/출력을 사용하거나 다른 유형의 처리를 실행합니다.
- 초기 데이터베이스 어플리케이션을 준비, 테스트 및 사용하는 방법을 조사하십시오.
- 가능하면 SQL 세트 처리 기능을 사용하십시오. 이렇게 하면 어플리케이션 서버와의 통신이 최소화됩니다. 예를 들면, 가능하면 하나의 SQL문을 사용하여 여러 열을 갱신하십시오.
- 작업 단위 내의 데이터베이스 갱신은 프로그램이 준비되었을 때 RUW 연결 메소드가 사용되거나 분산 어플리케이션의 다른 노드가 DUW를 지원하지 않을 경우, 단일 사이트에서 실행되어야 합니다.
- DUW 연결 메소드가 단일 명령문을 하나 이상의 관계형 데이터베이스로 방향 지정하는 것을 제한해야 합니다.
- 성능은 연결 관리 메소드의 선택에 의해 영향을 받습니다. RUW 연결 관리 메소드의 사용은 여러 다른 리모트 관계형 데이터베이스를 앞 뒤로 전환할 필요가 없을 때 선호됩니다. 왜냐하면 DUW 연결 관리에 사용되는 2단계 확약 프로토콜에 많은 오버헤드가 관련되기 때문입니다.

그러나, 여러 리모트 데이터베이스 관리 시스템 사이를 빈번히 전환해야 할 경우에는 DUW 연결 관리를 사용하십시오. DUW 연결 관리를 실행할 때 한 데이터베이스 관리 시스템의 통신 대화는 사용자가 다른 데이터베이스 관리 시스템으로 연결을 전환할 때 종료할 필요가 없습니다. 유사 환경에서는 유사 환경의 대화가 다플트 DDMCNV(*KEEP) 작업 정의 속성의 사용으로 활동을 유지할 수 있기 때문에 상이한 환경에서 만큼 큰 요소는 아닙니다. 그러나, 유사 환경에서도 새 연결을 설정하기 위해 커서를 닫고 통신 흐름을 송신하는 비용을 줄이도록 DUW를 사용하여 성능을 향상시킬 수 있습니다.

- 연결 관리 메소드는 CONNECT 명령문의 의미를 판별합니다. RUW 연결 관리 메소드를 사용하여 CONNECT 명령문은 관계형 데이터베이스의 새 연결을 설정하기 전에 기존 연결을 종료합니다. DUW 연결 관리 메소드로 CONNECT 명령문은 기존 연결을 종료하지 않습니다.

분산 관계형 데이터베이스에 대한 네트워크 고려사항

네트워크 설계는 분산 관계형 데이터베이스의 성능에 직접 영향을 줍니다. 특정 네트워크로 잘 작동하는 분산 관계형 데이터베이스를 적절히 설계하려면 다음을 실행하십시오.

- 회선 속도는 어플리케이션 성능에 매우 중요하므로 기본 분산 관계형 데이터베이스 어플리케이션이 효율적인 성능을 달성하려면 네트워크의 적절한 위치에 충분한 용량을 제공하십시오. 자세한 정보는 V5R1 보충 매뉴얼 웹 사이트의 *Communications Management*  매뉴얼을 참조하십시오.
- 필요한 경우, 사용자의 능력을 갱신하는 데 사용될 수 있는 통신 하드웨어와 소프트웨어를 평가하십시오.
- APPC 연결의 경우, 네트워크를 정의할 때 지정한 세션 한계와 대화 한계를 고려하십시오.
- 필요한 하드웨어, 소프트웨어 및 통신 장비(테스트와 생산 환경 모두)와 분산 관계형 데이터베이스 네트워크에 대한 장비의 가장 적합한 구성을 식별하십시오.
- APPC를 지원하는 데 필요한 기술과는 반대로 TCP/IP를 지원하는 데 필요한 기술을 고려하십시오.
- 일반 사용자 그룹과의 초기 서비스 레벨 계약(예: 주어진 분산 관계형 데이터베이스 어플리케이션에 예상되는 응답 시간)과 제공되는 실제 서비스를 모니터링하고 조정하는 전략을 고려하십시오.
- 현재 스프레드를 위한 보조 기억장치 풀(ASP) 그룹으로 설정되어 있는 AR로부터 데이터베이스에 연결하는데 APPC로 보호되는 DUW 대화를 사용할 수 없다는 점을 명심하십시오.
- 분산 관계형 데이터베이스의 데이터베이스 오브젝트와 분산 관계형 데이터베이스의 각 위치에 대한 명명 전략을 개발하십시오. 위치는 분산 관계형 데이터베이스에 참여하고 있는 관계형 데이터베이스 관리 시스템의 상호 연결된 네트워크에 있는 특정 관계형 데이터베이스 관리 시스템입니다. 여기에서 '위치'는 독립형 ASP 그룹으로 구성된 시스템의 사용자 데이터베이스가 될 수도 있습니다. 이 전략을 개발할 때 다음을 고려하십시오.
 - 분산 데이터베이스에 있는 오브젝트의 완전히 규정된 이름은(두 부분이 아닌) 세 부분으로 되어 있으며 최고 레벨의 규정자는 오브젝트의 위치를 식별합니다.
 - 분산 관계형 데이터베이스의 각 위치는 고유한 ID를 제공해야 하고 데이터베이스의 각 오브젝트는 고유한 ID를 갖고 있어야 합니다. 중복 ID는 심각한 문제를 일으킬 수 있습니다. 예를 들어, 중복 위치와 오브젝트명은 어플리케이션을 의도하지 않은 리모트 데이터베이스에 연결시키고 일단 연결하면 의도하지 않은 오브젝트에 연결할 수 있습니다. 네트워크를 연결할 때는 명명에 특별한 주의를 기울이십시오.
 - 사용자 데이터베이스의 각 위치는 고유한 ID를 제공해야 합니다. 다른 두 서버에 있는 사용자 데이터베이스가 'PAYROLL'이라고 이름 붙여지는 경우, 어플리케이션이 같은 서버에서 두 가지 모두를 액세스할 필요가 있을 때 명명 상의충돌이 발생할 수 있습니다. 독립형 ASP 장치가 구성될 때, 사용자는 그 장치에 대해 ASP 장치의 이름과는 다른 RDB 이름을 지정하는 옵션이 있다는 점을 명심하십시오. 이는 ASP 그룹에서 1차 장치와 연관된 RDB 이름이며 이를 통해 사용자 데이터베이스를 알 수 있습니다.

분산 관계형 데이터베이스에 대한 자료 고려사항

필요한 어플리케이션에 관한 자료 위치는 분산 관계형 데이터베이스를 설계할 때 중요한 고려사항입니다. 배치 결정을 할 때 다음을 고려하십시오.

- 어플리케이션에서 필요한 성능 레벨
- 위치에 따른 자료의 보안, 통화, 일관성 및 가용성에 대한 요구사항
- 필요한 자료의 양과 자료 액세스의 예상 패턴
- 필요한 분산 관계형 데이터베이스 기능을 사용할 수 있는 경우
- 서버를 지원하는 데 필요한 기술과 실제로 사용할 수 있는 기술
- 자료를 "소유"하고 있는 사용자(즉, 자료의 정확도를 유지보수하는 담당자)
- 공통 시스템 보안, 계정, 모니터링과 조정, 문제 처리, 자료 백업 및 회복, 변경 제어에 대한 관리 전략
- 네트워크의 자료 위치와 자료의 단일 또는 복수 복사본의 유지보수 여부와 같은 분산 데이터베이스 설계 결정

분산 관계형 데이터베이스에 대한 관리 전략 개발

이 섹션에서는 분산 관계형 데이터베이스를 관리하는 전략을 설명합니다.

- 분산 관계형 데이터베이스에 대한 일반 조작
- 분산 관계형 데이터베이스에 대한 보안 고려사항
- 분산 관계형 데이터베이스 계정
- 분산 관계형 데이터베이스에 대한 문제 분석
- 분산 관계형 데이터베이스 백업 및 회복

분산 관계형 데이터베이스에 대한 일반 조작

분산 관계형 데이터베이스의 일반 작업을 계획하려면 성능과 가용성을 모두 고려하십시오.

다음 설계 고려사항은 분산 관계형 데이터베이스의 성능과 가용성을 모두 개선하는 데 도움을 줄 수 있습니다.

- 단일 어플리케이션이 자주 실행하거나 다량의 자료를 송신 또는 수신하는 트랜잭션을 동반할 경우, 반드시 트랜잭션을 자료와 같은 위치에 보유하도록 노력하십시오.
- 다른 위치에 있는 어플리케이션이 공유해야 하는 자료의 경우, 가장 활동이 많은 위치에 자료를 기록하십시오.
- 한 위치의 어플리케이션이 다른 위치의 어플리케이션이 필요로 하는 만큼의 자료를 필요로 하면 두 위치에 모두 자료의 복사본을 보유하십시오. 여러 위치에 복사본을 둘 때 관리 전략에 대한 다음 질문을 해 보십시오.
 - 사용자가 복사본 갱신을 허용할 것인가?
 - 복사본을 최신 자료로 새로 고치는 방법과 시기는?
 - 모든 복사본을 백업할 것인가, 한 복사본만 백업하면 충분한가?
 - 모든 복사본에 대해 일관성 있게 일반 관리를 실행하는 방법은?

- 복사본 중 하나를 삭제해도 되는 때는?
- 분산 데이터베이스를 중앙 위치에서 관리할지 각 데이터베이스 위치에서 관리할지의 여부를 고려하십시오.

성능은 다음을 실행하여 개선될 수 있습니다.

- 자료와 어플리케이션을 다른 위치에 두어야 하는 경우, 허용 가능한 한계 내에서 성능을 유지하려면 다음을 실행하십시오.
 - 어플리케이션이 사용할 자료 열만 검색하여 네트워크의 자료 통신량을 가능한 낮게 유지하십시오. 즉, SELECT 명령문의 부분으로 열 이름 리스트 자리에 * 사용을 삼가하십시오.
 - 프로그래머는 많은 양의 자료를 송신하거나 리모트 위치에서 수신하는 코딩 명령문 사용을 자제하도록 하십시오. 즉, SELECT 명령문의 WHERE절의 사용을 자제하여 자료 행의 수를 제한하십시오.
 - 참조 무결성, 트리거 및 저장 프로시저(리모트 관계형 데이터베이스 관리 시스템에 CONNECT한 후에 SQL CALL 명령문)를 사용하면 어플리케이션 서버(AS)에 대한 처리가 분산되어 성능이 향상되고 회선 사용량이 줄어듭니다.
 - FOR FETCH ONLY절을 지정하여 읽기 전용 조회를 사용하십시오.
 - 조회 블록에 대한 규칙을 확인하십시오. 예를 들어, iSeries-iSeries 조회에서 읽기 전용 자료의 블록은 COMMIT(*NONE)에 대해서만 또는 ALWBLK(*ALLREAD)가 지정되었을 때는 COMMIT(*CHG)와 COMMIT(*CS)에 대해 실행됩니다.
 - 가능하면 리모트 자료의 위치에 있는 로컬 자료를 사용하여 리모트 자료에 대한 액세스 수를 낮게 유지 하십시오.
 - SQL 세트 작업을 사용하여 단일 SQL 요구로 어플리케이션 리퀘스터에서 복수 행을 처리하십시오.
 - RUW 연결 관리를 실행할 때 DDMCNV(*KEEP)를 실행하거나 DUW 연결 관리를 실행하여 연결을 해제하지 않도록 하십시오.
- 다음을 실행하여 충분한 네트워크 용량을 제공하십시오.
 - 고속, 고 대역폭의 회선을 설치하거나 네트워크의 적절한 점에 회선을 추가하여 네트워크 용량을 증가시 키십시오.
 - 경합을 줄이거나 특정 프로세서에서의 경합 균형을 개선하십시오. 예를 들면, 기존 어플리케이션을 호스 트 서버에서 부서 서버로 이동시키거나 일부 분산 관계형 데이터베이스 작업을 일괄처리로 분류하십시오.
- 보기 좋은 표 설계를 하십시오. 분산 관계형 데이터베이스 위치에서 1차 키, 표 색인 및 정상화 기술을 적절 히 사용하십시오.
- WHERE절에 사용되는 호스트 변수의 자료 유형이 관련된 키 열 자료 유형의 자료 유형과 일치하는지 확인 하십시오. 예를 들어, 부동 소수점 호스트 변수는 다른 자료 유형의 열에 빌드된 색인의 사용을 규정하지 않는 것으로 알려져 있습니다.

가용성은 다음을 실행하여 개선될 수 있습니다.

- 일반적으로 네트워크의 자료 통신량을 제한하십시오.
- 자료와 어플리케이션을 다른 위치에 두어야 하는 경우, 허용 가능한 한계 내에서 가용성을 유지하려면 다음 을 실행하십시오.

- 대체 네트워크 라우트를 설정하십시오.
- 가용성에 대한 시간 존 차이점의 효과를 고려하십시오.
 - 서버를 시작하는 데 규정된 사용자를 사용할 수 있는가?
 - 근무 시간외 일괄처리 작업이 처리를 방해하는가?
- 좋은 백업과 회복 피처를 확인하십시오.
- 백업과 회복 기술을 갖는 사용자인지 확인하십시오.

분산 관계형 데이터베이스에 대한 보안 고려사항

분산 관계형 데이터베이스 계획에 분산 자료의 보안에 대해 실행해야 하는 결정 사항이 일부 포함되어 있습니다. 이 결정에는 다음이 포함됩니다.

- 다른 위치의 사용자에게 액세스를 허용할 시스템과 이 시스템에 액세스할 다른 시스템의 사용자.
- 이 시스템에 대한 액세스를 어느 정도 제어해야 하는가. 예를 들어, 리모트 사용자가 대화를 시작할 때 사용자 암호를 요구해야 하는가?
- 암호화 형식의 전선에 암호가 흐르도록 해야 하는가?
- 사용자 프로파일이 사용자가 연결하고 있는 관계형 데이터베이스의 이름에 근거한 다른 사용자 ID와 암호에 클라이언트 작업이 맵핑되도록 해야 하는가?
- 다른 위치의 사용자에게 액세스를 허용할 자료와 이 자료에 액세스할 다른 위치의 사용자.
- 사용자가 자료에 대해 취할 수 있도록 허용된 조치.
- 자료에 대한 권한을 중앙에서 제어할 것인지, 로컬에서 제어할 것인지의 여부.
- 여러 시스템이 연결되어 있기 때문에 특별한 주의사항을 취해야 하는지의 여부. 예를 들면, 이름 변환을 사용해야 하는가?

이전 결정을 할 경우, 위치 선택을 할 때 다음을 고려하십시오.

- 실제 보호. 예를 들어, 위치는 제한된 액세스의 여지를 제공할 수 있습니다.
- 시스템 보안 레벨. 시스템 보안 레벨은 종종 위치에 따라 다릅니다. 분산 데이터베이스의 보안 레벨은 네트워크에서 사용되는 가장 낮은 레벨보다 크지 않습니다.

APPC로 연결된 모든 서버는 다음을 실행할 수 있습니다.

- 두 서버가 모두 iSeries 서버이면 암호화 형식으로 암호를 통신합니다.
- 한 서버가 네트워크에 있는 다른 서버와 통신하기 위한 요구를 수신하고, 요구하는 서버가 실제 "사용자"이며 수신 서버와 통신할 권한을 부여 받았는지 확인합니다.

모든 서버는 다음을 실행할 수 있습니다.

- 로컬 서버에서 리모트 서버로 사용자 ID와 암호를 전달하여 리모트 자료 액세스를 허용하기 전에 확인합니다.
- 표와 뷰와 같은 SQL 오브젝트에 액세스하고 조작하는 권한을 부여하거나 취소합니다.

iSeries 서버는 사용자가 자료에 액세스하기 위한 권한이 없는 시도를 추적하거나, 보안에 관련된 다른 이벤트를 추적하는 보안 감사 기능을 갖고 있습니다. 또한 서버는 리모트 서버에서 모든 분산 데이터베이스에 액세스하는 것을 방지할 수 있는 기능을 제공합니다.

- 보안 관련 비용. 보안 비용을 고려할 때 보안 관련 제품 구입 비용과 다음 활동을 실행하는 정보 직원의 시간 단가를 고려하십시오.
- 로컬과 리모트 서버에서 리모트 자료에 액세스하는 사용자의 시스템 ID를 유지보수하십시오.
- 사이트간의 감사 기능을 조정하십시오.

보안에 관한 자세한 정보는 iSeries 분산 관계형 데이터베이스를 위한 보안을 참조하십시오.

분산 관계형 데이터베이스 계정

분산 자료 사용에 대한 통계와 비용을 계산할 수 있어야 합니다. 다음을 고려하십시오.

- 분산 자료 사용의 계정에는 하나 이상의 리모트 서버에 있는 자원 사용, 로컬 서버에 있는 자원 사용 및 서버를 연결하는 네트워크 자원 사용이 있습니다.
- 계정 정보는 각 서버에 의해 독립적으로 누적됩니다. 네트워크 계정 정보는 서버가 누적한 자료와는 독립적으로 누적됩니다.
- 다양한 서버의 시간 존은 계정 정보를 연관시킬 때 고려해야 합니다. 각 서버 시계는 리모트 서버 시계와 동기화되지 않을 수 있습니다.
- 각 서버의 허용된 계정 코드(번호) 사이에 차이가 있을 수 있습니다. 예를 들면, iSeries 서버는 계정 코드를 최대 15자로 제한합니다.

다음 기능은 분산 자료 사용의 통계를 내는 데 사용될 수 있습니다.

- iSeries 서버 작업 사용 계정 저널. iSeries 서버는 작업 사용 계정 정보를 각 분산 관계형 데이터베이스 어플리케이션의 작업 사용 계정 저널에 기록합니다. 누적된 저널 항목을 데이터베이스 파일에 기록하는 데 저널 표시(DSPJRN) 명령을 사용할 수 있습니다. 그런 다음, 사용자 작성 프로그램이나 조회 기능을 사용하여 계정 자료를 분석할 수 있습니다. 자세한 정보는 112 페이지의 『분산 관계형 데이터베이스에서 작업 계정』을 참조하십시오.
- NetView* 계정 자료. 네트워크 자원 사용에 대한 계정 자료를 기록하는 데 NetView® 사용권 프로그램을 사용할 수 있습니다.

분산 관계형 데이터베이스에 대한 문제 분석

문제점 분석은 분산 데이터베이스 환경을 관리하는 데 필요합니다. 문제점 분석에는 서버의 네트워크에서 처리되는 어플리케이션에 대한 문제점의 식별 및 해결 모두 포함됩니다. 다음을 고려하십시오.

- 분산 데이터베이스 처리 문제는 다양한 방법으로 명시됩니다. 예를 들면, 문제점을 감지한 서버가 분산 데이터베이스 어플리케이션에 오류 리턴 코드를 전달할 수 있습니다. 또한, 응답은 느리거나, 잘못되거나 존재하지 않을 수 있습니다.
- 분산 데이터베이스 처리 문제를 진단하는 데 툴을 사용할 수 있습니다. 예를 들어, 각 분산 관계형 데이터베이스 제품은 분산 자료 처리 문제를 진단하는 데 도움을 주는 추적 기능을 제공합니다.

- iSeries 서버가 서버 고장을 감지하면 서버는 고장을 감지한 후에 즉시 프로그램 상태에 대한 정보를 기록합니다.

주: IBM 프로그램에 대한 수정이 필요하고 네트워크에 네트워크 분배 관리자(NDM)가 설치된 System/390* 이 있으면, NDM과 분배 시스템 노드 제품을 사용하여 네트워크의 해당 서버에 대한 갱신이나 대체를 수신하고 전송할 수 있습니다.

분산 관계형 데이터베이스 백업 및 회복

단일 서버 환경에서 백업과 회복은 로컬로 발생합니다. 그러나 분산 데이터베이스에서는 백업과 회복도 리모트 위치에 영향을 줍니다.

iSeries 서버를 사용하여 개별 표나 컬렉션 그룹을 백업하고 회복시킬 수 있습니다. 백업과 회복은 로컬로만 실행될 수 있지만 적절한 백업을 지원하지 않는 서버에 덜 중요한 자료를 두기로 계획할 수 있습니다. 백업과 회복 절차는 하나 이상의 어플리케이션 서버에 있는 자료와 일치해야 합니다. 네트워크에는 하나 이상의 서버가 있기 때문에 어떤 형식으로 네트워크에서 항상 사용할 수 있도록 자료를 두 번째 서버에 저장할 수 있습니다. 이와 같은 전략은 데이터베이스를 네트워크에 분산하기 전에 특별히 계획하고 구성해야 합니다.

제 3 장 iSeries 분산 관계형 데이터베이스의 통신

이 장에서는 다음의 분산 관계형 데이터베이스(RDB) 지원되는 통신 주제를 다룹니다.

- DRDA 구현을 위한 통신 틀은 통신 유형과 회선을 비롯한 분산 관계형 데이터베이스를 지원하는데 사용되는 다양한 통신 틀을 다룹니다.
- 분산 관계형 데이터베이스 통신 네트워크 고려사항은 사용자가 데이터베이스 처리를 분산 관계형 데이터베이스에 의존할 때 통신 네트워크에 관해 고려해야 할 사항입니다.
- 분산 관계형 데이터베이스(RDB)용 통신 구성은 네트워크 구성 단계를 제공합니다.

이 안내서에는 필요한 정보가 모두 들어 있지는 않습니다. 이 안내서의 목적은 사용자가 옳은 질문을 하고 스스로 응답할 수 있도록 도와주며, 업무의 필요에 따라 자원을 최대한 활용하도록 하는 데 있습니다.

DRDA 구현용 통신 틀

iSeries에서 DRDA 실행을 위한 통신 지원은 APPN(확장 대등 시스템간 네트워크[®])의 유무에 관계없이 APPC(확장 프로그램간 통신) 프로토콜을 통해 IBM SNA(시스템 네트워크 구조)하에서만 제공됩니다.

일단계 확약을 갖춘 시스템 TCP/IP DRDA 지원은 V4R2 이후로 사용 가능하게 되었습니다. V5R1에서 이 단계 확약을 갖춘 완전한 DUW 지원이 사용 가능하게 되었습니다.

분산 관계형 데이터베이스의 시스템 네트워크 구조: SNA는 여러 개의 논리 장치(LU) 유형으로 이루어진 구조입니다. 이 논리 장치는 같은 LU 유형을 지원하는 서버, 제어기 및 단말기와 통신하는 방법에 관한 구조적인 정의입니다. iSeries 서버의 분산 관계형 데이터베이스에 필요한 모든 SNA 지원은 OS/400 사용권 프로그램의 부분입니다.

기타 통신 도구에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 분산 관계형 데이터베이스용 APPC/APPN
- DDM 및 분산 관계형 데이터베이스 사용

이 장에 있는 예제와 스펙은 SMA 구성과 기본 TCP/IP에만 특정된 것입니다. TCP/IP를 통한 APPC의 자세한

정보는 V5R1 보충 매뉴얼 웹 사이트에 있는 *Communications Configuration*  매뉴얼을 참조하십시오. 시스템 TCP/IP 지원 설정에 대한 자세한 정보는 TCP/IP 설치 주제를 참조하십시오.

분산 관계형 데이터베이스용 APPC/APPN

APPC는 iSeries 서버가 실행하는 SNA LU 6.2 및 물리 장치(PU) T2.1 구조입니다 이는 다른 프로세서에 있는 어플리케이션이 대등한 관계에서 통신하며 자료를 교환할 수 있게 합니다.

APPN 지원은 다음과 같은 네트워크 기능을 제공하는 PU T2.1 구조를 개선합니다.

- 분산 디렉토리를 탐색하여 네트워크의 LU를 동적으로 위치지정
- 어플리케이션이 세션을 요청할 때 선택 특성을 기초로 하는 LU에 라우트를 동적으로 선택
- 다른 LU 6.2 상대간에 세션의 노드를 통한 LU 6.2 세션 통신망의 중간 라우팅
- 전송 우선순위에 기초한 세션 자료의 라우팅
- 리모트 위치 상대 정의를 동적으로 작성하고 시작
- 고성능 라우팅(HPR)은 APPN 구조에 추가된 것으로 특히 고속 링크를 사용할 때 APPN 라우팅 성능과 신뢰도를 향상

또한, APPC와 APPN은 IBM이 공급하는 기능을 지원합니다.

- SNADS(SNA 분배 서비스)
- iSeries 서버의 표시장치 pass-through

iSeries APPN 및 HPR은 APPC, APPN 및 HPR 주제에 문서화되어 있습니다.

DDM 및 분산 관계형 데이터베이스(RDB) 사용

iSeries 서버에서 DRDA를 실행할 때 분산 자료 관리(DDM) 구조 명령을 사용하여 다른 서버와 통신합니다. 그러나 분산 관계형 데이터베이스(RDB) 및 DDM 파일 액세스 지원으로 일부 기능이 다르게 처리됩니다.

분산 관계형 데이터베이스(RDB) 처리를 사용하면 어플리케이션이 로컬 시스템의 관계형 데이터베이스 디렉토리를 사용하는 리모트에 연결됩니다. 관계형 데이터베이스 디렉토리는 관계형 데이터베이스명과 해당 데이터베이스의 통신 경로 간에 필요한 링크를 제공합니다. 분산 관계형 데이터베이스(RDB)하에서만 실행하는 어플리케이션은 데이터베이스명을 식별하고 처리에 필요한 SQL문을 실행해야 합니다.

DDM 지원을 사용하면 리모트 파일이 식별되고 로컬 시스템의 DDM 파일에 의해 통신 경로가 제공됩니다. V5R2에서 DDM 파일도 RDB 디렉토리 항목에 대한 참조로 작성됩니다.

DDM을 사용하여 리모트 명령 제출, 파일 복사 및 시스템간 자료 이동 등과 같이 관리 작업을 위한 분산 관계형 데이터베이스(RDB) 처리를 지원할 수 있습니다. DDM 지원을 사용하려면 DDM 파일을 작성해야 합니다. 이는 85 페이지의 『DDM 파일 설정』에서 설명됩니다.

iSeries 서버 파일 복사명령으로 DDM 파일을 사용하는 방법은 91 페이지의 『파일 복사 명령을 사용한 iSeries 서버 간의 자료 이동』에서 설명됩니다.

사용자가 IP 네트워크를 갖고 있는 경우, 이 섹션에서 다루고 있는 일부 DDM 관련사항에 사용할 수 있는 유사한 기능이 있습니다. 예를 들어, 사용자는 FTP와 리모트 명령 실행(RUNRMTCMD) 명령을 사용할 수 있습니다.

분산 관계형 데이터베이스 통신 네트워크 고려사항

통신 사용은 분산 관계형 데이터베이스(RDB)에서 증가하고 있으며, 사용자가 데이터베이스 처리를 통신 네트워크에 의존할 때 고려해야 할 몇 가지 사항이 있습니다.

사용자는 분산 관계형 데이터베이스(RDB) 처리에 따르는 사용 증가로 로컬 및 리모트 위치에 작성된 MODE 설명의 최대 세션수(MAXSSN) 매개변수와 최대 대화수(MAXCNU) 매개변수를 증가시킬 수 있습니다.

또한, MODE 설명을 통한 용량 증가에 더하여 분산 관계형 데이터베이스(RDB) 처리의 네트워크 성능을 개선하기 위해 네트워크 내에 있는 다양한 회선의 속도 증가와 더 나은 품질의 회선 선택을 고려할 수도 있습니다.

사용자의 분산 관계형 데이터베이스(RDB) 네트워크에 대해 또 하나 고려해야 할 문제는 자료 액세스 능력과 가용성 부분입니다. 특정의 데이터베이스는 일별 또는 특수 기업망 조작에서의 중요도에 따라 사용자가 이 데이터베이스에 액세스할 수 있는 방법에 대해 고려해야 할 사항이 더 많아집니다. 이는 필요에 따라 자료의 가용성을 제공할 수 있도록 네트워크를 통한 경로와 대체 경로를 검사한다는 것을 의미합니다. 이 주제에 대한 자세한 정보는 제 7 장 『분산 관계형 데이터베이스에 대한 자료 가용성 및 보호』에서 설명됩니다.

회선 속도와 사용자가 통신 회선을 구성하는 방법은 네트워크 성능에 상당한 영향을 미칠 수 있습니다. 그러나 중요한 점은 회선 속도와 사용 유형에 관련하여 전송 중인 정보의 특성에 대해 몇 가지 질문을 해보는 것입니다. 예를 들면, 다음과 같습니다.

- 이동시켜야 하는 정보의 양은 얼마인가?
- 일반적인 트랜잭션과 일괄처리 어플리케이션의 작업단위는 무엇인가?
- 동시에 그 회선을 사용하는 어플리케이션 프로그램이나 사용자의 수는 어느 정도인가?
- 일반적인 트랜잭션과 대화식 어플리케이션의 작업단위는 무엇이며, 각 트랜잭션에 송, 수신되는 자료의 양은 얼마인가?

분산 관계형 데이터베이스(RDB)용 통신 구성

iSeries 서버의 DRDA 구현을 위한 통신 지원은 분산 자료 관리(DDM) 구조를 기초로 합니다. 이 지원은 IBM 시스템 네트워크 구조(SNA)에서 APPN*(확장 대등 시스템간 네트워크*)과 HPR(고성능 라우팅)이 있거나 없는 APPC(확장 대등 프로그램간 통신)는 물론 원래의 TCP/IP 연결을 모두 포함합니다. 또한 OS/400이 APPC를 제공하므로 AnyNet* 지원을 사용하여 DDM과 분산 관계형 데이터베이스에 액세스합니다. AnyNet®은 TCP/IP의 DRDA 리모트 단위 작업 지원에 필요하지 않지만 TCP/IP의 분산 단위 작업 기능에는 유용할 수 있습니다.

이 기능에 관한 자세한 정보는 다음 주제를 참조하십시오. 또한 이 주제는 기본적인 구성 예를 제공하여 네트워크의 시스템을 구성하는 데 필요한 단계를 설명합니다.

- APPC용 통신 네트워크 구성
- TCP/IP용 통신 네트워크 구성
- OptiConnect를 통한 통신 구성

주: 관계형 데이터베이스를 액세스하는 데 사용할 수 있는 어플리케이션 리퀘스터 드라이버 종료 프로그램 통신에 대한 제한사항은 없습니다. 자세한 정보는 어플리케이션 리퀘스터 드라이버 프로그램 주제를 참조하십시오.

APPC용 통신 네트워크 구성

분산 관계형 데이터베이스(RDB)의 통신 구성은 로컬 및 리모트 시스템이 네트워크에 정의되어야 합니다. 일단 네트워크의 시스템이 정의되면, 사용자는 분산 자료 관리(DDM) 기능이나 SNA 분배 서비스(SNADS)를 사용하여 네트워크 전체에 정보를 분배하고, 표시장치 pass-through를 사용하여 로컬 서버의 워크스테이션에서 어플리케이션 서버(AS)까지 연결하고, 분산 관계형 데이터베이스(RDB) 네트워크에 서버의 관계형 데이터베이스 디렉토리를 설정할 수 있습니다. 관계형 데이터베이스 디렉토리는 통신 구성 값을 분산 관계형 데이터베이스(RDB) 네트워크에 있는 관계형 데이터베이스의 이름과 연관시킵니다. 관계형 데이터베이스 디렉토리의 설정에 대한 내용은 제 5 장 『iSeries 분산 관계형 데이터베이스 설정』을 참조하십시오.

네트워크의 각 iSeries 서버는 서버가 스스로 인식하고 네트워크의 리모트 서버를 식별할 수 있도록 정의되어야 합니다. 네트워크의 서버를 정의하려면 사용자는 다음을 수행해야 합니다.

1. 네트워크 속성을 정의해야 합니다.
2. 필요에 따라 네트워크 인터페이스와 네트워크 서버 설명을 작성해야 합니다.
3. 해당 회선 설명을 작성해야 합니다.
4. 제어기 설명을 작성해야 합니다.
5. APPC 연결의 서비스 클래스 설명을 작성해야 합니다.
6. APPC 연결의 모드 설명을 작성해야 합니다.
7. 장치 설명을 자동 혹은 수동으로 작성해야 합니다.

분산 관계형 데이터베이스에 대한 네트워크 속성 정의

네트워크 속성을 정의하려면 네트워크 속성 변경(CHGNETA) 명령을 사용하십시오. 네트워크 속성에는 로컬 서버명, 디폴트 로컬 위치명, 디폴트 제어점명, 로컬 네트워크 ID 및 네트워크 노드 유형이 있습니다. 기계가 끝 노드이면, 속성에는 이 iSeries 서버가 사용하는 네트워크 서버명도 들어갑니다. 또한, 네트워크 속성은 서버가 고성능 라우팅(HPR)을 사용하는지의 여부를 결정합니다. HPR 네트워크 계획 및 구성에 대한 자세한 정보는 iSeries Information Center의 APPC, APPN 및 HPR 주제를 참조하십시오.

분산 관계형 데이터베이스에 대한 네트워크 인터페이스 설명 정의

네트워크 인터페이스 설명을 작성하십시오. 네트워크 인터페이스를 작성하려면 네트워크 인터페이스 작성(Frame-Relay Network) (CRTNWIFR)을 사용하십시오.

분산 관계형 데이터베이스에 대한 회선 설명 정의

iSeries 서버와 네트워크 간에 사용될 실제의 회선 연결과 자료 링크 프로토콜을 설명하는 회선 설명을 작성하십시오. 다음의 명령을 사용하여 회선 설명을 작성하십시오.

- 회선 설명(이더넷) 작성(CRTLINETH)
- 회선 설명(DDI 네트워크) 작성(CRTLINDDI)
- 회선 설명(프레임 교체 네트워크) 작성(CRTLINFR)
- 회선 설명(SDLC) 작성(CRTLINS DLC)
- 회선 설명(토큰링 네트워크) 작성(CRTLINTRN)

- 회선 설명(무선) 작성(CRTLINWLS)
- 회선 설명(X.25) 작성(CRTLINX25)

분산 관계형 데이터베이스에 대한 제어기 설명 정의

제어기 설명은 네트워크에서 인접한 서버를 설명합니다. APPN 지원의 사용을 제어기 설명을 작성할 때 APPN(*YES)을 지정하여 표시합니다. 다음의 명령을 사용하여 제어기 설명을 작성하십시오.

- 제어기 설명(APPC) 작성(CRTCTLAPPC)
- 제어기 설명(SNA HOST) 작성(CRTCTLHOST)

토큰 링, 이더넷, 무선 또는 DDI 회선 설명의 AUTOCRTCTL 매개변수가 *YES로 설정되어 있으면, 이 때 제어기 설명은 서버가 그 회선으로 세션 시작 요청을 수신할 때 자동으로 작성됩니다.

AnyNet 지원을 지정하기 위해 사용자는 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령의 LINKTYPE 매개 변수에 *ANYNW를 지정합니다.

분산 관계형 데이터베이스에 대한 기타 구성 고려사항

추가 로컬 위치나 APPN의 리모트 위치의 특별한 특성이 요구되는 경우, APPN 위치 리스트를 작성해야 합니다. 하나의 로컬 위치 명은 네트워크 속성에 할당된 제어점 이름입니다. iSeries 서버에 추가 위치가 필요한 경우, APPN 로컬 위치 리스트가 필요합니다. 특별한 리모트 위치 특성은 리모트 위치가 로컬 위치 및 보안 요구사항과 다른 네트워크에 있는지의 여부에 관련된 것입니다. 특별한 리모트 위치 특성이 존재하는 경우, APPN 리모트 위치 리스트가 필요합니다. APPN 위치 리스트를 구성 리스트 작성(CRTCFGL) 명령을 사용하여 작성할 수 있습니다. APPN 구성 리스트 및 보안 요구사항에 대한 자세한 정보는 APPC 네트워크의 DRDA 보안 요소를 참조하십시오.

구성 변환(VRYCFG) 명령이나 구성 상태에 대한 작업(WRKCFG SIS) 명령을 사용하여 통신 설명을 연결변환(작동)할 수 있습니다. 비교환 회선 설명이 연결변환되는 경우, 그 회선에 접속된 해당 제어기와 장치도 연결변환됩니다. WRKCFGSTS 명령은 각 연결의 상태를 공급합니다. 통신 구성 상태 작업에 관한 자세한 정보는 제 6 장 『분산 관계형 데이터베이스(DRDB) 관리 및 조작 태스크』를 참조하십시오.

기본 구성 예의 설명에 더하여 예: 분산 관계형 데이터베이스에 대한 APPN 구성에서 설명하는 spiffy사의 네트워크를 참고하십시오.

주:

1. 제어기 설명은 IBM 네트워크 제어 프로그램과 가상 텔레커뮤니케이션 액세스 메소드(NCP/VTAM*) PU 매크로와 같습니다. 제어기 설명에 들어 있는 정보는 확장 서비스 통신 관리자 상대 LU 프로파일을 참조하십시오.
2. 장치 설명은 NCP/VTAM 논리 장치(LU) 매크로와 같습니다. 장치 설명에 들어 있는 정보는 확장 서비스 통신 관리자 상대 LU 및 LU 프로파일을 참조하십시오.
3. 모드 설명은 NCP/VTAM 모드 표와 같습니다. 모드 설명에 들어 있는 정보는 확장 서비스 통신 관리자 전송 서비스 모드 프로파일 및 초기 세션 한계 프로파일을 참조하십시오.

V5R1 보충 매뉴얼 웹 사이트의 *Communications Configuration* 매뉴얼 및 APPC, APPN 및 HPR 주제에는 네트워킹 지원 구성 및 위치 리스트에 대한 자세한 정보가 들어 있습니다.

예: 분산 관계형 데이터베이스에 대한 APPN 구성

기본 구성 예의 설명에 더하여 다음 예에서 설명하는 spiffy사의 네트워크를 참고하십시오.

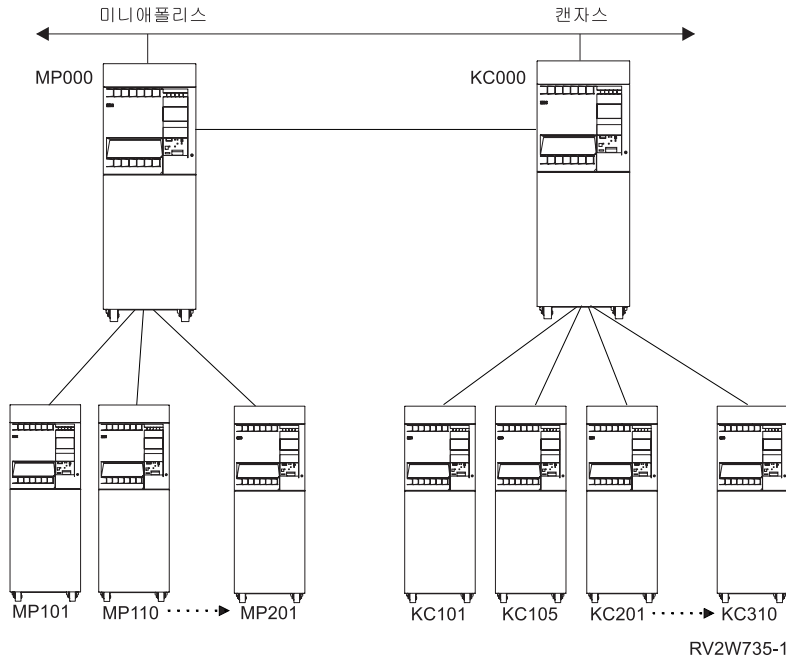


그림 6. spiffy사의 네트워크 조직

이 네트워크 조직에서 spiffy사의 지역 오피스 두 군데는 MP000과 KC000이란 이름의 네트워크 노드 서버입니다. 미국 미니애폴리스에 있는 MP000 서버와 캔자스에 있는 KC000 서버는 백업 회선으로의 SDLC 교환 회선과 함께 SDLC 비교환 회선을 사용하여 통신합니다. MP000 iSeries 서버는 KC000 서버와 다른 지역의 네트워크 노드에 대한 개발과 문제점 처리 센터로 사용됩니다.

다음의 예 프로그램과 설명은 미니애폴리스 및 캔자스의 iSeries 서버를 네트워크에서 네트워크 노드로 구성하는 방법에 대해 설명하며 미니애폴리스가 지역 판매 대리점 중 하나에 자체 네트워크를 구성하는 방법에 대해 나타내고 있습니다. 이 예는 그림 6에 제시된 네트워크를 구성하는 데 필요한 TASK의 부분일 뿐 네트워크에 대한 전체 구성을 설명하는 것은 아닙니다.

네트워크 노드 MP000 구성

다음 예 프로그램은 MP000(네트워크 노드 1)으로 식별된 서버의 구성을 정의하는 데 사용되는 제어 언어(CL)를 나타내고 있습니다. 예는 CL 프로그램 내에서 사용되는 명령을 보여 주고 구성 메뉴를 사용하여 구성을 실행할 수 있습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.


```

/*****
/*
/* MODULE: MP000 LIBRARY: PUBSCFGS */
/*
/* LANGUAGE: CL */
/*
/* FUNCTION: CONFIGURES APPN NETWORK: */
/*
/* THIS IS: MP000 TO KC000 (nonswitched) */
/* MP000 TO KC000 (switched) */
/* MP000 TO MP101 - MP299 (nonswitched) */
/*
/*
/*
/*****
PGM
/* Change network attributes for MP000 */ 1
CHGNETA LCLNETID(APPN) LCLCPNAME(MP000) +
LCLLOCNAME(MP000) NODETYPE(*NETNODE)
/*****
/* MP000 to KC000 (nonswitched) */
/*****
/* Create nonswitched line description for MP000 to KC000*/
CRTLINS DLC LIND(KC000L) RSRCNAME(LIN021) 2
/* Create controller description for MP000 to KC000 */
CRTCTLAPPC CTLD(KC000L) LINKTYPE(*SDLC) + 3
LINE(KC000L) RMTNETID(APPN) +
RMTCPNAME(KC000) STNADR(01) +
NODETYPE(*NETNODE)
/*****
/* MP000 TO KC000 (switched) */
/*****
/* Create switched line description for MP000 to KC000 */
CRTLINS DLC LIND(KC000S) RSRCNAME(LIN022) + 4
CNN(*SWTPP) AUTOANS(*NO) STNADR(01)
/* Create controller description for MP000 to KC000 */
CRTCTLAPPC CTLD(KC000S) LINKTYPE(*SDLC) + 5
SWITCHED(*YES) SWTLINLST(KC000S) +
RMTNETID(APPN) RMTCPNAME(KC000) +
INLCNN(*DIAL) CNNBR(8165551111) +
STNADR(01) TMSGRPNBR(3) NODETYPE(*NETNODE)

/*****

/*****
/* MP000 to MP101 (nonswitched) */
/*****
/* Create nonswitched line description for MP000 to KC000*/
CRTLINS DLC LIND(MP101L) RSRCNAME(LIN031) 6
/* Create controller description for MP000 to MP101 */
CRTCTLAPPC CTLD(MP101L) LINKTYPE(*SDLC) +
LINE(MP101L) RMTNETID(APPN) +
RMTCPNAME(MP101) STNADR(01) +
NODETYPE(*ENDNODE)
/*****
ENDPGM

```

1 네트워크 속성 변경(MP000)

네트워크 속성 변경(CHGNETA) 명령은 네트워크에서 서버의 속성을 설치하는 데 사용됩니다. 다음 속성은 MP000 지역 서버에 정의되고, 이 속성은 네트워크 노드의 네트워크 내 모든 연결에 적용됩니다.

LCLNETID(APPN)

로컬 네트워크명은 APPN입니다. 리모트 서버(예 프로그램의 KC000)는 이 이름을 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령의 리모트 네트워크 ID(RMTNETID)로 지정해야 합니다. 이 예에서, APPN은 네트워크 속성에 디폴트로 되어 있습니다.

LCLCPNAME(MP000)

미니아폴리스 지역 서버의 로컬 제어점에 할당된 이름은 MP000입니다. 리모트 서버는 이 이름을 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령의 리모트 제어점 이름((RMTCPNAME)으로 지정해야 합니다.

LCLLOCNAME(MP000)

디폴트 로컬 위치명은 MP000입니다. 이 이름은 APPN 지원에 의해 작성되는 장치 설명에 사용됩니다.

NODETYPE(*NETNODE)

로컬 서버(MP000)는 APPN 네트워크 노드입니다.

2 회선 설명(MP000에서 KC000, 비교환) 작성

이 예에서 사용된 회선은 SDLC 비교환 회선입니다. 회선을 작성하는 데 사용되는 명령은 회선 설명(SDLC) 작성(CRTLINS DLC)입니다. 지정된 매개변수는 다음과 같습니다.

LIND(KC000L)

회선 설명에 할당된 이름은 KC000L입니다.

RSRCNAME(LIN021)

LIN021이라는 실제 통신 포트가 정의됩니다.

3 제어기 설명(MP000에서 KC000, 비교환) 작성

이는 APPN 환경(iSeries 서버에서 iSeries 서버간)이므로 제어기는 APPC 제어기이고, 제어기의 속성을 정의하는 데 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령을 사용합니다. 다음 속성은 예 명령에 의해 정의됩니다.

CTLD(KC000L)

제어기 설명에 할당된 이름은 KC000L입니다.

LINKTYPE(*SDLC)

이 제어기는 SDLC 통신 회선을 통해 접속되므로 지정된 값은 *SDLC입니다. 이 값은 CRTLINxxx(회선 설명 작성) 명령에 의해 정의된 회선 유형과 일치해야 합니다.

LINE(KC000L)

이 제어기가 접속되어 있는 회선 설명의 이름은 KC000L입니다. 이 값은 회선 설명의 LIND 매개변수에 의해 지정된 이름과 일치해야 합니다.

RMTNETID(APPN)

리모트 제어점이 있는 네트워크의 이름은 APPN입니다.

RMTCPPNAME(KC000)

리모트 제어점명은 KC000입니다. 여기에 지정된 이름은 로컬 제어점 이름에 대해서 리모트 서버에서 지정된 이름과 일치해야 합니다. 예에서, 이름은 네트워크 속성 변경(CHGNETA) 명령의 LCLCPNAME 매개변수에 의해 리모트 서버(KC000)에서 지정됩니다.

STNADR(01)

리모트 제어기에 할당된 주소는 16진 01입니다.

NODETYPE(*NETNODE)

리모트 서버(KC000)는 APPN 네트워크 노드입니다.

4 회선 설명(MP000에서 KC000, 교환) 작성

이 예에서 사용된 회선은 SDLC 교환 회선입니다. 회선을 작성하는 데 사용되는 명령은 회선 설명(SDLC) 작성(CRTLINS DLC)입니다. 지정된 매개변수는 다음과 같습니다.

LIND(KC000S)

회선 설명에 지정된 이름은 KC000S입니다.

RSRCNAME(LIN022)

LIN022로 명명된 실제 통신 포트가 정의됩니다.

CNN(*SWTPP)

이것은 교환 회선 연결입니다.

AUTOANS(*NO)

이 서버는 수신 호출에 대해 자동으로 응답하지 않습니다.

STNADR(01)

로컬 서버에 할당된 주소는 16진수 01입니다.

5 제어기 설명(MP000에서 KC000, 교환) 작성

이는 APPN 환경(iSeries 서버에서 iSeries 서버간)이므로 제어기는 APPC 제어기이고, 제어기의 속성을 정의하는 데 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령을 사용합니다. 다음 속성은 예 명령에 의해 정의됩니다.

CTLD(KC000S)

제어기 설명에 할당된 이름은 KC000S입니다.

LINKTYPE(*SDLC)

이 제어기는 SDLC 통신 회선을 통해 접속되므로 지정된 값은 *SDLC입니다. 이 값은 CRTLINxxx(회선 설명 작성) 명령에 의해 정의된 회선 유형과 일치해야 합니다.

SWITCHED(*YES)

이 제어기는 교환 SDLC 회선에 접속됩니다.

SWTLINLST(KC000S)

이 제어기에 접속될 수 있는 회선 설명(교환 회선용)의 이름은 KC000S입니다. 예에는 단 하나의 회선(KC000S)이 있습니다. 이 값은 회선 설명의 LIND 매개변수에 의해 지정된 이름과 일치해야 합니다.

RMTNETID(APPN)

리모트 제어점이 있는 네트워크의 이름은 APPN입니다.

RMTCPPNAME(KC000)

리모트 제어점명은 KC000입니다. 여기에 지정된 이름은 로컬 제어점 이름에 대해서 리모트 서버에서 지정된 이름과 일치해야 합니다. 예에서, 이름은 네트워크 속성 변경(CHGNETA) 명령의 LCLCPNAME 매개변수에 의해 리모트 서버에서 지정됩니다.

INLCNN(*DIAL)

초기 연결은 수신 호출에 응답하거나 호출을 송신하는 iSeries 서버에 의해 이루어집니다.

CNNBR(8165551111)

리모트 캔자스 시 제어기의 연결(전화) 번호는 8165551111입니다.

STNADR(01)

리모트 캔자스 시 제어기에 할당된 주소는 16진 01입니다.

TMSGRPBR(3)

값(3)은 리모트 서버와 전송 그룹 협상의 APPC 지원에 의해 사용됩니다.

리모트 서버는 전송 그룹에 대해 같은 값을 지정해야 합니다.

NODETYPE(*NETNODE)

리모트 서버(KC000)는 APPN 네트워크 노드입니다.

6 MP101의 회선 및 제어기 작성

예에서 이 부분은 MP000에서 MP101인, 판매 대리점 끝 노드의 회선 및 제어기 구성을 나타내고 있습니다. MP000으로부터 이 각 판매점 끝 노드까지 구성을 비슷하게 해야 합니다. 또한, 미니애폴리스의 구성을 완료하려면 각 판매점에서 이것과 유사한 구성 명령이나 프로그램을 사용하여 자체에서 통신하는 서버의 회선과 제어기를 작성하여야 합니다.

마찬가지로, 30 페이지의 그림 6에 제시된 네트워크 구성을 완료하려면, KC000 서버는 판매 대리점 끝 노드까지 구성해야 하고, 각 끝 노드는 KC000 서버와 통신할 수 있도록 회선과 제어기를 구성해야 합니다.

이 연결은 예에 나와 있지 않습니다.

네트워크 노드 KC000 구성

다음의 예 프로그램은 KC000으로 식별된 지역 서버의 구성을 정의하는데 사용되는 CL 명령을 나타내고 있습니다. 이 예는 CL 프로그램 내에서 사용되는 명령을 보여 주고 구성은 구성 메뉴를 사용하여 실행될 수 있습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

/*****
/*
/* MODULE: KC000 LIBRARY: PUBSCFGS */
/*
/* LANGUAGE: CL */
/*
/* FUNCTION: CONFIGURES APPN NETWORK: */
/*
/* THIS IS: KC000 TO MP000 (nonswitched) */
/* KC000 TO MP000 (switched) */
/*
/*
/*****
PGM
/* Change network attributes for KC000 */
CHGNETA LCLNETID(APPN) LCLCPNAME(KC000) + 7
LCLLOCNAME(KC000) NODETYPE(*NETNODE)
/*****
/* KC000 TO MP000 (nonswitched) */
/*****
/* Create line description for KC000 to MP000 */
CRTLINS DLC LIND(MP000L) RSRNAME(LIN022) 8
/* Create controller description for KC000 to MP000 */
CRTCTLAPPC CTLD(MP000) LINKTYPE(*SDLC) + 9
LINE(MP000L) RMTNETID(APPN) +
RMTCPNAME(MP000) STNADR(01) +
NODETYPE(*NETNODE)
/*****
/* KC000 TO MP000 (switched) */
/*****
/* Create switched line description for KC000 to MP000S */
CRTLINS DLC LIND(MP000S) RSRNAME(LIN031) + 10
CNN(*SWTPP) AUTOANS(*NO) STNADR(01)
/* Create controller description for KC000 to MP000 */
CRTCTLAPPC CTLD(MP000S) LINKTYPE(*SDLC) + 11
SWITCHED(*YES) SWTLINLST(MP000S) +
RMTNETID(APPN) RMTCPNAME(MP000) +
INLCNN(*ANS) CNNNBR(6125551111) +
STNADR(01) TMSGRPNBR(3) NODETYPE(*NETNODE)
ENDPGM

```

7 네트워크 속성 변경(KC000)

네트워크 속성 변경(CHGNETA) 명령은 네트워크에서 서버의 속성을 설치하는 데 사용됩니다. 다음 속성은 KC000으로 명명된 지역 서버에 정의되며, 이 속성은 네트워크 노드에 대한 네트워크 내의 모든 연결에 적용됩니다.

LCLNETID(APPN)

로컬 네트워크명은 APPN입니다. 리모트 서버(이 예에서 미니아폴리스 네트워크 노드)는 이 이름을 제어기 설명(APPC) 작성 (CRTCTLAPPC) 명령의 리모트 네트워크 ID(RMTNETID)로 지정해야 합니다.

LCLCPNAME(KC000)

로컬 제어점으로 할당된 이름은 KC000입니다. 리모트 서버는 이 이름을 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령의 리모트 제어점 이름(RMTCPNAME)으로 지정해야 합니다.

LCLLOCNAME(KC000)

디폴트 로컬 위치명은 KC000입니다. 이 이름은 APPN 지원에 의해 작성되는 장치 설명에 사용됩니다.

NODETYPE(*NETNODE)

로컬 서버(KC000)는 APPN 네트워크 노드입니다.

8 회선 설명(KC000에서 MP000, 비교환) 작성

이 예에서 사용된 회선은 SDLC 비교환 회선입니다. 회선을 작성하는 데 사용되는 명령은 회선 설명(SDLC) 작성(CRTLINS DLC)입니다. 지정된 매개변수는 다음과 같습니다.

LIND(MP000L)

회선 설명에 할당된 이름은 MP000L입니다.

RSRCNAME(LIN022)

LIN022로 명명된 실제 통신 포트가 정의됩니다.

9 제어기 설명(KC000에서 MP000, 비교환) 작성

이는 APPN 환경(iSeries 서버에서 iSeries 서버간)이므로 제어기는 APPC 제어기이고, 제어기의 속성을 정의하는 데 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령을 사용합니다. 다음 속성은 예 명령에 의해 정의됩니다.

CTLID(MP000L)

제어기 설명에 할당된 이름은 MP000입니다.

LINKTYPE(*SDLC)

이 제어기는 SDLC 통신 회선을 통해 접속되므로 지정된 값은 *SDLC입니다. 이 값은 CRTLINxxx(회선 설명 작성) 명령에 의해 정의된 회선 유형과 일치해야 합니다.

LINE(MP000L)

이 제어기가 접속되는 회선 설명의 이름은 MP000L입니다. 이 값은 회선 설명의 LIND 매개변수에 의해 지정된 이름과 일치해야 합니다.

RMTNETID(APPN)

리모트 서버가 있는 네트워크의 이름은 APPN입니다.

RMTCPPNAME(MP000)

리모트 제어점 이름은 MP000입니다. 여기에 지정된 이름은 로컬 제어점 이름에 대해서 리모트 서버에서 지정된 이름과 일치해야 합니다. 예에서, 이름은 네트워크 속성 변경(CHGNETA) 명령의 LCLCPNAME 매개변수에 의해 미니애폴리스 지역 리모트 서버(MP000)에서 지정됩니다.

STNADR(01)

리모트 제어기에 할당된 주소는 16진 01입니다.

NODETYPE(*NETNODE)

리모트 서버(MP000)는 APPN 네트워크 노드입니다.

10 회선 설명(KC000에서 MP000, 교환) 작성

이 예에서 사용된 회선은 SDLC 교환 회선입니다. 회선을 작성하는 데 사용되는 명령은 회선 설명 (SDLC) 작성(CRTLINSDLC)입니다. 지정된 매개변수는 다음과 같습니다.

LIND(MP000S)

회선 설명에 지정된 이름은 MP000S입니다.

RSRCNAME(LIN031)

LIN031로 명명된 실제 통신 포트가 정의됩니다.

CNN(*SWTPP)

이것은 교환 회선 연결입니다.

AUTOANS(*NO)

이 서버는 수신 호출에 대해 자동으로 응답하지 않습니다.

STNADR(01)

로컬 서버에 할당된 주소는 16진수 01입니다.

11 제어기 설명(KC000에서 MP000, 교환) 작성

이는 APPN 환경(iSeries 서버에서 iSeries 서버간)이므로 제어기는 APPC 제어기이고, 제어기의 속성을 정의하는 데 제어기 설명(APPC) 작성(CRTCTLAPPC) 명령을 사용합니다. 다음 속성은 예 명령에 의해 정의됩니다.

CTLD(MP000S)

제어기 설명에 할당된 이름은 MP000S입니다.

LINKTYPE(*SDLC)

이 제어기는 SDLC 통신 회선을 통해 접속되므로 지정된 값은 *SDLC입니다. 이 값은 CRTLINxxx(회선 설명 작성) 명령에 의해 정의된 회선 유형과 일치해야 합니다.

SWITCHED(*YES)

이 제어기는 교환 SDLC 회선에 접속됩니다.

SWTLINLST(MP000S)

이 제어기에 접속될 수 있는 회선 설명(교환 회선용)의 이름은 MP000S입니다. 예에는 단 하나의 회선(MP000)이 있습니다. 이 값은 회선 설명의 LIND 매개변수에 의해 지정된 이름과 일치해야 합니다.

RMTNETID(APPN)

리모트 제어점이 있는 네트워크의 이름은 APPN입니다.

RMTCPNAME(MP000)

리모트 제어점 이름은 MP000입니다. 여기에 지정된 이름은 로컬 제어점 이름에 대해서 리모트 지역 서버에서 지정된 이름과 일치해야 합니다. 예에서, 이름은 네트워크 속성 변경(CHGNETA) 명령의 LCLCPNAME 매개변수에 의해 리모트 미니애폴리스 지역 서버(MP000)에서 지정됩니다.

INLCNN(*ANS)

초기 연결은 수신 호출에 응답하는 iSeries 서버에 의해 이루어집니다.

CNNNBR(6125551111)

리모트 미니애폴리스 제어기의 연결(전화) 번호는 6125551111입니다.

STNADR(01)

리모트 미니애폴리스 제어기에 할당된 주소는 16진 01입니다.

TMSGRPNBR(3)

값(3)은 리모트 서버와 전송 그룹 협상의 APPC 지원에 의해 사용됩니다. 리모트 서버는 전송 그룹에 대해 같은 값을 지정해야 합니다.

NODETYPE(*NETNODE)

리모트 서버(MP000)는 APPN 네트워크 노드입니다.

TCP/IP용 통신 네트워크 구성

사용자는 iSeries Navigator을 사용하여 TCP/IP 네트워크를 빠르고 쉽게 설정할 수 있습니다. 또는, TCP/IP 네트워크를 설정하기 위해 취하는 단계 중에서 높은 수준의 개요를 제공하는 다음과 같은 단계를 사용할 수 있습니다. 세부사항은 iSeries Information Center의 TCP/IP 구성 주제를 참조하십시오.

1. 로컬 네트워크(iSeries 서버가 직접 연결한 네트워크)에 대해 사용자의 iSeries 서버를 식별하십시오.
 - a. 회선 설명이 이미 있는지 판별하십시오.
 - b. 회선 설명이 없으면 하나를 작성하십시오.
 - c. TCP/IP 인터페이스를 정의하여 사용자의 iSeries 서버에 IP 주소를 부여하십시오
2. TCP/IP 경로를 정의하십시오. TCP/IP로 iSeries 서버가 리모트 TCP/IP 네트워크(iSeries 서버가 직접 연결되지 않는 네트워크)에 있는 서버와 통신할 수 있습니다.
3. 로컬 정의역명과 호스트명을 정의하십시오. 사용자 서버에 이름을 할당합니다.
4. 네트워크의 서버명을 식별하십시오.
 - a. 로컬 호스트 표를 빌드하십시오.
 - b. 리모트명 서버를 식별하십시오.
5. TCP/IP를 시작하십시오.
6. TCP/IP가 작동하는지 확인하십시오.

OptiConnect를 통한 통신 구성

세 가지 방법 중 하나를 이용하여 opticonnect 상에서 DRDA를 사용할 수 있습니다.

- RDB 디렉토리 항목의 장치로서 QYCTSOC를 지정하십시오. 분배된 작업 단위는 이 방법으로는 지원되지 않습니다.
- OptiConnect 제어기 및 장치를 설치하고 RDB 디렉토리 항목에서 이를 구성하십시오. 이는 opticonnect를 종료하기 전에 사용자가 제어기를 단절변환해야 한다는 단점이 있습니다.
- opticonnect 상에서 IP를 구성하고 IP 실행을 위해 RDB 디렉토리 항목을 설정하십시오. 작성하여 연결변환 또는 단절변환시킬 제어기가 없습니다. IP 인터페이스는 자동시작하도록 설정될 수 있고 opticonnect가 시작되면 활동 상태가 될 것입니다. 또, LAN 어댑터의 IP 주소를 연관된 로컬 주소로 지정하는 단말 인터

페이스를 설정할 수 있습니다. 이는 자동으로 opticonnect가 올라가면 opticonnect 상에서 통신을 방향지시하고, opticonnect가 내려가면 LAN 상에서 방향지시할 것입니다. OptiConnect 설정에 대한 자세한 정보는

OptiConnect for OS/400  서적을 참조하십시오.

제 4 장 iSeries 분산 관계형 데이터베이스에 대한 보안

iSeries 서버는 어플리케이션 서버의 자료 자원에 대한 액세스를 제한하기 위해 오퍼레이팅 시스템에 보안 요소를 빌드하였습니다. 보안 옵션은 간단한 실제 보안에서 명령과 자료 오브젝트에 대한 승인을 요구하는 암호 보안에 이르기까지 다양합니다. 사용자는 로컬이나 리모트 데이터베이스에 액세스하는 적절한 권한을 부여 받아야 합니다. 또한 어플리케이션 프로그램을 실행하는 데 필요한 컬렉션, 표 및 기타 관계형 데이터베이스 오브젝트에 대한 적절한 권한을 가져야 합니다. 이는 분산 데이터베이스 사용자가 네트워크에서 사용하는 데이터베이스에 대한 유효한 사용자 프로파일을 가져야 한다는 의미입니다. 보안 계획은 네트워크에서 사용자와 어플리케이션 프로그램 요구를 고려해야 합니다.

보안 요소에 대한 자세한 정보는 분산 관계형 데이터베이스(RDB) 보안 요소를 참조하십시오.

분산 관계형 데이터베이스(RDB) 관리자는 해결해야 할 두 가지의 보안 문제점을 갖고 있습니다.

- 시스템간 보호
- 리모트 사이트에서의 사용자 식별

둘 이상의 시스템이 서로의 데이터베이스에 액세스하도록 설정할 때, 통신 회선 상 상대방이 의도한 위치이며 침입자가 아닌지 확인하는 것이 중요합니다. 리모트 관계형 데이터베이스에 대한 DRDA 액세스의 경우, iSeries 서버에서 APPC(advanced program-to-program communications) 및 APPN(Advanced Peer-to-Peer Networking) 통신 구성 기능을 사용할 경우, 이 네트워크 레벨 보안을 실행할 수 있는 옵션이 제공됩니다.

분산 관계형 데이터베이스(RDB) 관리자의 두 번째 관심사는 자료 보안이 자료를 저장하는 시스템에 의해 유지보수되도록 하는 것입니다. 분산 관계형 데이터베이스(RDB)에서, 사용자는 데이터베이스가 로컬이든 리모트 이든(시스템의 보안 레벨에 따라) 데이터베이스에 액세스할 수 있는 올바른 권한을 부여받아야 합니다. 분산 관계형 데이터베이스 네트워크 사용자는 AS에서 실행하는 모든 작업에 대하여 어플리케이션 서버(AS)에서 사용자 ID로 올바르게 식별되어야 합니다. APPC/APPN 및 TCP/IP 통신 프로토콜을 모두 사용하는 DRDA 지원은 연결 요청에 따라 사용자 ID 및 암호 송신을 제공합니다.

이 장에서는 통신에 관련된 보안 주제와 리모트 관계형 데이터베이스에 대한 DRDA 액세스에 대해 논의합니다. APPC 네트워크 연결에서의 대화 레벨 보안과 DRDA 어플리케이션에 의해 개시된 TCP/IP 연결에서의 해당 보안 레벨 사이의 중요한 차이점을 다루고 있습니다. 나머지 보안에 대한 내용에서, 사용자란 용어에는 통신 작업을 시작하는 리모트 사용자도 포함됩니다.

사용자의 필요에 맞게 제시된 보안 전략에 관해서는 분산 관계형 데이터베이스(DRDB)의 보호 전략을 참조하십시오.

일반적인 iSeries 보안 개념에 대한 설명은 iSeries Information Center의 기본 시스템 보안 주제를 참조하십시오.

분산 관계형 데이터베이스(RDB) 보안 요소

분산 관계형 데이터베이스(RDB) 관리자는 네트워크 내에서 어플리케이션 리퀘스터(AR)에 의한 자료 액세스를 불필요할 정도로 제한하지 않으면서 네트워크에 있는 어플리케이션 서버의 자원을 보호해야 합니다.

AR은 오브젝트와 관계형 데이터베이스를 보안하여 권한이 있는 사용자만이 분산 관계형 데이터베이스(RDB) 프로그램에 액세스하도록 보장합니다. 이것은 사용자를 식별하고 각 사용자(또는 사용자 그룹)에게 오브젝트를 다루도록 허용하는 것을 지정하기 위해 일반적인 iSeries 서버 오브젝트 권한을 사용하여 실행됩니다. 마찬가지로, 표, 보기 및 SQL 패키지에 대한 권한은 SQL GRANT 및 REVOKE문을 사용하여 부여되거나 취소될 수 있습니다. AR 도움말에서 SQL 오브젝트에 대한 권한 레벨을 제공하면 권한이 있는 사용자가 다른 시스템의 자료에 액세스하는 SQL 어플리케이션에 액세스하도록 보장합니다.

실제로 어플리케이션 서버(AS)에서 시스템 보안 레벨은 AR의 요청이 수렴되는지와 리모트 사용자가 AS의 오브젝트에 대한 권한이 있는지 여부를 결정합니다.

분산 관계형 데이터베이스(RDB) 네트워크에 있는 iSeries 서버에 대하여 다음과 같은 몇 가지 측면의 보안 계획이 있습니다.

- 기밀의 표, 프로그램 및 패키지와 같이 특정 자원에 대한 사용자 액세스를 제어하기 위한 오브젝트에 관련된 보안
- 네트워크에 있는 다른 시스템의 ID를 확인하는 위치 보안
- 로컬 시스템과 리모트 시스템에서 사용자의 ID와 권리를 확인하기 위한 사용자에게 관련된 보안
- 기밀의 표, 프로그램 및 패키지와 같이 특정 자원에 대한 사용자 액세스를 제어하기 위한 오브젝트에 관련된 보안
- 회선 설명에 구성되고 라우트 선택 프로세스에서 사용될 수 있는 시스템, 모뎀, 통신 회선 및 단말기를 둘러싸고 있는 잠긴 문 또는 보안 빌드와 같은 실제적인 보안

위치, 사용자 관련 및 오브젝트 관련 보안 등은 시스템 보안 레벨이 20 또는 그 이상의 레벨에서 확정되는 경우에만 가능합니다.

APPC 대화에서, 시스템이 보안 레벨 10을 사용하고 있을 때, iSeries 서버는 비보안 시스템으로서 네트워크에 연결됩니다. AS/400은 세션 설치 중에 리모트 시스템의 ID를 확인하지 않으며, 프로그램 시작 요청의 수신에 대해 대화 보안을 요구하지도 않습니다. 레벨 10에 대하여, APPC 리모트 위치를 위해 구성된 보안 정보는 무시되며 세션이나 대화 설치 중에 사용되지도 않습니다. 사용자 프로파일이 서버에 없을 경우에는 사용자 프로파일이 하나 작성됩니다.

시스템이 20 또는 그 이상의 보안 레벨을 사용하고 있을 때, iSeries 서버는 보안 시스템으로서 네트워크에 연결됩니다. iSeries 시스템은 대화 레벨의 보안 기능을 제공하고 APPC의 경우에는 세션 레벨 보안도 제공합니다.

사용자의 네트워크에 속하는 시스템 전체에 같은 레벨로 시스템 보안을 설치함으로써 보안 관리의 타스크가 보다 용이해집니다. AS는 세션 설치를 위해 AR로부터 요구되는 것을 지정하여 세션과 대화가 설치될 수 있는

지 여부를 제어합니다. 예를 들어, AR에 대한 보안 레벨이 10으로 정해져 있고 AS에 대한 보안 레벨이 10 이상이면, 해당하는 정보가 송신되지 않을 수도 있고, 세션은 여러 시스템 중 하나에서 보안 요소를 변경하지 않는 이상 설치되지 않을 수도 있습니다.

DRDA 액세스용 암호

리모트 사용자에게 데이터베이스 액세스 권한을 부여하는 가장 일반적인 방법은 연결 시에 사용자 ID와 암호를 부여하는 것입니다. 이를 위해 어플리케이션 프로그래머가 사용할 수 있는 한 방법은 삽입된 SQL CONNECT문에서 USER/USING절을 코딩하는 것입니다. 예를 들면, 다음과 같습니다.

```
EXEC SQL CONNECT TO :locn USER :userid USING :pw
```

리모트 관계형 데이터베이스로 DRDA 액세스에서 일단 대화가 설치되면, 사용자는 암호를 다시 입력할 필요가 없습니다. RUW 연결 관리 메소드로 실행 중일 때에 RELEASE, DISCONNECT 또는 CONNECT 명령문과 연결을 종료하는 경우, 사용자가 연관되어 있는 AS 유형과 사용자의 어플리케이션 리퀘스터(AR) 작업 속성에 따라 첫 번째 어플리케이션 서버(AS)와 사용자의 대화는 제거될 수도 제거되지 않을 수도 있습니다. (특정의 규칙에 대해서는 DDM 대화 제어를 참조하십시오.) 첫 번째 AS에 연결된 대화가 제거되지 않으면, 이것은 사용자가 두 번째 AS에 연결되는 동안 사용하지 않은 상태로 유지됩니다. 첫 번째 AS에 다시 연결하여 대화가 사용하지 않은 상태이면, 사용자 ID와 암호를 입력할 필요없이 대화가 다시 활동 상태가 됩니다. 대화의 두 번째 사용에서, 사용자의 암호 또한 다시 유효하지 않은 상태가 됩니다.

특정 보안 시스템에 대한 자세한 내용은 다음 주제를 참조하십시오.

- APPC 네트워크의 DRDA 보안 요소
- APPC 네트워크의 DRDA 어플리케이션 서버(AS) 보안
- TCP/IP를 사용한 DDM/DRDA 보안 요소
- DRDA 서버 액세스 제어 종료 프로그램
- DRDA용 오브젝트 관련 보안
- 분산 관계형 데이터베이스 오브젝트에 권한
- 분산 관계형 데이터베이스에 대한 허용 권한 하에서 실행되는 프로그램

보안 레벨에 대한 자세한 정보는 보안 및 iSeries Information Center의 APPC, APPN 및 HPR 보안 고려사항 주제를 참조하십시오.

APPC 네트워크의 DRDA 보안 요소

DRDA가 사용될 때는 DRDA 환경의 각 서버의 자료 자원은 보호되어야 합니다. 이는 다음 매개변수에 의해 제어되는 세 가지 그룹의 보안 요소를 사용하여 이루어집니다.

- 시스템 관련 보안이나 세션에 대해서는 **LOCPWD** 매개변수가 각 iSeries 서버에서 사용되어 APPC 통신 세션이 소스 시스템과 목표 시스템간에 처음 설정될 때, 시스템 유효성 암호가 소스 시스템과 목표 시스템간에 교환되는 것을 나타냅니다. 두 시스템 모두 세션에 시작되기 전에 같은 암호를 교환해야 합니다. (System/36에서 이 암호는 위치 암호라고 불립니다.) APPC 네트워크에서, 장치 설명(APPC) 작성 (CRTDEVAPPC) 명령의 LOCPWD 매개변수는 이 암호를 지정합니다. 장치는 자동으로 APPN을 사용하

여 작성되고 리모트 위치 리스트의 위치 암호는 신원 확인을 위해 두 위치에서 사용되는 암호를 지정합니다. 구성 리스트 작성(CRTCFGL) 명령을 사용하여 리모트 위치 리스트 유형(*APPNRMT)을 작성하십시오.

- 사용자 관련 또는 위치 보안에 대해서는 **SECURELOC** 매개변수가 각 iSeries 서버에서 사용되어 (목표 서버로서)소스 서버에서 이미 보안이 확인된 들어오는 액세스 요구를 받아들일 것인지 또는 사용자 ID와 암호화된 암호를 요구할 것인지를 나타냅니다. APPC 네트워크에서, 장치 설명(APPC) 작성(CRTDEVAPPC) 명령의 SECURELOC 매개변수는 로컬 서버가 리모트 서버에서 보안을 확인하도록 허용 여부를 지정합니다. 장치는 APPN을 사용하여 자동으로 작성되며 APPN 리모트 구성 리스트의 보안 위치를 사용하여 로컬 서버는 리모트 서버가 사용자 보안 정보를 확인할 수 있도록 허용할지를 결정합니다. SECURELOC 값은 각 리모트 위치에 대해 다르게 지정될 수도 있습니다.

SECURELOC 매개변수는 다음 보안 요소와 함께 사용됩니다.

- 이 매개변수에 의해 허용된 경우, 소스 서버가 송신한 사용자 ID
- 이 매개변수에 의해 허용된 경우, 사용자 ID와 암호화된 암호
- 디폴트 사용자 프로파일을 포함한 목표 서버 사용자 프로필

자세한 정보는 APPC 네트워크의 DDM 소스 시스템 보안 주제를 참조하십시오.

- 오브젝트 관련 보안에 대해서는 네트워크 속성 변경(CHGNETA) 명령에서 **DDMACC** 매개변수가 사용되어 iSeries 서버의 파일을 다른 서버가 액세스할 수 있는지와 액세스할 수 있는 경우, 어떤 보안 레벨에서 들어오는 요구를 검사할 것인지를 나타냅니다. 오브젝트 관련 매개변수에 관한 자세한 정보는 iSeries Information Center의 DDM 네트워크 속성(DDMACC 매개변수) 주제에서 제공됩니다.
 - DDMACC 매개변수에 *REJECT를 지정하면, iSeries 서버에 의해 수신된 모든 DRDA 요구가 거부됩니다.
 - DDMACC 매개변수에 *OBJAUT가 지정되면, 목표 서버에서 일반 오브젝트 레벨의 보안을 사용하게 됩니다.
 - 선택형의 사용자 공급 사용자 종료 프로그램(또는 액세스 제어 프로그램)이 DDMACC 매개변수에 지정되면 추가 보안 레벨이 사용됩니다. 사용자 종료 프로그램은 특정 소스 서버의 지정된 사용자가 목표 서버의 특정 파일을 액세스하기(어떤 방식으로든) 위한 특정 명령을 사용할 수 있는지 여부를 제어하는데 사용됩니다(세부사항은 추가 보안을 위한 DDM 서버 액세스 제어 종료 프로그램 주제를 참조하십시오).
 - 파일이 DRDA를 사용하여 목표 서버에 작성될 때, 지정된 라이브러리명이 이 파일을 포함합니다. DRDA 요구 시 아무런 라이브러리명도 지정되지 않은 경우, 현재 라이브러리(*CURLIB)가 사용됩니다. 파일 권한은 디폴트로 해당 파일을 작성한 사용자만 또는 목표 서버의 보안 담당자만 파일에 액세스할 수 있도록 허용합니다.

리모트 파일 액세스를 제한하는 대부분의 보안 제어는 목표 서버가 처리합니다. 소스 서버에서 제공되는 사용자 ID를 제외하면 모든 요소는 목표 서버에서 지정되고 사용됩니다. 하지만 소스 서버도 소스 서버의 DRDA 파일에 액세스하는 것을 제어하고 필요하면 사용자 ID를 목표 서버에 송신하여 목표 서버 파일에 액세스하는 것을 제한합니다.

APPN 구성 리스트

APPC 네트워크에서, 위치 간에 단말(end-to-end) 세션을 갖고 있는 위치 쌍에 대하여 위치 암호를 지정합니다. 위치 암호는 중간 노드인 위치에 대하여 지정되어서는 안됩니다.

리모트 위치 리스트는 구성 리스트 작성(CRTCFGL) 명령으로 작성되며, 이것은 모든 리모트 위치, 리모트 위치 암호 및 리모트 위치의 보안 상태 여부에 대한 리스트를 포함합니다. iSeries 서버에는 전 시스템에 걸친 리모트 위치 구성 리스트가 하나 있습니다. 중심부의 iSeries 서버는 CL(제어 언어) 프로그램을 송신함으로써 리모트 iSeries 서버에 대해 위치 리스트를 작성할 수 있습니다.

구성 리스트 변경(CHGCFGL) 명령을 사용하여 리모트 구성 리스트를 변경할 수 있습니다. 그러나 그 위치의 모든 장치는 단절 전환된 후에만 실행됩니다.

구성 리스트 표시(DSPCFGL) 명령을 사용할 때에는 암호가 존재한다는 표시가 없습니다. 구성 리스트 변경(CHGCFGL) 명령은 암호가 입력된 경우, 필드에 *PASSWORD를 위치지정함으로써 암호의 존재를 표시합니다. 암호를 표시하는 방법이 없습니다. 사용자는 위치 보안을 설정하면서 문제점이 발생하는 경우, 양 시스템에 암호를 다시 입력하여 암호가 맞는지 확인해야 합니다.

구성 리스트에 대한 자세한 정보는 Information Center의 APPC, APPN 및 HPR 주제를 참조하십시오.

대화 레벨 보안

시스템 네트워크 구조(SNA) 논리 장치(LU) 6.2 구조는 SNA 네트워크에 있는 다양한 유형의 시스템이 서로 다른 시스템으로 구성된 네트워크를 통해 일정한 대화 보안을 제공하는 데 사용할 수 있도록 세 가지로 지정된 대화 보안을 식별합니다. SNA 보안 레벨은 다음과 같습니다.

SECURITY(NONE)

사용자 ID나 암호를 송신하여 통신을 설치합니다.

SECURITY(SAME)

로컬 서버와 같은 사용자 ID를 갖고 있는 리모트 서버에 사용자에게 대해 알리십시오.

SECURITY(PGM)

사용자 ID와 암호는 모두 통신용으로 송신됩니다.

SECURITY(PROGRAM_STRONG)

암호가 지우기 상태로 송신되지 않는 경우에만 사용자 ID와 암호가 모두 통신용으로 송신됩니다. 그렇지 않으면 오류가 보고됩니다. OS/400의 DRDA에서는 이를 지원하지 않습니다.

iSeries 서버가 네 가지 SNA 대화 보안 레벨을 모두 지원하는 반면에, DRDA는 오직 처음 세 가지 레벨만을 사용합니다. 목표는 대화에 사용된 SNA 대화 레벨을 제어합니다.

SECURITY(NONE) 레벨에서, 목표는 사용자 ID나 암호를 요구하지 않습니다. 목표에서 디폴트 사용자 프로파일을 사용하여 대화할 수 있습니다. 디폴트 사용자 프로파일이 대화에 사용될 수 있는지는 주어진 시스템에 대한 통신 항목 추가(ADDCNNE) 명령이나 통신 항목 변경(CHGCNNE) 명령의 DFTUSR 매개변수에 지정된 값에 따라 달라집니다. DFTUSE 매개변수에 대한 값 *NONE은 목표에서 어플리케이션 서버(AS)가 디폴

트 사용자 프로파일을 사용하여 대화할 수 없음을 나타냅니다. SECURITY(NONE)는 암호나 사용자 ID를 공급하지 않고 목표에 SECURELOC(*NO)는 지정되어 있을 때 송신됩니다.

SECURITY(SAME) 레벨의 경우, 리모트 서버의 SECURELOC 값은 리모트 서버가 iSeries라고 가정할 때 송신되는 보안 정보는 무엇이든지 제어합니다. SECURELOC 값이 *NONE인 경우, 마치 SECURITY(NONE)가 요구된 것처럼 사용자 ID나 암호가 송신되지 않습니다. SECURITY(NONE)를 처리하는 방법에 대해서는 이전의 단락을 참조하십시오. SECURELOC 값이 *YES인 경우, 암호가 이미 로컬 서버에 의해 확인되었다는 표시와 함께 사용자 프로파일명이 압축되어 송신됩니다. SECURELOC 값이 *VFYENCPWD인 경우, 암호가 이 값을 기밀로 유지하도록 암호화한 후에 사용자 프로파일과 이것에 연관된 암호를 리모트 서버로 송신합니다. 따라서 DRDA를 사용하려면 사용자는 같은 사용자 프로파일명과 암호를 가져야 합니다.

주: 대부분의 정보가 리모트 서버에 의해 확인되기 때문에 SECURELOC(*VFYENCPWD)는 세 가지 옵션 중 가장 안전합니다. 그러나, 이것은 복수 서버에서 사용자가 같은 암호를 유지보수하도록 요구하며, 그 이유는 사용자가 하나의 서버를 변경하면서 동시에 다른 서버를 갱신하지 않을 경우 문제가 생길 수 있기 때문입니다.

SECURITY(PGM) 레벨에서, 목표는 대화를 위해 소스로부터 사용자 ID와 암호를 모두 요구합니다. 대화가 설치될 때 암호가 유효화되는 반면, 해당 대화에서 다음과 같은 사용에 대해서는 암호가 무효화됩니다.

APPC 네트워크의 DRDA 어플리케이션 서버(AS) 보안

목표 서버가 iSeries 서버일 때, 함께 사용된 여러 요소가 리모트 파일에 액세스하려는 요구를 허용할지 여부를 결정합니다.

사용자 관련 보안 요소: 목표 서버의 SECURELOC 매개변수, 소스 서버에서 송신된 사용자 ID(허용된 경우), 소스 서버에서 송신된 사용자 ID에 대한 암호 및 목표 서버의 사용자 프로파일 또는 디폴트 사용자 프로파일.

오브젝트 관련 보안 요소: DDMACC 매개변수 및 정상적인 오브젝트 권한 제어를 보완하기 위해 사용자가 제공하는 사용자 종료 프로그램(선택적).

목표 보안의 사용자 관련 요소

유효한 사용자 프로파일은 분산 관계형 데이터베이스 작업을 처리하기 위해 어플리케이션 서버(AS)에 존재해야 합니다. 사용자는 iSeries 서버에서 통신 작업을 처리하는 서브시스템에 대하여 디폴트 사용자 프로파일을 지정할 수 있습니다. 디폴트 사용자 프로파일명은 AS에서 통신 항목 추가(ADDCMNE) 명령의 DFTUSR 매개변수에 지정됩니다. ADDCMNE 명령은 통신 작업에 사용된 서브시스템 설명에 통신 항목을 추가합니다.

디폴트 사용자 프로파일이 통신 서브시스템에 지정되는 경우, AS가 보안 위치인지 아닌지는 디폴트 사용자 프로파일이 이 요구에 사용되는지 여부를 결정합니다. 장치 설명(APPC) 작성(CRTDEVAPPC) 명령의 SECURELOC 매개변수 또는 APPN 리모트 위치 리스트에 명시된 보안 위치는 AS가 보안 위치인지 아닌지를 지정합니다.

• SECURELOC 또는 AS의 보안 위치에 대해 *YES가 지정되어 있는 경우, AS는 어플리케이션 리퀘스터 (AR) 보안 위치를 고려합니다. 사용자 ID 및 이미 확인된 인디케이터는 요구를 갖고 있는 AR에 필요합니

다. 사용자 프로파일이 리퀘스터에 의해 송신된 사용자 ID와 일치하는 AS에 있는 경우, 해당 요구가 허용됩니다. 사용자 ID와 일치하지 않으면 요구는 거부됩니다.

- *AS의 SECURELOC 매개변수에 대해 *NO가 지정되는 경우, AS는 AR을 보안 위치로 간주하지 않습니다. AR이 여전히 사용자 ID를 송신한다 하더라도 AS는 요구에 대해 이 ID를 사용하지 않습니다. 그 대신에, AS에 사용할 수 있는 디폴트 사용자 프로파일이 있을 경우, 요구로 사용됩니다. 디폴트 사용자 프로파일이 AS에 없을 경우, 요구가 거부됩니다.
- AS에서 SECURELOC에 대해 *VFYENCPWD를 지정하는 경우, AS는 AR을 보안 위치로 간주하지만, 현재의 사용자의 ID를 확인하기 위해 사용자 ID와 이 암호를(암호화 양식으로) 전송하도록 요구합니다. 사용자 프로파일이 리퀘스터가 송신한 사용자 ID와 일치하는 AS에 존재하고, 해당 리퀘스터가 두 시스템에 대해 같은 암호를 갖고 있는 경우에는 요구가 허용됩니다. 그렇지 않으면 요구는 거부됩니다.

표 4에서 iSeries 서버에서 SNA SECURITY(PGM)를 제어하는 요소의 가능한 모든 조합을 나타내고 있습니다. 모든 열에서 『Y』는 요소가 존재하거나 조건이 충족되어 있음을 표시합니다. PWD 열에서 『M』은 보안 관리자가 사용자 암호를 검색하고 암호 보호가 활동 중인 경우 보호(암호화된) 암호를 송신하고 있음을 나타냅니다. 보호 암호가 송신되지 않을 경우, 어떠한 암호도 송신하지 못합니다. 보호 암호는 APPC가 대화를 시작할 때 사용자 암호에 대해 대체하는 문자 스트링입니다. 보호 암호는 양쪽 상대의 시스템이 암호 보호를 지원할 때와 OS/400 버전 2 릴리스 2 또는 후속 기종을 실행하는 시스템에 암호를 작성할 때에만 사용될 수 있습니다.

표 4. 분산 관계형 데이터베이스로 리모트 액세스

행	UID	PWD ¹	AVI	SEC(Y)	DFT	유효	액세스
1	Y	Y		Y	Y	Y	UID 사용
2	Y	Y		Y	Y		거부
3	Y	Y		Y		Y	UID 사용
4	Y	Y		Y			거부
5	Y	Y			Y	Y	UID 사용
6	Y	Y			Y		거부
7	Y	Y				Y	UID 사용
8	Y	Y					거부
9	Y		Y	Y	Y	Y	UID 사용
10	Y		Y	Y	Y		거부
11	Y		Y	Y		Y	UID 사용
12	Y		Y	Y			거부
13	Y	M ³			Y	Y	DFT 사용 또는 UID ²
14	Y	M ³			Y		DFT 사용 또는 UID ²
15	Y	M ³				Y	거부 또는 UID ²
16	Y	M ³					거부 또는 UID ²
17				Y	Y		사용된 DFT
18				Y			거부
19					Y		DFT 사용
20							거부

표 4. 분산 관계형 데이터베이스로 리모트 액세스 (계속)

행	UID	PWD ¹	AVI	SEC(Y)	DFT	유효	액세스
키:							
UID	사용자 ID 송신						
PWD	암호 송신						
AVI	이미 확인된 인디케이터 설치						
SEC(Y)	SECURELOC(YES) 지정						
DFT	통신 서브시스템에 지정된 디폴트 사용자 ID						
유효한	사용자 ID와 암호는 유효합니다						
UID 사용							
	공급된 사용자 ID로 연결						
DFT 사용							
	디폴트 사용자 ID로 연결						
거부	연결되지 않음						
주:							
1.	암호 보호가 활동 중인 경우, 보호되는 암호가 송신됩니다.						
2.	암호 보호가 활동 중일 때 UID를 사용하십시오.						
3.	암호 보호가 활동 중인 경우, 사용자의 암호는 보안 관리자에 의해 검색되고, 보호되는 암호가 송신됩니다. 그러나 활동 중이 아닌 경우에는 어떠한 암호도 송신되지 않습니다.						

디폴트 사용자 프로파일을 사용해야 하는 경우가 생기지 않도록 하려면, 분산 관계형 데이터베이스(RDB) 오브젝트에 액세스를 해야 하는 모든 AR 사용자를 위해 AS에 사용자 프로파일을 작성하십시오. 그러나, 디폴트 사용자 프로파일을 사용하기로 결정하는 경우, 사용자가 정식 권한이 없이 시스템을 사용할 수 없도록 하십시오. 예를 들어, 다음의 명령은 디폴트 사용자 매개변수를 DFTUSER(QUSER)로 지정하며, 이것은 시스템이 통신 요구로부터 사용자 ID나 암호를 가지지 않고 작업 시작 요구를 받아들일 수 있도록 합니다. 통신 작업은 QUSER 사용자 프로파일을 사용하여 시작됩니다.

```
ADDCMNE SBSD(SAMPLE) DEV(*ALL) DFTUSER(QUSER)
```

TCP/IP를 사용한 DDM/DRDA 보안 요소

고유의 TCP/IP를 통한 DDM/DRDA는 APPC 통신과 연관되는 통신 장치, 모드, 보안 위치 속성 및 대화 보안 레벨과 같은 OS/400 통신 보안 서비스와 개념을 사용하지 못합니다. 그러므로, TCP/IP에 대한 보안 설정은 상당한 차이가 있습니다.

TCP/IP 서버에 대해 가능한 보안 유형은 다음과 같습니다.

- DDM/DRDA용 연결 보안 프로토콜
- DDM/DRDA용 보안 소켓층(SSL)
- DDM/DRDA용 인터넷 프로토콜 보안 프로토콜(IPSec)

iSeries 서버 관리자는 분산 자료 관리(DDM) 통신 보안에 대한 새로운 선택사항을 통해 사용 중인 포트를 블로킹함으로써 일정한 통신 모드를 제한할 수 있습니다. DDM용 포트 및 포트 제한사항에서 이에 대한 몇가지의 고려사항을 참조하십시오.

DDM 보안에 대한 자세한 정보는 다음을 참조하십시오.

- TCP/IP 네트워크의 어플리케이션 리퀘스터(AR) 보안
- TCP/IP 네트워크의 어플리케이션 서버(AS) 보안

DDM/DRDA용 연결 보안 프로토콜

TCP/IP를 통한 DDM/DRDA의 현재 iSeries용 DB2 UDB의 구현은 다음과 같은 몇 가지의 연결 보안 프로토콜을 지원합니다.

- 사용자 ID 전용
- 텍스트로만 이루어진 암호를 가진 사용자 ID
- 암호화된 암호를 갖고 있는 사용자 ID
- Kerberos

암호화된 자료 스트림을 사용하면 기존의 통신 추적 지원이 거의 쓸모가 없습니다. TCP/IP 어플리케이션 추적 (TRCTCPAPP) 명령은 암호화 이전 에이 명령을 사용하는 방법에 관한 기본 정보는 통신 추적 주제를 참조하십시오.

일반 텍스트로 부여하는 특정 암호에 대한 주의

iSeries가 연결 암호의 암호화를 지원하지만 RDB 디렉토리 항목 설정에 지정할 수 있는 연결 보안 옵션 중 하나는 *USRIDPWD입니다. (자세한 정보는 관계형 데이터베이스 디렉토리에 대한 작업에서 RDB 디렉토리 항목 추가 명령 및 관계형 데이터베이스 디렉토리 항목 변경 명령을 참조하십시오.)

연결이 설정된 서버가 *USRIDPWD 보안 옵션을 허용하면 연결 암호는 일반 텍스트로 부여할 수 있습니다. 또한 V5R3에서 SQL SET ENCRYPTION PASSWORD문과 ENCRYPT 기능을 사용하면 암호가 네트워크에서 일반 텍스트로 흐를 수 있습니다. 현재 자료 스트림을 암호화하는 솔루션은 두 가지가 있습니다. 하나는 IPsec을 사용하는 것입니다. 50 페이지의 『DDM/DRDA용 인터넷 프로토콜 보안 프로토콜(IPsec)』을 참조하십시오. 두 번째로, SSL을 지원하는 AR을 사용할 경우 해당 프로토콜을 사용하여 iSeries AS로부터 전송되는 자료를 암호화할 수 있습니다.

DDM/DRDA용 보안 소켓층(SSL)

iSeries용 DB2 UDB DRDA 클라이언트는 SSL을 지원하지 않습니다. 그러나, 유사한 기능이 인터넷 프로토콜 보안 프로토콜(IPsec)에 제공되어 있습니다.

DDM TCP/IP 서버는 SSL(소켓 층 보안) 자료 암호화 프로토콜을 지원합니다. 이 프로토콜을 사용하여 레코드 레벨 액세스를 위한 SSL을 지원하는 Java™용 iSeries Toolbox 및 iSeries Access 제품군 OLE DB Provider와 같은 클라이언트 및 SSL을 지원할 수 있는 독립 소프트웨어 벤더에서 제공하는 DDM 파일 I/O 클라이언트와 상호 작용할 수 있습니다.

iSeries DDM TCP/IP 서버를 갖고 있는 SSL을 사용하려면 서버에 잘 알려진 SSL 포트 448로 연결되는 클라이언트를 구성해야 합니다.

서버에서 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령에 대해 PWDRQD(*ENCRYPTED)를 지정하면 보안 소켓층(SSL)과 함께 유효한 암호를 모두 사용할 수 있습니다. 이는 서버가 암호를 포함하여 전체 자료 스트림이 암호화되어 있다는 것을 인식하기 때문에 가능합니다.

SSL에 대한 자세한 내용은 iSeries Information Center의 네트워킹 주제에 있는 SSL을 사용한 어플리케이션 보안을 참조하십시오.

필수 프로그램: PC 및 iSeries 서버에 SSL 지원을 설정하고 설치하는 데 관한 완전한 문서는 Windows용 iSeries Access를 참조하십시오.

iSeries 서버 요구사항: SSL을 통해 통신하는 iSeries 서버는 OS/400 V4R4 이상을 실행하고 다음을 설치해야 합니다.

- TCP/IP Connectivity Utilities for iSeries, 5769-TC1(기본 TCP/IP 지원)
- 암호화 액세스 제공자, 5769-ACx
- iSeries용 IBM HTTP Server, 5769-DG1(디지털 인증 관리자 DCM) 액세스용)
- DCM*(Digital Certificate Manager), 5769-SS1 - 보스 옵션 34
- 클라이언트 암호화, 5769-CEx -- 이 제품을 iSeries에 설치해야 하고 네트워크상의 모든 PC 클라이언트 필요한 SSL 클라이언트 코드를 검색해야 합니다. 이 제품은 SSL 통신을 실행하는 서버에 대해서는 필요하지 않고 오직 클라이언트에 대해서만 필요합니다(주 참조).

PC 요구사항(iSeries Access 제품군 및 DRDA를 사용하는 PC용): SSL을 통해 통신하는 네트워크상의 클라이언트 PC는 다음 제품 중 하나를 설치해야 합니다.

- 40비트 클라이언트 암호화, 5769-CE1
- 56비트 클라이언트 암호화, 5769-CE2
- 128비트 클라이언트 암호화, 5769-CE3

주: SSL 클라이언트 암호화 제품(5769-CEx)에 대한 서비스는 iSeries Access 제품군 서비스 팩이 아닌 별도의 서비스 팩을 통해 처리됩니다. 세부사항은 iSeries Access 제품군 홈 페이지의 Informational APAR II10598을 참조하십시오.

DDM/DRDA용 인터넷 프로토콜 보안 프로토콜(IPSec)

IPSec(인터넷 프로토콜 보안 프로토콜)은 암호화의 보안 서비스를 제공하는 네트워크 층에 있는 보안 프로토콜입니다. 이 서비스는 인터넷이나 인트라넷을 통한 기밀의 자료 전달을 지원합니다.

iSeries에서, IPSec 즉 VPN(가상의 사설망) 지원의 구성요소는(DRDA 또는 DDM과 같이) 어플리케이션에 관계없이 두 개의 IP 주소나 포트 조합 사이에서 모든 자료가 암호화될 수 있게 합니다. 사용자는 IPSec에 사

용되는 주소와 포트를 구성할 수 있습니다. IBM은 DRDA 액세스나 DDM 액세스용 IPsec에 대하여 포트 447을 사용할 것을 제안합니다. VPN 지원 설정에 관한 자세한 정보는 iSeries Information Center의 네트워크 주제에서 VPN(가상 사설망)을 참조하십시오.

어플리케이션(DRDA 또는 DDM)은 IPsec가 사용 중인지를 판별할 수 없기 때문에, 서버에서 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령에 대해 PWDRQD(*ENCRYPTED)를 지정함으로써 부과된 요구사항은 일반적으로 IPsec와 함께 유효한 암호를 사용하더라도 충족되지 않습니다. 따라서, 사용자는 IPsec로 PWDRQD(*ENCRYPTED)를 사용해서는 안됩니다.

DDM/DRDA용 포트 및 포트 제한사항

DDM/DRDA TCP/IP 서버는 448(잘알려진 SSL 포트) 뿐만 아니라 포트 447(잘알려진 DDM 포트)과 446(잘알려진 DRDA 포트)에서 청취합니다. iSeries용 DB2 UDB의 DDM 실시는 446과 447의 두 포트 간을 구분하지 못합니다. 그러나, DDM과 DRDA 양쪽의 액세스는 둘 중 하나의 포트에서 실행될 수 있습니다.

IPsec에 관련된 권장 규약을 사용하면, DDM TCP/IP 서버에 대한 포트 사용법은 다음과 같습니다.

- 텍스트 자료 스트림 비우기에 대해 446
- IPsec 암호화 자료 스트림에 대해 447(제안)
- SSL 암호화 자료 스트림에 대해 448(요구)

사용자는 TCP/IP 구성(CFGTCP) 명령을 사용하여 서버에서 하나 또는 그 이상의 포트 사용을 막을 수 있습니다. 이를 실행하려면, 그 명령의 "TCP/IP 포트 제한에 대한 작업" 옵션을 선택하십시오. 사용자는 QRWTLSTN을 실행시키는 사용자 프로파일(보통 QUSER) 외에 특정 사용자 프로파일만이 446과 같이 특정 포트를 사용할 수 있도록 제한사항을 추가할 수 있습니다. 이것은 실제로 446을 블록 표시합니다. 447이 IPsec로만 사용되도록 구성되었으면, 446의 블록 표시는 암호화된 자료 스트림만이 고유의 TCP/IP를 통한 DDM 및 DRDA 액세스에 사용되는 결과를 가져옵니다. 사용자는 447과 448을 둘 다 SSL 전용으로만 제한하도록 블록 표시할 수도 있습니다. 성능이나(SSL 기능을 할 수 있는 현재의 제한된 클라이언트의 가용성과 같이) 다른 이유로 이 예를 따르는 것은 실용적이지 않을 수도 있습니다. 그러나 이 예는 가능한 구성을 보여 줄 목적으로 주어진 것입니다.

인증 방법 협상

연결 시나리오에 따라 다른 레벨의 인증을 사용해야 합니다. 따라서 관리자는 각 RDB 디렉토리 항목의 우선 인증 방법 필드를 설정하여 어플리케이션 서버(AS)에 연결할 때 어플리케이션 리퀘스터(AR)가 요구하는 가장 낮은 보안 인증 방법을 설정할 수도 있습니다. 또, 관리자는 하위 보안 인증 방법을 선택함으로써 인증 방법을 결정하는 것을 서버와 협상하도록 할 수도 있습니다. 이 경우에도 우선 인증 방법이 시도되긴 하지만 AS가 우선 인증 방법을 받아들이지 않는다면, 서버 보안 설정과 암호 지원 가용성과 같은 다른 요소에 따라 하위 방법이 사용될 수도 있습니다. 예를 들어, 두 시스템이 물리적으로 보호되지 않는 환경에 있다면, 관리자는 하위 보안 인증방법을 허용하지 않고 Kerberos 인증을 요구하는 것을 선택할 수 있습니다.

어플리케이션 리퀘스터(클라이언트)쪽에서, 두 개의 메소드 중 하나를 사용하여 DRDA TCP/IP 연결 요구를 받았을 때 사용자 ID와 함께 암호를 송신할 수 있습니다. 두 가지 방법 중 어느 하나도 사용하지 않으면 CONNECT 명령은 오직 사용자 ID만 송신합니다.

암호를 송신하는 첫 번째 방법은 대화식 SQL 환경의 다음의 예에서와 같이 SQL CONNECT 명령문의 USER/USING 양식을 사용하는 것입니다.

```
CONNECT TO rdbname USER userid USING 'password'
```

삽입된 SQL을 사용하는 프로그램에서, 사용자 ID와 암호의 값은 USER/USING 데이터베이스의 호스트 변수에 포함될 수 있습니다.

다음은 CLI를 사용한 프로그램에서 사용자 ID와 암호가 호스트 변수에 저장되어 DRDA 어플리케이션 리퀘스터(AR)에 제공되는 방법의 예입니다.

```
SQLConnect(hdbc,sysname,SQL_NTS, /*do the connect to the application server */  
uid,SQL_NTS,pwd,SQL_NTS);
```

두 번째 암호 송신 방법은 서버 권한 부여 항목을 사용하여 TCP/IP를 통해 연결 요구를 송신하는 것입니다. 서버 권한 리스트는 시스템의 모든 사용자 프로파일과 연관됩니다. 디폴트에 의해, 리스트는 비어 있지만 사용자는 서버 인증 항목 추가(ADDSVRAUTE) 명령을 사용하여 항목을 추가할 수 있습니다. 사용자가 TCP/IP를 통한 DRDA 연결을 시도할 때, iSeries용 DB2 UDB 클라이언트(AR)는 클라이언트 작업을 실행하는 사용자 프로파일에 대한 서버 권한 리스트를 검사합니다. CONNECT문의 RDB 이름과 권한 부여 항목(반드시 대문자)의 SERVER 이름이 일치할 경우에, 해당 항목의 관련 USRID 매개변수가 연결 사용자 ID로 사용됩니다. PASSWORD 매개변수가 항목에 저장되어 있는 경우, 연결 요구 시 이 암호도 역시 송신됩니다.

또, 서버 권한부여 항목은 DDM 파일 I/O 조작을 위해 TCP/IP로 암호를 송신하는데도 사용될 수 있습니다. 사용자가 TCP/IP를 통한 DDM 연결을 시도할 때, iSeries용 DB2 UDB는 클라이언트 작업을 실행하는 사용자 프로파일에 대한 서버 권한 리스트를 검사합니다. RDB 이름(RDB 디렉토리 항목이 사용되는 경우)이나 'QDDMSERVER' 중 하나와 권한부여 항목의 SERVER 이름이 일치하는 경우, 해당 항목의 관련 USRID 매개변수는 연결 사용자 ID로 사용됩니다. PASSWORD 매개변수가 항목에 저장되어 있는 경우, 연결 요구 시 이 암호도 역시 송신됩니다.

서버 인증 항목 추가(ADDSVRAUTE) 명령으로 암호를 저장하려면 QRETSVRSEC 시스템 값을 '1'로 설정해야 합니다. 디폴트에 의해, 이 값은 '0'입니다. 이 값을 변경하려면 다음의 명령을 입력하십시오.

```
CHGSYSVAL QRETSVRSEC VALUE('1')
```

다음의 예는 RDB 디렉토리 항목을 사용할 때 서버 인증 항목 추가(ADDSVRAUTE) 명령 구문을 나타내고 있습니다.

```
ADDSVRAUTE USRPRF(user-profile) SERVER(rdbname) USRID(userid) PASSWORD(password)
```

USRPRF 매개변수는 어플리케이션 리퀘스터 작업을 실행하는 사용자 프로파일을 지정합니다. SERVER 매개변수에 넣는 것은 보통 사용자가 연결하려는 RDB 이름입니다. 예외는 RDB 디렉토리를 사용하기 위해 작성하지 않았던 DDM 파일을 사용하고 있을 때입니다. 그럴 경우에, SERVER 매개변수에 QDDMSERVER를 지정해야 합니다. RDB 이름을 지정할 때는 반드시 대문자여야 합니다. USRID 매개변수는 서버 작업을 실행할 사용자 프로파일을 지정합니다. PASSWORD 매개변수는 사용자 프로파일에 대한 암호를 지정합니다.

사용자가 USRPRF 매개변수를 생략하면 이것은 서버 인증 항목 추가(ADDSVRAUTE) 명령을 실행하는 사용자 프로파일에 디폴트 값으로 됩니다. 사용자가 USRID 매개변수를 생략하면, 이것은 USRPRF 매개변수의 값에 디폴트 값으로 됩니다. 사용자가 PASSWORD 매개변수를 생략하거나 QRETSVRSEC 값을 0으로 설정하면, 항목에 어떠한 암호도 저장되지 않으며 항목을 사용하여 연결을 시도할 때에 보안 메카니즘으로 사용자 ID만 시도될 것입니다.

- | DSPSVRAUTE 명령을 사용하여 서버 인증 리스트에 추가한 인증 항목을 판별할 수 있습니다. 자세한 정보는 CL 주제에서 DSPSVRAUTE (서버 인증 항목 표시) 명령 설명을 참조하십시오. 사용자 작성 프로그램의 QsyRetrieveServerEntries(QSYRTVSE) API를 사용할 수도 있습니다. API 주제에서 서버 인증 항목 검색(QSYRTVSE, QsyRetrieveServerEntries) API를 참조하십시오.

사용자는 서버 인증 항목 제거(RMVSVRAUTE) 명령을 사용하여 서버 권한 항목을 제거할 수 있습니다. 사용자는 서버 인증 항목 변경(CHGSVRAUTE) 명령을 사용하여 서버 권한 항목을 변경할 수 있습니다. 이 명령에 대한 자세한 정보는 Information Center의 제어 언어(CL) 주제를 참조하십시오.

서버 권한 항목이 RDB(관계형 데이터베이스)에 대해 존재하고 CONNECT 명령문의 USER/USING 양식도 사용되는 경우, 후자가 우선합니다.

Kerberos 소스 구성

DRDA와 DDM은 두 시스템 모두 Kerberos를 위해 구성되어 있으면 Kerberos 인증을 이용할 수 있습니다. Kerberos 구성에 관한 정보는 iSeries Information Center의 네트워크 인증 서비스 주제를 참조하십시오. 작업의 사용자 프로파일이 유효한 티켓 허가 티켓(TGT)을 가지고 있으면, DRDA 어플리케이션 리퀘스터(AR)는 이 TGT를 사용하여 서비스 티켓을 생성하고 사용자를 리모트 서버에 인증합니다. 사용할 수 있는 유효한 TGT를 가지고 있으면 직접적으로 암호를 요구하지 않기 때문에 서버 인증 항목을 사용할 필요가 없게 됩니다. 하지만 작업의 사용자 프로파일이 유효한 TGT를 가지고 있지 않으면, 사용자 ID와 암호는 서버 인증 항목으로부터 검색되어 필요한 TGT와 서비스 티켓을 생성하게 됩니다.

Kerberos를 사용할 때, RDB 디렉토리 항목의 리모트 위치(RMTLOCNAME)가 리모트 호스트명으로 입력되어야 합니다. IP 주소는 Kerberos 인증에는 작동하지 않습니다.

Kerberos 영역 이름이 DNS 접미사 이름과 다른 경우에, 이는 올바른 영역으로 맵핑되어야만 합니다. 그러기 위해서, Kerberos 구성 파일(krb5.conf)에 항목이 있어야 각 리모트 호스트명을 올바른 영역 이름에 매핑할 수 있습니다. 입력된 호스트명은 정확히 리모트 위치명(RMTLOCNAME)과 일치해야 합니다. DSPRDBDIRE 또는 DSPDDMF 명령으로 표시되는 리모트 위치 매개변수는 krb5.conf 파일의 정의역명과 일치해야 합니다. 다음 그래픽은 DSPRDBDIRE 화면의 예를 보여줍니다.

관계형 데이터베이스 상세 표시

관계형 데이터베이스 : RCHASXXX

리모트 위치:

리모트 위치 : rchasxxx.rchland.ibm.com

유형 : *IP

포트 번호 또는 서비스명 : *DRDA

리모트 인증 메소드 :

우선되는 방법 : *KERBEROS

하위 인증 허용 : *NOALWLOWER

텍스트 :

관계형 데이터베이스 유형 : *REMOTE

계속하려면 Enter를 누르십시오.
F3=종료 F12=취소

다음은 리모트 위치명과 일치하는 정의역명을 보여주는 krb5.conf 파일의 해당 내용의 부분입니다. (주: 파일 표시(DSPF) 명령은 구성 파일 내용을 표시하는데 사용됩니다.)

```
DSPF STMF('/QIBM/UserData/OS400/NetworkAuthentication/krb5.conf')
```

```
[domain_realm]
```

```
; Convert host names to realm names. Individual host names may be  
; specified. Domain suffixes may be specified with a leading period  
; and will apply to all host names ending in that suffix.  
rchasxxx.rchland.ibm.com = REALM.RCHLAND.IBM.COM
```

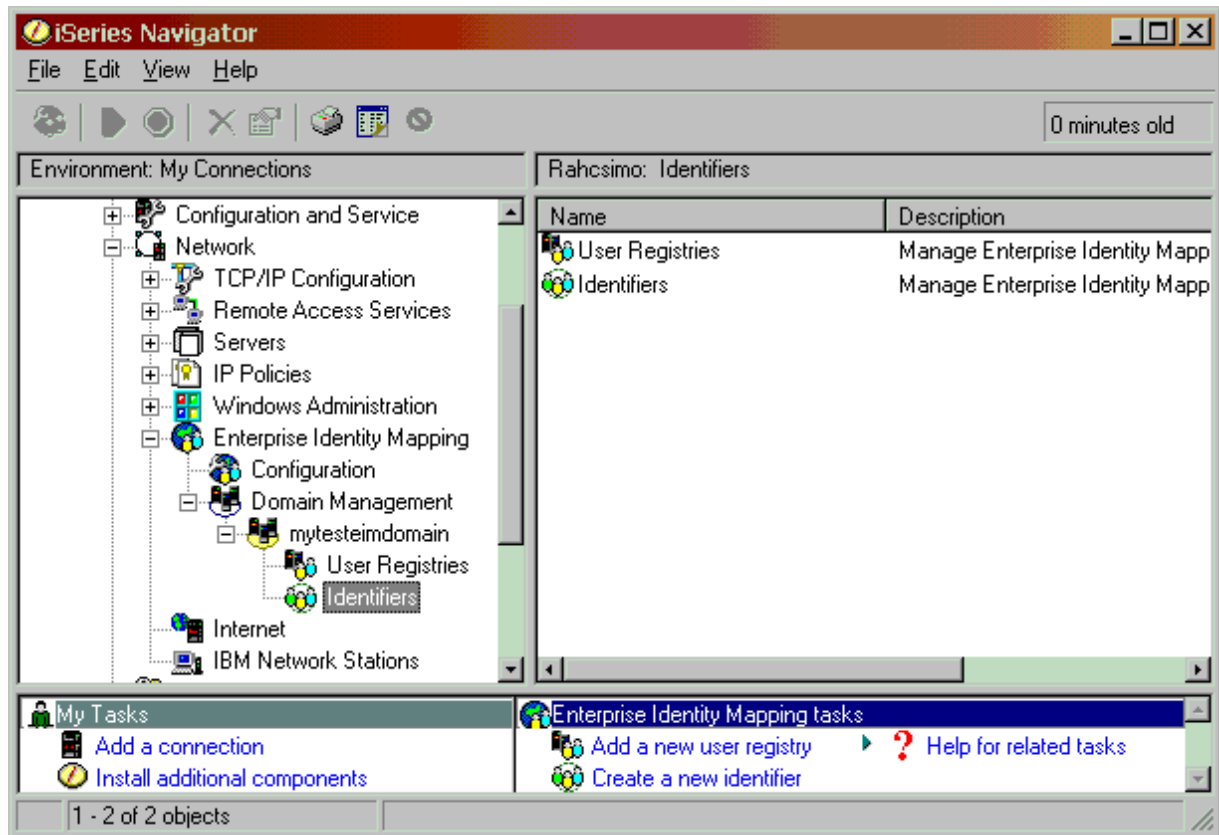
Kerberos를 사용하는 작업은 krb5.conf 파일에 구성 변경이 발생하면 재시작해야 합니다.

Kerberos DRDA 서비스명 정의

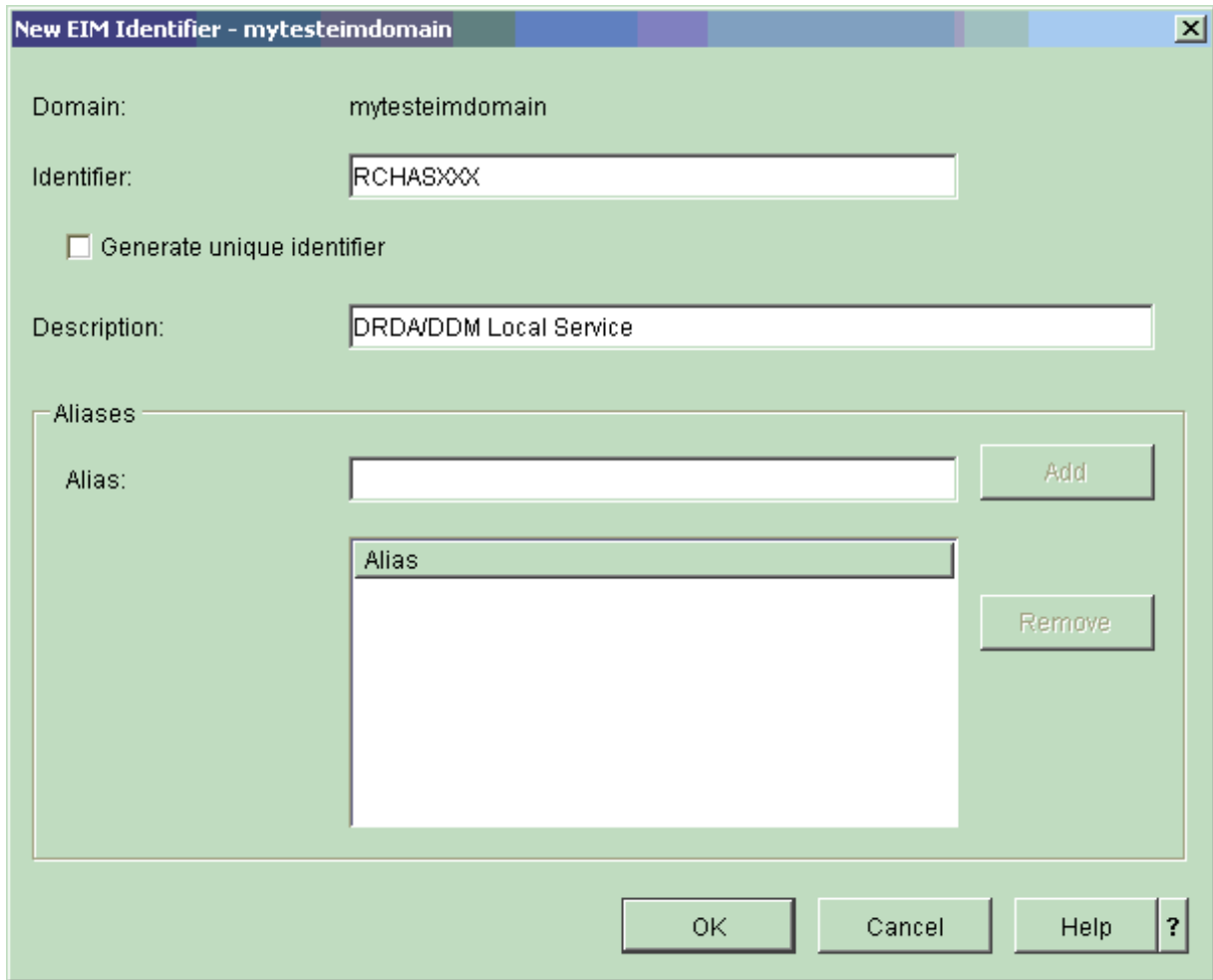
- 1 주: DRDA 레벨 5를 사용하는 버전 8 DB2 제품군 멤버와 함께 Kerberos 구성에 대한 자세한 정보는 네트
- 1 워크 인증 서비스를 참조하십시오.

비iSeries 서버에 연결하는데 Kerberos 인증을 사용하기 위해서는 비iSeries 서비스명이 Enterprise Identity Mapping(EIM) 하에 정의되어야 합니다. 자세한 정보는 iSeries Information Center의 Enterprise Identity Mapping(EIM) 주제를 참조하십시오. DRDA 서비스명을 정의하려면 다음 단계를 수행하십시오.

1. **iSeries Navigator**를 시작하십시오.
2. **Network**를 확장하십시오.
3. **Enterprise Identity Mapping**을 확장하십시오.
4. **Domain Management**를 확장하십시오.
5. 사용자의 EIM 정의역명을 확장하십시오.
6. **Identifiers**를 마우스 오른쪽 버튼으로 클릭한 다음 **New Identifiers**를 선택하십시오.

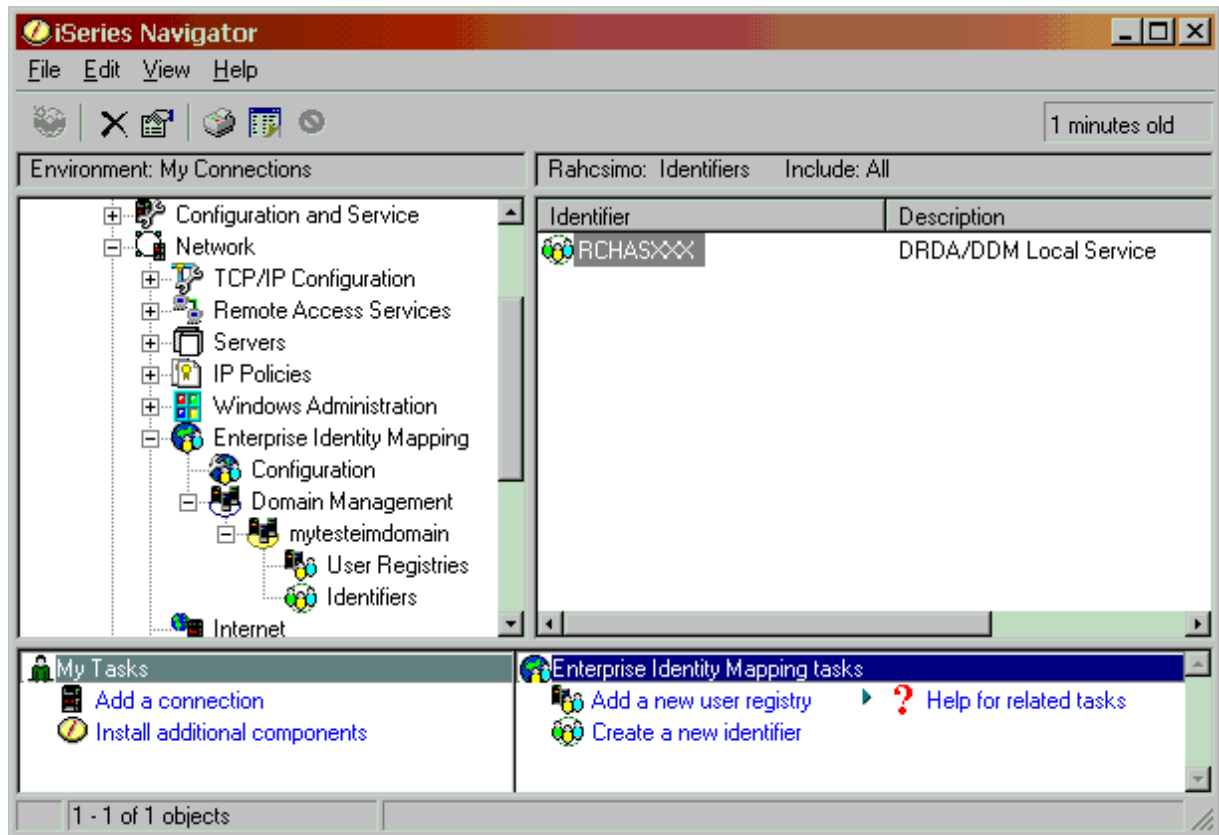


7. 로컬 RDB 이름을 identifier로 입력하고 필요하다면 설명도 입력하십시오.

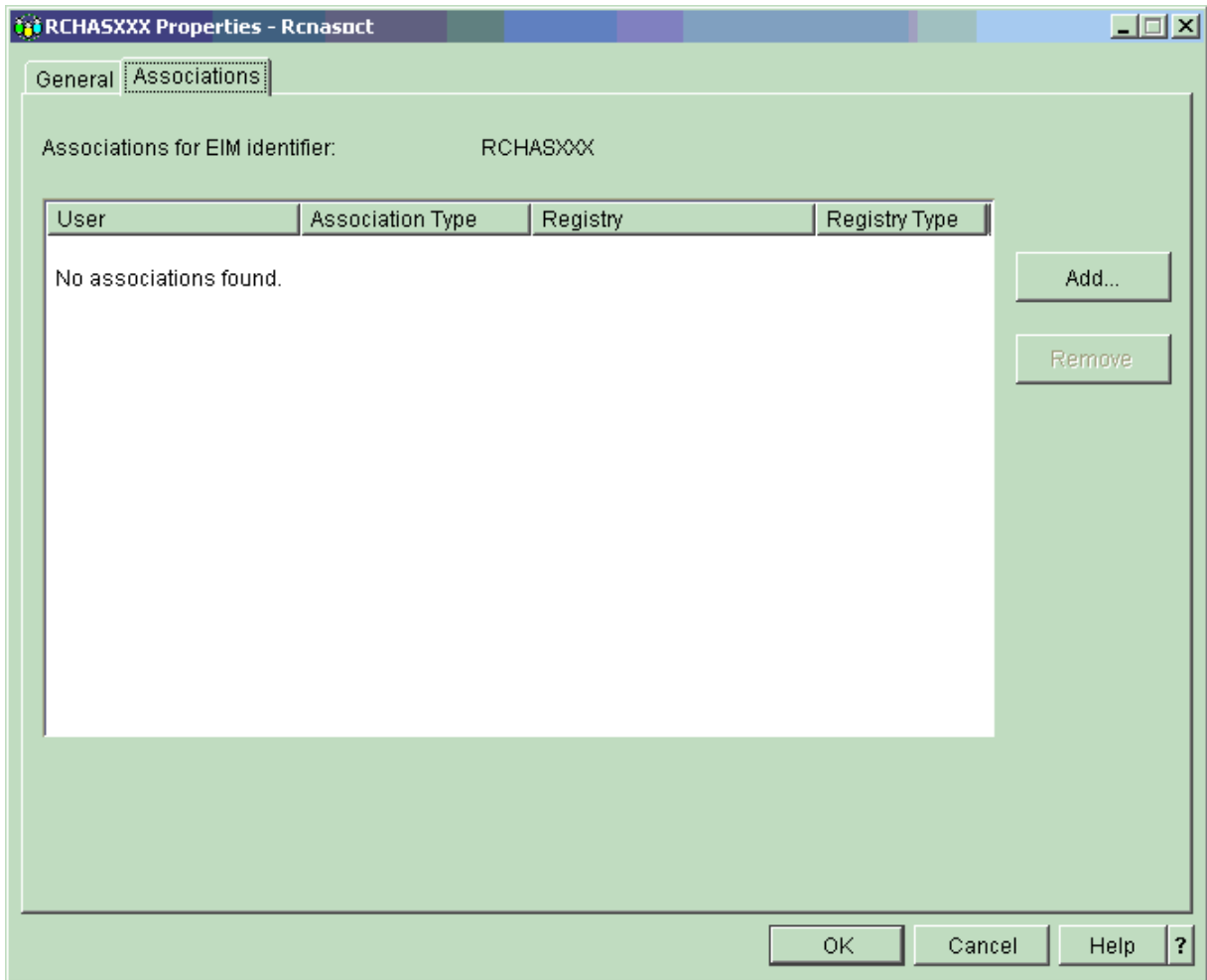


8. **OK**를 클릭하십시오.

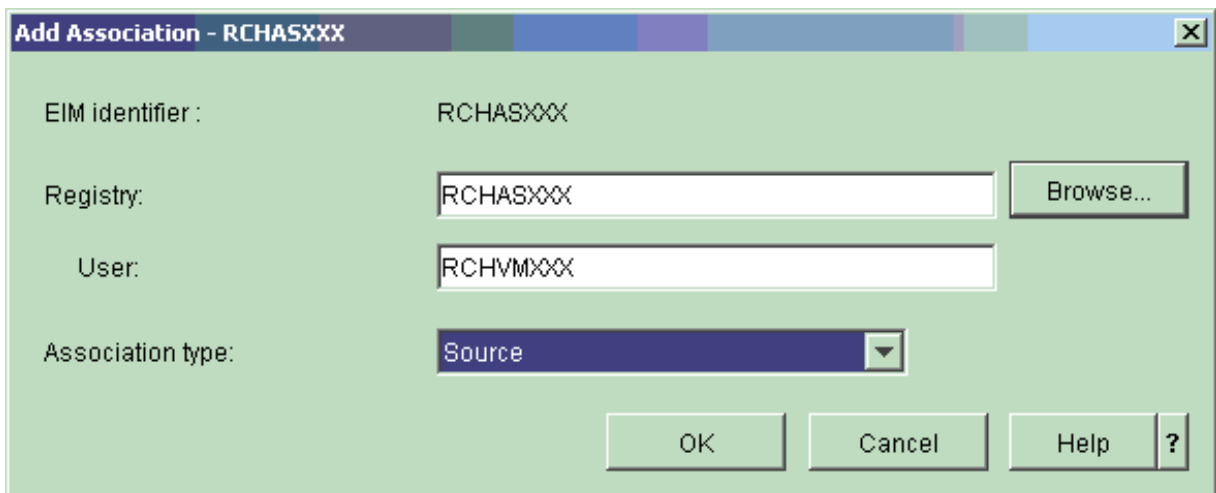
사용자가 작성한 ID는 iSeries Navigator의 오른쪽 분할 창에 나타납니다.



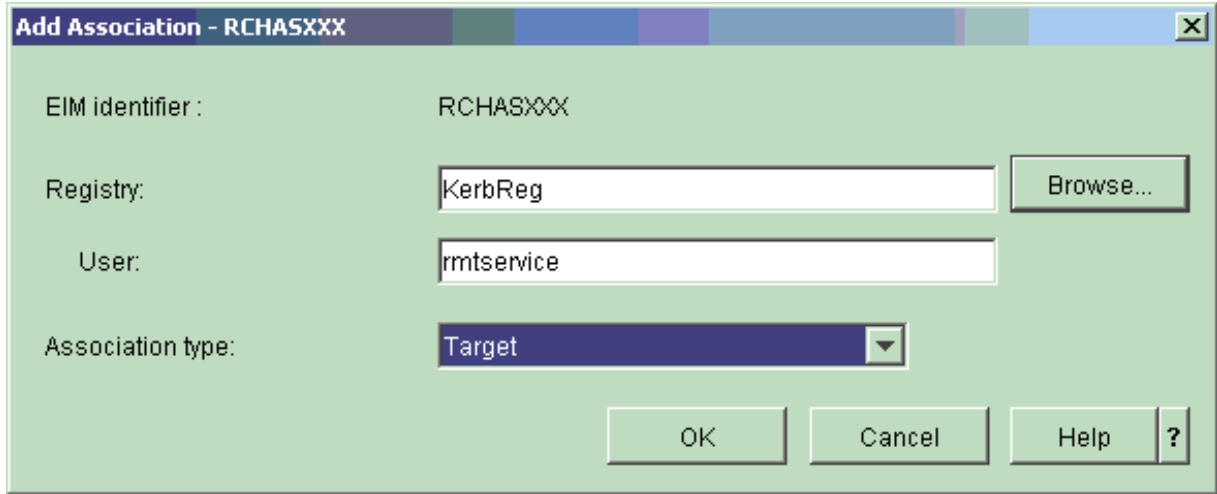
9. 작성한 ID를 마우스 오른쪽 버튼으로 클릭한 다음 **Properties**를 선택하십시오.
10. **Associations** 탭을 클릭하십시오.



11. **Add**를 클릭하여 새 연관을 추가하십시오.
12. 리모트 위치명(RMTLOCNAME)을 **User** 필드에 입력하고 Association type 필드에서 **Source**를 선택하십시오.



13. **OK**를 클릭하십시오. Identifier의 **Properties** 대화 상자로 돌아갑니다.
14. **Add**를 클릭하여 두 번째 연관을 입력하십시오.
15. **Registry** 필드에 Kerberos 레지스트리를 입력하십시오. **User** 필드에 리모트 서버의 Kerberos 서비스명을 입력하십시오. Association type 필드의 **Target**을 선택하십시오.



16. **OK**를 클릭하십시오.

TCP/IP 네트워크의 어플리케이션 서버(AS) 보안

TCP/IP 서버는 clear-text 암호가 있는 사용자 ID의 디폴트 보안을 가지고 있습니다. 즉, 서버가 설치될 때 인바운드 TCP/IP 연결 요구는 실행하는 서버 작업이 속하는 사용자 ID에 수반되는 최소한의 clear-text 암호를 가져야 한다는 뜻입니다. 보안은 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령으로 또는 iSeries Navigator의 네트워크 -> 서버 -> TCP/IP -> DDM 서버 등록 정보에서 변경될 수 있습니다. 이 설정을 변경하려면 *IOSYSCFG 특수 권한이 있어야 합니다.

하위 서버 보안에 사용되는 설정이 두 가지 있습니다.

- PWDRQD(*NO)

암호가 필요하지 않습니다.

- PWDRQD(*VLDONLY)

암호가 요구되지는 않지만 송신된 경우 유효해야 합니다.

*NO와 *VLDONLY의 차이점은 암호가 클라이언트 시스템으로부터 송신된 경우, *NO 옵션에서 무시됩니다. 하지만, *VLDONLY 옵션에서 암호가 송신되는 경우, 수반된 사용자 ID에 대한 암호는 유효하고 부정확한 액세스는 거부됩니다.

요구되는 암호화된 암호 또는 PWDRQD(*ENCRYPTED) 및 Kerberos 또는PWDRQD(*KERBEROS)는 상위 보안 레벨에서 사용됩니다. Kerberos가 사용된 경우, 사용자 프로파일은 Enterprise Identity Mapping(EIM)을 사용하여 Kerberos 프린시플에 맵핑되어야 합니다. 자세한 정보는 iSeries Information Center의 Enterprise Identity Mapping(EIM) 주제를 참조하십시오.

- 주: *ENCRYPTED를 사용하려면 암호화 제품 5669-AC2 또는 5769-AC3이나 그와 동등한 제품을 두 시스템에 설치해야 합니다.

다음은 암호화된 암호가 사용자 ID를 수반해야 한다는 것을 지정하기 위해 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령을 사용한 예입니다. 이 옵션을 설정하려면 다음을 입력하십시오.

CHGDDMTCPA PWDRQD(*ENCRYPTED)

주: DDM/DRDA TCP/IP 서버는 V4R4에서 향상되어 암호 대체라고 불리는 암호 암호화 양식을 지원합니다. V4R5에서는 더욱 널리 사용되는 암호 암호화 기법(Diffie-Hellman 공용키 알고리즘이라고 함)이 구현되었습니다. 이것은 DRDA 표준 알고리즘으로서 가장 최근에 릴리스된 IBM DRDA 어플리케이션 리퀘스터에 사용됩니다. PC 클라이언트의 DDM 파일 액세스에 대해서는 구식 암호 대체 알고리즘이 우선적으로 사용됩니다. V5R1에서는 '강력한' 암호 대체 알고리즘도 지원되었습니다. SSL(Secure Sockets Layer) 자료 스트림이 사용됨에 따라, 클라이언트와 서버는 사용될 보안 메커니즘을 협상하고, 세 가지 암호화 방법 중 하나가 PWDRQD(*ENCRYPTED) 요구사항을 충족시킵니다.

DRDA 서버 액세스 제어 종료 프로그램

DRDA 서버의 보안 피쳐는 APPC와 TCP/IP와 같이 사용할 수 있도록 네트워크 속성 변경(CHGNETA) 명령의 DDMACC 매개변수를 DRDA로 연장합니다. 이 매개변수는 이전에 DDM 파일 I/O 액세스에만 적용되었습니다. 해당 기능에서 DRDA의 사용은 연결 요구에 한하여 사용되며 연결된 후에 자료 요구에 대하여는 사용이 제한되지 않습니다.

사용자가 이 보안 기능의 사용을 선택하지 않는 경우에는 보통 어떠한 것도 할 필요가 없습니다. 단, 알 수 없는 기능 코드가 수신될 때 작동을 거부하도록 코드화되어 있는 DDM 종료 프로그램을 현재 사용 중에 있는 경우와 DRDA를 사용하여 그 시스템에서 자료에 액세스하고 있는 경우에만 예외입니다. 이 경우에, 사용자는 기능 코드가 'SQLCNN'이면 '1'을 리턴하여 DRDA 액세스가 가능해지도록 종료 프로그램을 변경하여야 합니다.

DRDA 연결을 블로킹하거나 필터링하기 위해 종료 프로그램을 사용하려면, 사용자는 새로운 DRDA 종료 프로그램을 작성하거나 기존의 것을 변경해야 합니다.

주: 사용자의 시스템이 복수 데이터베이스(ASP 그룹)로 구성된 경우, 종료 프로그램은 시스템 데이터베이스의 라이브러리에 상주해야 합니다(1-32 범위의 보조 기억장치 풀에).

iSeries Information Center의 분산 데이터베이스 관리 주제에서 DRDA 종료 프로그램 작성에 대한 일반 지침을 볼 수 있습니다.

이 보안 피처는 항목 매개변수 구조에서 프로그램으로 입력될 수 있는 DRDA 기능 코드를 요구 기능의 리스트에 추가합니다. 'SQLCNN'(SQL 연결 요구)으로 명명되는 기능 코드는 DRDA 연결 요구가 프로세스 중임을 표시합니다. (63 페이지의 그림 7에서 FUNC 매개변수를 참조하십시오.) APP(어플리케이션) 항목 매개변수는 DRDA 연결 요구 호출에 대하여 '*DDM ' 대신에 '*DRDA '로 정해져 있습니다.

사용자가 DRDA에 대해 종료 프로그램을 코드화함에 따라 매개변수 구조의 다음 필드가 사용될 수 있습니다.

- USER 필드는 프로그램이 사용자 프로파일 ID에 기초하여 DRDA 액세스를 허락하거나 거부할 수 있도록 합니다.
- RDBNAME 필드는 사용자가 연결하려는 RDB 이름을 포함합니다. 이는 시스템 데이터베이스 또는 사용자 데이터베이스(ASP 그룹)가 될 수 있습니다. 이 필드는 복수의 데이터베이스가 구성되어 있는 환경에서 하나 이상의 데이터베이스에 액세스를 거부하려는 경우 유용합니다.
- 63 페이지의 그림 7에서 SRVNAME 매개변수는 종료 프로그램의 호출자에 의해 설정되거나 설정되지 않을 이 매개변수가 설정되면 이것은 클라이언트 시스템명을 표시합니다. 이것이 정해지지 않으면 값은 *N이 됩니다. DRDA 어플리케이션 리퀘스터가 iSeries 서버일 때는 언제나 이 값으로 지정됩니다.
- TYPDEFN 매개변수는 연결 중인 클라이언트 유형에 대한 추가 정보를 제공합니다. IBM 주 프레임에 대하여 TYPDEFN은 QTDSQL370이 됩니다. iSeries 서버의 경우에는 QTDSQL400이 됩니다. Intel® PC의 경우, QTDSQLX86이 됩니다. RS/6000® 클라이언트의 경우, QTDSQLASC가 됩니다.
- PRDID(제품 ID) 매개변수는 해당 제품의 릴리스 레벨과 함께 연결을 시도하는 제품을 식별합니다. 다음은 이 코드의 처음 세 문자의 리스트 일부입니다. (사용자는 먼저 비IBM 코드를 확인한 후 종료 프로그램에서 이것을 사용하여야 합니다).

QSQ IBM iSeries용 DB2 UDB

DSN IBM DB2 UDB for z/OS

SQL IBM DB2 연결™(이전에는 DDCS로 명명됨)

ARI IBM DB2 UDB for VSE & VM

GTW Oracle Corporation 제품

GVW DB/DC Systems 제품 그랜드 뷰(Grandview)

XDB XDB Systems 제품

IFX Informix® Software 제품

RUM 데이터베이스 액세스에 대한 Wall Data Rumba

SIG StarQuest 제품

STH FileTek 제품

I **JCC** IBM DB2 Universal Driver for SQLJ 및 JDBC

vv가 버전이고 rr이 릴리스이고 m이 수정 레벨인 경우, 필드의 나머지는 vvrrm으로서 조직됩니다.

종료 프로그램이 RTNCODE 값 '0'을 리턴하고 연결 요구가 iSeries 클라이언트로부터 오고 이 때, 사용자에게 연결 장애를 나타내는 메시지가 SQ30060인 경우, '사용자는 관계형 데이터베이스에 대한 권한을 가지지 못합니다...'. 일반적으로, 종료 프로그램에 의한 액세스 거부에 대한 응답은 DRDA RDBATHRM 응답 메시지가 지입니다. 이 메시지는 사용자에게 관계형 데이터베이스에 대한 권한이 없음을 나타냅니다. 클라이언트 플랫폼에 따라 사용자에게 오류를 다르게 보고할 수 있다는 점을 주의하십시오.

제한사항:

- 기능 검사가 사용자 종료 프로그램에서 발생되면, 프로그램은 같은 응답 메시지를 리턴하고 연결 시도에 실패합니다. 종료 프로그램은 iSeries용 DB2 UDB에 어떠한 확장성 갱신도 해서는 안됩니다. 그렇지 않으면, 예측할 수 없는 결과가 발생할 수 있습니다.
- 사용자는 사전시작 서버 작업의 이전 호출로 열려 있는 파일에 액세스를 시도하기 위해 종료 프로그램을 사용해서는 안됩니다.
- V5R2 이전에는 TCP/IP 서버에서 사용된 사전시작 작업이 후속 사용을 위해 재순환될 때 제한이 더 심해졌습니다. 다음 사용을 위해 작업을 준비하기 위해 일정한 클린업이 행해집니다. 이 처리의 일부는 *ELIGIBLE 값을 가진 ACTGRP 매개변수로 활성 그룹 재생(RCLACTGRP) 명령을 사용하는 것입니다. 그 결과로, RCLACTGRP에 의해 파기된 활성 그룹에 대해 사전시작 서버 작업에서 모든 잔여 연계를 사용하는 시도를 함으로써(프로그램이 더이상 존재하지 않는 오브젝트의 전부 또는 일부에 관련하려고 하는 경우) MCH3402 예외사항이 발생할 수 있습니다. 이 제한을 피하는 방법 하나는 다음과 같이 QRWTSRVR 사전시작 작업의 MAXUSE 값을 1로 설정하는 것입니다: CHGPJE SBS(DQSYSWRK) PGM(QRWTSRVR) MAXUSE(1).

예: DRDA 서버 액세스 제어 종료 프로그램

63 페이지의 그림 7은 전체 DRDA 조작과 사용자 ID가 'ALIEN'일 때를 제외하고 모든 DRDA 연결을 허용하는 PL/I 사용자 종료 프로그램의 예를 나타내고 있습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.


```

/*****/
/*
/* PROGRAM NAME: UEPALIEN
/*
/* FUNCTION: USER EXIT PROGRAM THAT IS DESIGNED TO
*/
/* RETURN AN UNSUCCESSFUL RETURN CODE WHEN
/* USERID 'ALIEN' ATTEMPTS A DRDA CONNECTION.
*/
/* IT ALLOWS ALL TYPES OF DDM OPERATIONS.
*/
/* EXECUTION: CALLED WHEN ESTABLISHED AS THE USER EXIT
/* PROGRAM.
*/
/* ALL PARAMETER VARIABLES ARE PASSED IN EXCEPT:
/*
/* RTNCODE - USER EXIT RETURN CODE ON WHETHER FUNCTION IS
/* ALLOWED: '1' INDICATES SUCCESS; '0' FAILURE.
/*
/*****/

UEPALIEN: PROCEDURE (RTNCODE,CHARFLD);

DECLARE RTNCODE CHAR(1); /* DECLARATION OF THE EXIT
/* PROGRAM RETURN CODE. IT
/* INFORMS REQUEST HANDLER
/* WHETHER REQUEST IS ALLOWED.
DECLARE
1 CHARFLD, /* DECLARATION OF THE CHAR
/* FIELD PASSED IN ON THE CALL.

2 USER CHAR(10), /* USER PROFILE OF DDM/DRDA USER*/
2 APP CHAR(10), /* APPLICATION NAME
2 FUNC CHAR(10), /* REQUESTED FUNCTION
2 OBJECT CHAR(10), /* FILE NAME
2 DIRECT CHAR(10), /* LIBRARY NAME
2 MEMBER CHAR(10), /* MEMBER NAME
2 RESERVED CHAR(10), /* RESERVED FIELD
2 LNGLTH PIC '999999', /* LENGTH OF USED SPACE IN REST
2 REST, /* REST OF SPACE = CHAR(2000)

3 LUNAME CHAR(10), /* REMOTE LU NAME (IF SNA)
3 SRVNAME CHAR(10), /* REMOTE SERVER NAME
3 TYPDEFN CHAR(9), /* TYPE DEF NAME OF DRDA AR
3 PRDID, /* PRODUCT ID OF DRDA AR

5 PRODUCT CHAR(3), /* PRODUCT CODE
5 VERSION CHAR(2), /* VERSION ID
5 RELEASE CHAR(2), /* RELEASE ID
5 MOD CHAR(1), /* MODIFICATION LEVEL
5 RDBNAME CHAR(18), /* RDB NAME
5 REMAING CHAR(1965), /* REMAINING VARIABLE SPACE

START:
IF (USER = 'ALIEN' & /* IF USER IS 'ALIEN' AND
FUNC = 'SQLCNN') THEN /* FUNCTION IS DRDA CONNECT
RTNCODE = '0'; /* SET RETURN CODE TO UNSUCCESSFUL*/
ELSE /* IF ANY OTHER USER, OR DDM
RTNCODE = '1'; /* SET RETURN CODE TO SUCCESSFUL

END UEPALIEN;

```

그림 7. Example PL/I User Exit Program

DRDA용 오브젝트 관련 보안

iSeries 서버가 어플리케이션 서버(AS)인 경우, 그에 관련된 데이터베이스 표에 대한 액세스를 제어하도록 보안을 실행하는 오브젝트에 관련된 두 개의 레벨이 있습니다.

DDMACC 매개변수는 네트워크 속성 변경(CHGNETA) 명령에 사용되며, 이 서버의 표가 다른 시스템에 의해서라도 액세스될 수 있는지의 여부와 만약 액세스될 수 있다면 수신 DRDA 요구를 검사하는 보안 레벨은 어느 것인지를 표시합니다.

- DDMACC 매개변수에 *REJECT를 지정하면, AS에 의해 수신된 모든 분산 관계형 데이터베이스(RDB) 요구가 거부됩니다. 그럼에도 불구하고 이 시스템(어플리케이션 리퀘스터(AR)로써)은 SQL 요구를 사용하여 이것을 허용하는 다른 시스템의 표에 액세스할 수 있습니다. 어떠한 리모트 시스템이라도 *REJECT를 지정하는 iSeries 서버에서 데이터베이스에 액세스할 수 없습니다.

SQL 요구가 이미 사용중인 동안 *REJECT를 지정하면, 이 시스템의 데이터베이스에 대한 액세스를 요구하는 모든 시스템의 새로운 작업들이 모두 거부되고 오류 메시지가 해당 작업들로 리턴됩니다. 그러나 기존 작업에는 영향을 미치지 않습니다.

- DDMACC 매개변수에 *OBJAUT가 지정되면, AS에서 일반 오브젝트 레벨의 보안을 사용하게 됩니다.

DDMACC 매개변수는 초기에 *OBJAUT로 설정됩니다. *OBJAUT 값은 모든 리모트 요구를 허용하지만 모든 리모트 요구는 이 AS에서 오브젝트 권한에 의해 제어됩니다. DDMACC 값이 *OBJAUT인 경우, 작업에 사용되는 사용자 프로파일은 개인, 공용, 그룹 또는 허용한 권한을 통해 해당 오브젝트 권한을 가지거나, 이 프로파일이 AR 작업에 필요한 오브젝트에 대한 권한 부여 리스트에 들어 있어야 합니다. 시스템에서 각 SQL 오브젝트에 대하여, 오브젝트에 액세스할 권한이 전체 사용자 또는(사용자 ID에 의한) 특정 사용자에게 부여될 수도 있지만 어떠한 사용자도 그 권한을 얻지 못할 수도 있습니다.

오브젝트에 대한 권한을 부여받아야 하는 사용자 ID는 AS 작업의 사용자 ID입니다. 어떤 사용자 프로파일에서 AS 작업이 실행되는지에 관한 자세한 정보는 APPC 네트워크의 DDM 보안 요소 주제를 참조하십시오.

TCP/IP 연결의 경우, 서버 작업은 초기에 QUSER하에서 실행하기 시작합니다. 사용자 ID가 유효화된 후에, 교환이 일어나고 작업은 그 다음에 연결 요구 시 지정된 사용자 프로파일하에서 실행합니다. 작업은 사용자 프로파일의 속성(예를 들어, 라이브러리 리스트)을 계승합니다.

*OBJAUT 값이 지정되면, 이것은 더이상의 확인(iSeries 오브젝트 레벨 이상의 보안)이 요구되지 않음을 나타냅니다.

- DDM 작업에서 선택형의 사용자 공급 사용자 종료 프로그램(또는 액세스 제어 프로그램)이 DDMACC 매개변수에 지정되면 추가 보안 레벨이 사용됩니다. 사용자 종료 프로그램은 DDM 클라이언트의 사용자가 특정한 명령을 사용하여 iSeries 서버에서 특정 파일에 액세스할 수 있는지 여부를 제어하는 데 사용될 수 있습니다.

DRDA 작업에 대하여, 사용자 공급에 의한 선택형의 사용자 종료 프로그램(액세스 제어 프로그램)이 DDMACC 매개변수에 지정되어 있는 경우, 시스템은 *DBJAUT가 지정되어 있는 것처럼 해당 입력을 하나의 예외로 처리합니다. 사용자 작성 종료 프로그램이 DRDA 작업에서 나타내는 유일한 효과는 연결 요구를 거부하는 것입니다. 자세한 정보는 DRDA 서버 액세스 제어 종료 프로그램 주제를 참조하십시오.

DDMACC 매개변수는 초기에 *OBJAUT로 설정되며 네트워크 속성 변경(CHGNETA) 명령을 사용하여 이전에 설명된 여러 개의 값 중 하나로 변경될 수 있습니다. 그리고 이 현재 값은 네트워크 속성 표시(DSPNETA) 명령에 의해 표시될 수 있습니다. 또한, 사용자는 네트워크 속성 검색(RTVNETA) 명령을 사용하여 CL 프로그램에서 이 값을 얻을 수 있습니다.

DDMACC 매개변수 값이 변경되면, 변경이 즉시 실행될지라도 이 시스템(AS)에서 시작된 새로운 분산 관계형 데이터베이스(RDB) 작업에만 영향을 미칩니다. 변경되기 전에 이 AS에서 실행하던 작업은 원래의 값을 계속 사용합니다.

DDMACC 매개변수에 대한 설명은 V5R1 보충 매뉴얼 웹 사이트에 있는 *Communications Management*



매뉴얼에서 CHGNETA(네트워크 속성 변경) 명령에 대한 설명을 참조하십시오.

분산 관계형 데이터베이스 오브젝트에 권한

관계형 데이터베이스 오브젝트에 대한 사용자 권한을 부여하거나 취소하기 위해서는 SQL GRANT 및 REVOKE 문을 사용하거나, 제어 언어(CL) 명령인 오브젝트 권한 부여(GRTOBJAUT) 및 오브젝트 권한 취소(RVKOBJAUT) 명령을 사용할 수 있습니다. SQL GRANT 및 REVOKE 명령문만이 패키지, 표 및 뷰에서 조작합니다. 어떤 경우에는 GRTOBJAUT 및 RVKOBJAUT를 사용하여 사용자에게 명령 및 프로그램과 같이 다른 오브젝트에 대한 권한을 부여하는 것이 필요합니다.

SQL 명령문에 대한 검사 권한은 명령문의 정적, 동적 여부 또는 상호 실행되고 있는가의 여부에 따라 달라집니다.

CRTSQLxxx 명령의 USRPRF 매개변수에서 지정할 수 있는 값의 의미와 그 의미가 정적 및 동적 명령문에서 어떻게 달라지는지에 대한 세부사항은 SQL 프로그래밍 개념 서적의 보안 섹션을 참조하십시오.

대화식 SQL 명령문에서, 명령문을 처리하는 사용자의 권한에 대한 권한을 점검합니다. 허용한 권한은 대화식 SQL 명령문에 사용되지 않습니다.

분산 관계형 데이터베이스(RDB) 어플리케이션을 실행하는 사용자는 어플리케이션 서버(AS)에서 SQL 패키지를 실행하는 권한을 요구합니다. GRANT EXECUTE ON PACKAGE 명령문은 SQL 패키지의 소유자나 그에 대한 관리 특권을 가지는 사용자가 SQL 패키지에 들어 있는 명령문을 실행하는 특권을 지정된 사용자에게 부여합니다. 사용자는 이 명령문을 사용하여 AS나 AS에서 하나 또는 그 이상의 사용자 프로파일에 관한 리스트에 대해 권한을 부여받은 모든 사용자에게 SQL 패키지에 들어 있는 명령문들 실행하는 특권을 줄 수 있습니다.

일반적으로, 사용자는 CRTSQLxxx 명령을 사용하여 작성된 분산 어플리케이션 프로그램에 대한 권한을 부여받은 경우, 패키지에 대한 처리권을 가집니다. 패키지가 SQL 패키지 작성(CRTSQLPKG) 명령을 사용하여 작성되어 있는 경우, 사용자에게 패키지에 대한 처리 특권을 부여할 수도 있습니다. 사용자는 SQL 프로그램에서 또는 대화식 SQL을 사용하여 이 명령문을 보낼 수 있습니다. 다음은 샘플 명령문입니다.

```
GRANT EXECUTE
ON PACKAGE SPIFFY.PARTS1
TO PUBLIC
```

REVOKE EXECUTE ON PACKAGE 명령문은 SQL 패키지의 소유자나 이에 대한 관리 특권을 갖고 있는 사용자이면 누구나 지정된 사용자로부터 받은 SQL 패키지에 들어 있는 명령문을 실행하는 특권을 제거할 수 있도록 해줍니다. 사용자는 AS나 AS에서 하나 이상의 사용자 프로파일에 관한 리스트에 대해 권한을 부여받은 모든 사용자에게 주는 EXECUTE 특권을 제거할 수 있습니다.

같은 사용자에게 한 번 이상 같은 특권을 부여하였다면, 이 사용자에게 대하여 해당 특권을 취소함으로써 부여한 모든 특권이 무효가 됩니다. 이전에 사용자에게 부여한 SQL 패키지에 대한 EXECUTE 특권을 취소하는 경우, 특권 부여자가 누구든 관계없이 해당 SQL 패키지에 대해 부여받은 EXECUTE 특권은 전부 무효가 됩니다. 다음은 샘플 명령문입니다.

```
REVOKE EXECUTE  
ON PACKAGE SPIFFY.PARTS1  
FROM PUBLIC
```

또한, 사용자는 오브젝트 권한 부여(GRTOBJAUT) 명령을 사용하여 SQL 패키지에 대한 권한을 부여하거나 오브젝트 권한 취소(RVKOBSAUT) 명령을 사용하여 SQL 패키지에 대한 권한을 취소할 수 있습니다.

분산 관계형 데이터베이스에 대한 허용 권한 하에서 실행되는 프로그램

분산 관계형 데이터베이스(RDB) 프로그램은 허용된 권한 하에서 실행될 수 있는데, 이것은 이 프로그램이 실행 중인 동안 프로그램이 사용하는 오브젝트에 대하여 프로그램 소유자의 권한을 허용한다는 의미입니다. 프로그램 작성 시 *SQL 사전검파일러 옵션을 사용하여 프로그램을 명명하면 프로그램은 프로그램 소유자의 사용자 프로파일 하에서 실행됩니다.

다른 시스템의 SQL 패키지는 언제나 이 패키지의 모든 정적 SQL 명령문에 대한 패키지 소유자의 권한을 허용합니다. 또한 CRTSQL_{xxx} 명령에 OPTION(*SQL)을 지정하여 iSeries 서버에서 SQL 패키지를 작성하는 경우에도 패키지의 모든 정적 SQL문에 대한 패키지 소유자의 권한을 허용하게 됩니다.

분산 관계형 데이터베이스(RDB) 관리자는 프로그램 허용 표시(DSPPGMADP) 명령을 사용하여 어플리케이션 서버에 관한 보안 노출을 검사할 수 있습니다. DSPPGMADP 명령은 아래에서 보는 바와 같이 지정된 사용자 프로파일을 사용하는 프로그램과 SQL 패키지를 표시합니다. 또한, 사용자는 명령의 결과를 프린터나 출력 파일로 송신할 수도 있습니다.

허용하는 프로그램

사용자 프로파일 : MPSUP

오브젝트	라이브러리	유형	속성	텍스트
INVENT	SPIFFY	*PGM		프로그램 허용
CLIENT1	SPIFFY	*PGM		프로그램 허용
TESTINV	TEST	*PGM	CLP	재고 관리 테스트 프로그램
INVENT1	SPIFFY	*SQLPKG		SQL 패키지
CLIENT1	SPIFFY	*SQLPKG		SQL 패키지
TESTINV	SPIFFY	*SQLPKG		SQL 패키지

맨 아래

계속하려면 Enter를 누르십시오.

F3=종료 F12=취소 F17=맨 위 F18=맨 아래
(C) COPYRIGHT IBM CORP. 1980, 1991.

분산 관계형 데이터베이스(DRDB)에서 보호 전략

iSeries 분산 관계형 데이터베이스(RDB)에서 네트워크 보안은 모든 어플리케이션 서버(AS)에서 중요한 자료에 대해 권한이 없는 액세스가 방지될 수 있도록 설계되어야 합니다. 그러나 관계형 데이터베이스의 분산형 특성으로 인해, 보안 계획은 네트워크에서 자료 가용성이 불필요하게 제한되지 않도록 보장해야 합니다.

분산 관계형 데이터베이스(RDB) 관리자가 해야 할 결정 중 하나는 네트워크에 속하는 각 시스템에 대한 시스템 보안 레벨이 적절해야 한다는 것입니다. 시스템 보안 레벨 10은 시스템 사이트에서 실제의 보안을 제공하지만 어플리케이션 서버에 대해서는 보안을 제공하지 않습니다. 시스템 보안 레벨 20은 로컬 및 리모트 시스템이 올바르게 식별되어 있는지를 확인하기 위해 네트워크 보안 검사를 실시하기 때문에 어플리케이션 서버에 어느 정도의 보호를 제공합니다. 그러나, 이 레벨은 권한이 없는 액세스로부터 중요한 데이터베이스 요소를 보호하는 데 필요한 오브젝트 권한을 제공하지 않습니다. iSeries 서버 보안 레벨 30 및 그 이상은 특정의 시스템 오브젝트를 보호하려는 네트워크에 속한 시스템에 대하여 선택하도록 권장하는 레벨입니다.

또한 분산 관계형 데이터베이스(RDB) 관리자는 네트워크 상의 어플리케이션 리퀘스터(AR)와 어플리케이션 서버 사이에 통신을 설치하는 방법을 고려하여야 합니다. 다음과 같이 해결해야 할 몇 가지 질문이 있습니다.

- 디폴트 사용자 프로파일이 AS에 존재해야 합니까?

네트워크를 전체를 통해 많은 사용자 프로파일을 유지보수하기 어려울 수 있습니다. 그러나, 통신 서버시스템 항목에 디폴트 사용자 프로파일을 작성하면 AS가 보안 위치가 아닌 경우에 AS가 열려 통신 요구를 수신하게 됩니다. 어떤 경우에는 이것이 허용가능한 상황이 될 수도 있지만 또다른 경우에는 디폴트 사용자 프로파일이 시스템 보호 기능을 떨어뜨려서 보안 요구를 충족시키기 어려울 수 있습니다.

예를 들어, 많은 AR을 공급하는 시스템은 높은 레벨의 보안을 요구합니다. 그런 시스템의 데이터베이스가 손실 또는 손상된다면 전체 네트워크에 영향을 줄 수 있습니다. 액세스를 요구하는 잠재적인 모든 사용자를 식별하는 AS에서 사용자 프로파일이나 그룹 프로파일을 작성하는 것이 가능하기 때문에, 데이터베이스 관

리자가 통신 서브시스템이나 분산 관계형 데이터베이스(RDB) 작업을 관리하는 서브시스템에 대한 디폴트 사용자 프로파일 작성을 고려하는 것은 필요하지 않습니다.

이와는 달리, 네트워크 내의 다른 시스템에 AS로서 거의 활동하지 않으며 센서티브하거나 중요한 자료를 포함하지도 않는 iSeries 서버는 분산 관계형 데이터베이스(RDB) 작업을 관리하는 통신 서브시스템에 대해 디폴트 사용자 프로파일을 사용할 수 있습니다. 이것은 같은 어플리케이션이 이 데이터베이스에서 작업을 처리하는 데 있어서 다른 시스템 전체에 의해 사용되는 경우에 특징적인 효과를 나타낼 수 있습니다.

엄격히 말하면, 디폴트 사용자의 개념은 APPC의 사용에만 적용됩니다. 그러나, 이와 같은 기술이 TCP/IP 를 사용하는 시스템에 사용될 수 있습니다. 단일 사용자 ID는 서버 작업이 실행하는 조건하에서 설치될 수 있습니다. 서버 인증 항목 추가(ADDSVRAUTE) 명령은 모든 AR에서 사용된다면 사용자 ID가 연결되는 모든 사용자가 사용하도록 지정할 수 있습니다. 서버 권한 항목은 AS에서 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령의 PWDRQD 매개변수 설정값에 따라 서버 권한 항목에 대해 지정된 암호를 가질 수 있거나 암호로 *NONE을 지정할 수 있습니다. 이 속성의 디폴트 값은 암호가 요구되는 것입니다.

- 데이터베이스 오브젝트에 대한 액세스 처리 방법은?

오브젝트에 대한 권한은 개인 권한, 그룹 권한, 공용 권한 및 권한 리스트를 통해 부여될 수 있습니다. 사용자 프로파일(또는 디폴트 프로파일)이 통신 요구를 받아들이는 AS에 존재해야 하는 상황에서 사용자가 오브젝트에 대한 권한을 부여받는 방법은 실행에 영향을 줄 수 있습니다.

가능하면 언제든지 그룹 권한이나 권한 리스트를 사용하여 분산 관계형 데이터베이스(RDB) 오브젝트에 대해 액세스를 부여하십시오. 개인 권한 전체를 검토하는 것보다 이를 점검하는 것이 시간과 시스템 자원을 절약하는 방법입니다.

TCP/IP 연결에서, 사용자는 사용자 ID를 자세히 나타낼 수 있기 때문에 AS에 연결할 수 있는 각 사용자에 대한 개인 사용자 ID가 필요하지 않습니다.

제 5 장 iSeries 분산 관계형 데이터베이스 설정

iSeries 분산 관계형 데이터베이스에 대한 실시간 지원은 OS/400 프로그램이 제공합니다. 따라서 오퍼레이팅 시스템을 설치할 때 분산 관계형 데이터베이스(RDB) 지원이 설치됩니다. 그러나, 일부 설정 작업을 위해서는 특히 APPC 환경에서 어플리케이션 리퀘스터와 어플리케이션 서버가 작업을 송신하고 수신할 준비가 되어 있어야 합니다. 대화식, 일괄처리, 스푼 및 통신 작업을 제어하는 데 하나 이상의 서브시스템을 사용할 수 있습니다. 네트워크의 모든 어플리케이션 리퀘스터(AR)에는 연결 정보를 사용하여 설정한 관계형 데이터베이스 디렉토리가 있어야 합니다. 마지막으로, 네트워크에 있는 어플리케이션 서버의 표에 자료를 기록합니다.

관계형 데이터베이스 디렉토리에는 통신 네트워크 매개변수로 변환된 데이터베이스 이름과 값이 있습니다. AR은 구성될 수 있는 로컬 데이터베이스 및 임의의 로컬 사용자 데이터베이스를 비롯하여 네트워크의 각 데이터베이스 항목이 있어야 합니다. 이 로컬 항목은 시스템에 의해 자동으로 또는 수동으로 추가될 수 있습니다. 각 디렉토리 항목은 고유한 관계형 데이터베이스 이름과 대응하는 통신 경로 정보로 구성됩니다. V5R2에서 아웃바운드 연결을 위한 우선 암호 보안에 관한 정보를 지정할 수 있습니다. ARD 프로그램이 제공하는 액세스의 경우, ARD 프로그램 이름을 관계형 데이터베이스 디렉토리 항목에 추가해야 합니다.

자료를 데이터베이스에 입력하는 여러 가지 방법이 있습니다. SQL 어플리케이션 프로그램, 다른 고급 언어 어플리케이션 프로그램 또는 다음 방법 중 하나를 사용할 수 있습니다.

- 대화식 SQL
- OS/400 조회 관리
- 자료 파일 유틸리티(DFU)
- 파일 복사(CPYF) 명령

iSeries 분산 관계형 데이터베이스(RDB)를 설정하려면 다음 주제에 관해 일정한 지식이 있어야 합니다.

- iSeries 서버에 관한 작업 관리
- 사용자 관계형 데이터베이스에 관한 DRDA 고쳐사항
- 관계형 데이터베이스 디렉토리 사용
- DRDA 보안 설정
- DRDA 또는 APPC용 TCP/IP 서버 설정
- 대화식 SQL용 SQL 패키지 설정
- DDM 파일 설정
- 분산 관계형 데이터베이스에서 자료를 표에 로딩

이 장에서는 이러한 주제를 소개하고 분산 관계형 데이터베이스(RDB) 작업을 위한 iSeries 서버 설정에 대해 다룹니다.

다른 서버의 분산 관계형 데이터베이스(RDB) 네트워크에 대한 연결 및 설정 정보는 *Distributed Relational Database Cross-Platform Connectivity(SG24-4311-02)* 서적을 참조하십시오.

iSeries 서버에 관한 작업 관리

iSeries 서버에서 행하는 모든 작업은 작업 관리 기능을 통해 제공됩니다. iSeries 서버에서, 전문화된 운영 환경을 설계함으로써 서로 다른 작업 유형을 처리하고 사용자 서버의 요구를 만족시킬 수 있습니다. 그러나, 오퍼레이팅 시스템이 설치되어 있을 때 이것은 대화식의 일괄처리, 통신 및 스펴 처리를 지원하는 작업 관리 환경을 포함합니다.

서버에서, 모든 사용자 작업은 서브시스템 설명에 의해 정의되는 서브시스템이라는 환경에서 운영되며, 서버는 그 곳에서 처리와 자원을 조정합니다. 작업이 같은 서브시스템에 배치되어 있는 경우에 공동의 특성을 갖고 있는 작업 그룹을 다른 작업과는 별도로 제어할 수 있습니다. 사용자는 요구되는 서브시스템을 시작하고 종료함으로써 실행 중인 작업을 지원하고 원하는 대로 성능 특성을 유지보수할 수 있습니다.

서버에서 실행하는 기본 작업 유형은 대화식, 통신, 일괄처리, 스펴, 자동시작 및 사전시작 등입니다.

대화식 작업(job)은 워크스테이션을 사인 온할 때 시작하고 사인 오프할 때 끝납니다. APPC 통신 일괄처리 작업은 다른 시스템으로부터 프로그램 시작 요구를 받아 시작되는 작업입니다. 비통신 일괄처리 작업은 작업 대기행렬로부터 시작됩니다. 작업 대기행렬은 통신 일괄처리 작업을 시작 중일 때에는 사용되지 않습니다. 스펴링 기능은 입력과 출력 모두에 대해 제공됩니다. 자동시작 작업은 반복 작업이나 1회의 초기화 작업을 실행합니다. 자동시작 작업은 특정 서브시스템과 연관되어 있으며, 서브시스템이 시작되는 각 시간마다 이것과 연관된 자동시작 작업이 시작됩니다. 사전시작 작업은 리모트 프로그램이 프로그램 시작 요구를 송신하기 전에 실행을 시작하는 작업입니다.

서브시스템에 대한 자세한 정보는 다음 주제를 참조하십시오.

- DRDA용 작업 관리 환경 설정
- APPC용 서브시스템 설정에 대한 고려사항

주: V5R2에서 DRDA TCP/IP 연결에 사용되는 DDM TCP/IP 서버 사전시작 작업은 디폴트로 QUSRWRK 서브시스템에서 실행됩니다. V5R2 이전에는 QSYSWRK에서 실행되었습니다. QUSRWRK는 사용자 작업 서브시스템입니다. 이는 사용자 대신 작업하기 위해 서버에 의해 시작되는 작업을 포함합니다. 작업을 사전시작 작업으로 디스패치하는 DRDA ‘리스너’ 작업은 QSYSWRK에서 실행됩니다. TCP/IP 서버 설정 및 관리에 대한 세부사항은 113 페이지의 『TCP/IP 서버 관리』를 참조하십시오.

DRDA용 작업 관리 환경 설정

제어 서브시스템이라는 한 서브시스템은 사용자가 서버를 로드할 때 자동으로 시작합니다. 2개의 제어 서브시스템 구성은 IBM에 의해 공급되며, 사용자는 변경하지 않고 이 구성을 사용할 수 있습니다. 첫 번째 구성은 다음과 같은 서브시스템을 포함합니다.

- 제어 서브시스템, QBASE는 대화식, 일괄처리 및 통신 작업을 지원합니다.
- QSPL은 스펴링 읽기 장치(readers) 및 출력기의 처리를 지원합니다.

- QSYSWRK는 TCP/IP를 사용하여 다양한 서버 기능을 지원합니다.
- QUSRWRK는 사용자 작업 서브시스템입니다. 이는 사용자 대신 작업하기 위해 서버에 의해 시작되는 작업을 포함합니다.

QBASE는 서버가 시작될 때 자동으로 시작합니다. QBASE에서 자동으로 시작된 작업은 QSPL을 시작합니다.

공급되는 두 번째 제어 서브시스템 구성은 이보다 더 복잡합니다. 이 구성은 다음과 같은 서브시스템을 포함합니다.

- 제어 서브시스템, QCTL은 콘솔에서 시작됩니다.
- QINTER는 다른 작업 장치에서 시작되는 대화식 작업을 지원합니다.
- QCMN은 통신 작업을 지원합니다.
- QBATCH는 일괄처리 작업을 지원합니다.
- QSPL은 스플링 읽기 장치(Readers) 및 출력기의 처리를 지원합니다.
- QSYSWRK는 TCP/IP를 사용하여 다양한 서버 기능을 지원합니다.
- QUSRWRK는 사용자 작업 서브시스템입니다. 이는 사용자 대신 작업하기 위해 서버에 의해 시작되는 작업을 포함합니다.

QCTL 제어 서브시스템을 사용하기 위해 사용자의 구성을 변경하면, 서브시스템은 시스템이 시작될 때 자동으로 시작합니다. QCTL에서 자동으로 시작된 작업은 다른 서브시스템을 시작합니다.

시스템 값 변경(CHGSYSVAL) 명령에서 시스템 값 QCTLSBSD(제어 서브시스템)를 QCTL로 변경하고 시스템을 다시 시작함으로써 사용자의 서브시스템 구성을 QBASE로부터 QCTL로 변경할 수 있습니다.

서브시스템 설명 변경(CHGSBSD) 명령을 사용하여 IBM이 공급하는 서브시스템 설명이나 사용자 작성 서브시스템 설명을 변경할 수 있습니다. 이 명령을 사용하여 기억장치 풀 크기, 기억장치 활동 레벨 및 활성 서브시스템의 서브시스템 설명에 대한 작업 최대수를 변경할 수 있습니다.

iSeries 서버에서의 작업 관리, 서브시스템 및 작업에 대한 자세한 정보는 Information Center의 작업 관리 주제를 참조하십시오. 통신 및 통신 서브시스템의 작업 관리에 대한 자세한 정보는 V5R1 보충 매뉴얼 웹 사이트의 *Communications Management*

 매뉴얼을 참조하십시오.

APPC용 서브시스템 설정에 대한 고려사항

SNA 네트워크를 사용한 분산 관계형 데이터베이스(RDB)에서, 통신 작업과 대화식 작업은 관리자가 각 시스템 관리를 위해 계획해야 하는 작업의 기본 유형입니다. 네트워크의 서버들은 통신 작업을 시작하여 어플리케이션 리퀘스터(AR)의 요구를 처리합니다. 다른 서버에 대한 AR의 통신 요구는 보통 로컬 시스템에서의 대화식 또는 일괄처리 작업에서 시작됩니다. 분산 관계형 데이터베이스(RDB) 네트워크 서버에 대해 유효한 작업 관리 환경은 네트워크에서 각각의 어플리케이션 서버(AS)와 AR에 관한 특정의 요구에 맞게 시스템 자원을 할당함으로써 네트워크 전체의 성능을 증진시킵니다.

OS/400 라이선스가 부여된 프로그램이 첫 번째로 설치되어 있을 때, QBASE는 디폴트 제어 서브시스템입니다. 제어 서브시스템으로서 QBASE는 두 서브시스템 QBASE와 QSPL 사이에 시스템 자원을 지정합니다. 대화식, 통신 작업, 일괄처리 작업 등은 QBASE 서브시스템 내에 자원을 할당합니다. 단지 스폴된 작업만이 다른 서브시스템, QSPL하에서 관리됩니다. QCTL 제어 서버시스템을 사용하는 것보다 통신 작업 대 대화식 작업을 처리하기 위한 시스템 자원의 제어가 덜 필요합니다.

QCTL 서브시스템 구성을 사용하여 시스템이 기억장치 풀과 다른 시스템 자원을 이미 할당한 4개의 추가 서브시스템을 제어합니다. QCTL 서브시스템을 변경하거나 사용자 소유의 서브시스템을 작성하면 사용자의 자원 처리를 훨씬 더 유연하게 조절하도록 해줍니다.

Spiffy사 분산 관계형 데이터베이스(RDB) 네트워크의 일부 시스템에 대한 다른 시스템 요구는 최고의 네트워크 효율을 제공하는 다른 작업 관리 환경을 요구할 수 있습니다. 다음의 거론사항은 분산 관계형 데이터베이스(RDB) 관리자가 Spiffy 분산 관계형 데이터베이스(RDB) 네트워크의 각 iSeries 서버의 요구를 충족시키기 위해 작업 관리 서브시스템을 계획하는 방법을 제시합니다.

Spiffy사 시스템 조직에서, 소형 판매 대리점은 자체의 사용자가 서버에서 실행하는 다양한 작업에 대한 QBASE 레벨의 제어를 만족시킬 수 있습니다. 예를 들어 지역 AR로부터 소형 판매 대리점의 관계형 데이터베이스에 대한 (출하를 위한 판매업자 재고 레벨 갱신) 요구는 통신 작업으로 처리됩니다. 지역 AS에 대한 판매 대리점 사용자의 요구, 즉 현재 지역의 재고에 없는 부품 요구는 판매 대리점 서버에서 대화식 작업으로서 처리됩니다. 양측의 활동 모두가 상대적으로 작은 작업입니다. 그 이유는 판매 대리점이 규모가 작아서 처리해야 할 서비스 주문량이 상대적으로 적기 때문입니다. QBASE 서브시스템의 자원에 대한 조정으로 이 기업망이 판매 대리점의 대화식 통신 요구에 대해 요구하는 제어 레벨이 제공됩니다.

한편, 대형 판매 대리점은 QCTL 서브시스템을 통해 작업을 관리할 수 있습니다. 이것은 다른 작업(work)이 다른 유형의 작업(jobs)과 연관된 상태로 로드하기 때문입니다.

로컬 관계형 데이터베이스로 부품에 대한 조회를, 또는 지역 센터 AS로 판매 대리점에 재고가 없는 부품에 대한 조회를 요구하면 매일 계약되는 서비스 주문량이 많아질 수 있습니다. 이 유형의 활동은 판매 대리점의 시스템에서 대화식 작업을 시작합니다. 또한, 판매 대리점은 기업망 전직원 레코드 보유, 마케팅, 판매 계획 및 보고서 작성 등과 같이 분산 관계형 데이터베이스(RDB) 관련 작업이 아닌 다수의 대화식 작업을 시작합니다. 실행 정보에 대해서 또는 재고나 작업 계획(work plans) 갱신에 대해서 해당 지역 센터에서 이 판매 대리점으로 요구하는 것은 해당 판매 대리점이 분리된 환경에서 관리하려는 통신 작업(job)입니다. 또한, 대형 판매 대리점은 지역 센터에서 재고에 없는 부품에 대해 또다른 판매 대리점으로부터 요구를 받을 수 있습니다.

대형 판매 대리점의 경우, QINTER와 QCMN에 대하여 분리 서브시스템 관리와 QCTL 구성은 자체의 서버 작업 환경을 관리하는 데 보다 더 유연한 처리를 제공합니다. 이 예에서, 판매 대리점 서버의 대화식 통신 작업은 다른 작업 유형보다 더 많은 서버 자원을 할당받을 수 있습니다. 이에 더하여, 통신 작업이 이 시스템에 대하여 일반적으로 대화식 작업보다 적으면, QINTER와 QCMN 양쪽에 대해 서브시스템 설명을 변경함으로써 자원이 대화식 작업쪽으로 목표가 정해질 수 있습니다.

Spiffy사 지역 센터 원근법에 대해 주문에 따라 맞춘 작업 관리 환경도 중요합니다. Spiffy 네트워크에서, 지역 센터는 이것이 정기의 부품 출하 자료를 갖고 있는 판매 대리점 재고표를 갱신하거나 특정의 수리 작업에

대해 신규 혹은 갱신된 서비스 계획을 포함하는 서비스 계획표를 갱신할 때, 각 판매 대리점에 대한 AR입니다. 이 작업 중 일부는 시스템 사용량이 일반적으로 적은 이른 아침이나 늦은 오후에 대화식 작업(지역 시스템에서)으로서 실행되거나 정규 업무 시간 이후에 일괄처리 작업(지역 서버에서)으로서 실행될 수 있습니다. 관리자는 QINTER와 QBATCH 서브시스템을 주문에 따라 맞추어 특정의 처리 시간과 자원 요구에 맞게 제공할 수 있습니다.

또한, 판매 대리점에서 재고에 없는 부품, 특정 서비스 작업(조종 랙(steering rack) 재설정 등)에 대한 서비스 계획, 또는 판매 대리점 관계형 데이터베이스에 대한 마지막 갱신 이후의 기술적인 게시나 취소 통지에 대해 해당 지역의 관계형 데이터베이스에 조회해야 할 때, 지역 센터는 각 판매 대리점의 AS이기도 합니다. 통신 작업은 전부 QCMN에서 관리될 수 있습니다.

그러나, KC000(캔자스 시티) 지역 센터와 이것이 지원하는 판매 대리점에 의한 분산 관계형 데이터베이스(RDB) 네트워크 사용에 관하여 일부 특정 측면을 자세히 살펴 보면 캔자스 시티의 분산 관계형 데이터베이스(RDB) 관리자에게 다른 대안을 제시하고 있습니다.

KC000 서버는 매일 하루에 수백 개의 서비스 주문을 처리하는 여러 개의 대형 판매 대리점과 매일 하루에 20개 이하의 서비스 주문을 처리하는 소수의 소형 판매 대리점을 지원합니다. 나머지의 중형 판매 대리점은 각각 하루에 100개의 서비스 주문을 처리합니다. 분산 관계형 데이터베이스(RDB) 관리자에게 나타나는 한 가지 문제점은 KC000 서버가 다른 시스템으로부터 받는 모든 통신 요구에 대한 최선의 처리 방법에 관한 것입니다. 대형 판매 대리점은 요구를 갖고 있는 QCMN 자원을 제어할 수도 있으므로 네트워크에 있는 다른 시스템에 대한 응답 시간과 비용이 불만족스러울 수 밖에 없습니다.

분산 관계형 데이터베이스(RDB) 관리자는 추가 통신 서브시스템을 작성할 수 있으며 따라서, 각 클래스의 판매 대리점(소형, 중형 또는 대형)은 AS로부터 지원을 요구하여 일반적으로 더 나은 응답을 받을 수 있습니다. 각 서브시스템 설명에 대하여 서브시스템 속성, 사전시작 작업 항목, 통신 작업 항목 및 라우팅 항목의 주문에 따라 맞춤으로써, 관리자가 서브시스템에서 활동할 수 있는 작업 수와 서브시스템에서 처리되는 작업 수를 제어합니다.

관리자는 다음의 예에서와 같이, 라우팅 항목을 추가하여 작업의 우선순위를 제어하는 클래스를 지정하고 CMPVAL 매개변수에 QCNTEDDM을 지정함으로써 DRDA/DDM 작업의 클래스(및 우선순위)를 변경할 수 있습니다.

```
ADDRTGE SBS(D(QCMN) SEQNBR(280) CLS(QINTER) CMPVAL('QCNTEDDM' 37)
```

또한, 다음의 예에서와 같이, 관리자는 QCNTEDDM을 사전시작 작업으로 지정함으로써 DRDA/DDM 작업에 사전시작 작업을 추가할 수 있습니다.

```
ADDPJE SBS(D(QCMN) PGM(QCNTEDDM)
```

iSeries 서버에서의 작업 관리에 대한 자세한 정보는 iSeries Information Center의 작업 관리 주제를 참조하십시오. 속성 변경, 통신의 작업 항목과 라우팅 항목에 대한 자세한 정보는 V5R1 보충 매뉴얼 웹 사이트의

Communications Management  매뉴얼을 참조하십시오.

사용자 관계형 데이터베이스에 관한 DRDA 고려사항

사용자는 서버에 독립형 보조 기억장치 풀을 구성함으로써 iSeries 서버에 추가로 관계형 데이터베이스를 작성할 수 있습니다. 각 독립형 보조 기억장치 풀 그룹은 관계형 데이터베이스입니다. 이 책에서는 이를 ‘사용자 데이터베이스’라고 칭합니다. 이는 독립형 보조 기억장치 풀 그룹 디스크에 존재하는 모든 데이터베이스 오브젝트로 이루어집니다. 또, 독립형 보조 기억장치 풀이 연결변환되는 iSeries 서버의 시스템 관계형 데이터베이스(이 책에서는 ‘시스템 데이터베이스’로 칭함)의 모든 데이터베이스 오브젝트는 논리적으로 사용자 관계형 데이터베이스에 포함됩니다. 하지만 확약 제어 측면에서는 시스템 데이터베이스는 다르게 취급됩니다. 자세한 정보는 iSeries Information Center의 트랜잭션 및 확약 제어 주제를 참조하십시오.

방금 언급한 확약 제어 고려사항으로 부과되는 것 외에도 사용자 데이터베이스 작성과 사용에 관한 수많은 규칙이 있습니다. 한 예가 현재 스테드를 위한 사용자 데이터베이스(보조 기억장치 풀(ASP) 그룹)로 설정되어 있는 AR로부터 데이터베이스에 연결하는데 APPC로 보호되는 DUW 대화를 사용할 수 없습니다. 다른 예는 사용자 데이터베이스에서 작성된 임의의 스키마 이름은 그 사용자 데이터베이스 또는 연관된 시스템 데이터베이스에 미리 존재해서는 안됩니다. 이 제한사항에 관한 자세한 정보는 iSeries Information Center의 SQL 참조서 주제를 참조하십시오.

어떤 DRDA 관련 오브젝트는 사용자 데이터베이스에 포함될 수 없습니다. DDM 사용자 종료 프로그램은 어느 어플리케이션 리퀘스터 드라이버 프로그램과 마찬가지로 시스템 데이터베이스의 라이브러리에 상주해야 합니다.

사용자 데이터베이스를 연결변환하는 프로세스는 RDB 디렉토리가 일정 기간동안 사용할 수 없도록 만든다는 점을 명심해야 합니다. 이는 DRDA 어플리케이션 리퀘스터(AR) 또는 어플리케이션 서버(AS)가 RDB 디렉토리를 이용하여 표시되거나 시간종료하려는 시도를 발생시킬 수 있습니다. 복수의 데이터베이스가 동시에 연결변환되면 데이터베이스의 연결변환으로 사용불가능이 발생하기 때문에 디렉토리 조작 시간종료에 훨씬 더 많이 노출되게 됩니다. 아래 언급한 대로, 사용자 데이터베이스가 처음 연결변환될 때, 서버는 그 데이터베이스에 대해 디렉토리 항목을 추가하려는 시도를 합니다. 동시 연결변환 조작으로 인해 그 디렉토리가 사용 불가능이면 추가 작업은 실패할 것이고 이럴 경우, 항목은 수동으로 추가되어야 합니다.

사용자 데이터베이스 사용에 관한 다른 고려사항 중에는 RDB 디렉토리 항목 구성이 있습니다. 사용자 데이터베이스의 명명 규칙 중 하나는 사용자 RDB 이름이 네트워크 속성에 지정된 시스템명과 일치할 수 없다는 것입니다(네트워크 속성 표시(DSPNETA) 명령에 표시된 것과 같음).

RDB 디렉토리의 로컬 사용자 데이터베이스 항목은 연관된 데이터베이스가 처음 연결변환될 때 자동으로 추가됩니다. 이는 *IP 프로토콜 유형을 사용하여 LOOPBACK으로 지정된 리모트 위치로 작성됩니다. LOOPBACK은 데이터베이스가 디렉토리과 같은 서버에 있다는 것을 나타냅니다. 서버들 간에 전환되도록 만들어진 사용자 데이터베이스는 각자의 전용 IP 주소를 가지도록 구성할 것을 강력히 추천합니다. 전환 가능한 데이터베이스에 전용 IP 주소가 없는 경우, 데이터베이스가 전환될 때마다 그 데이터베이스를 참조하는 모든 서버에 있는 해당 디렉토리 항목을 수동으로 갱신해야 합니다. 전용 IP 주소를 구성하는 방법에 관한 설명은 iSeries Information Center의 시스템 관리 아래 클러스터 주소에서 관리 어플리케이션 CRG IP 주소 내용을 참조하십시오. 사용자 데이터베이스의 RDB 디렉토리 항목에 관한 자세한 정보는 관계형 데이터베이스 디렉토리 사용을 참조하십시오.

관계형 데이터베이스 디렉토리 사용

OS/400 프로그램은 관계형 데이터베이스 디렉토리를 사용하여 iSeries 서버에서 실행되는 어플리케이션이 액세스하는 관계형 데이터베이스명을 정의하고, 연결할 때 SNA 또는 IP를 사용하는지를 지정하며, 관계형 데이터베이스명과 이에 대응되는 네트워크 매개변수를 연관시킵니다. 관계형 데이터베이스 디렉토리는 어플리케이션 리퀘스터(AR)가 어플리케이션의 관계형 데이터베이스명을 받아들일 수 있도록 하여 이 이름을 해당하는 인터넷 프로토콜(IP) 주소나 호스트명과 포트, 또는 해당하는 시스템 네트워크 구조(SNA) ID와 통신 처리용 논리 장치(LU) 이름값으로 변환할 수 있도록 합니다. 또, V5R2에서 RDB 디렉토리는 사용자의 우선 아웃바운드 연결 보안 메카니즘을 지정하는데도 사용됩니다. 또한, 관계형 데이터베이스 디렉토리는 ARD 프로그램을 관계형 데이터베이스명과 연관시킬 수 있도록 합니다.

분산 관계형 데이터베이스(RDB) 네트워크에 있는 각 iSeries 시스템은 관계형 데이터베이스 디렉토리가 구성 되도록 해야 합니다. 한 시스템에는 오직 하나의 관계형 데이터베이스 디렉토리가 존재합니다. 분산 관계형 데이터베이스(RDB) 네트워크에서 각 AR은 이 로컬 관계형 데이터베이스에 대해서 이것의 관계형 데이터베이스 디렉토리에 입력을 갖고 있어야 하며, AR이 액세스하는 각 리모트 관계형 데이터베이스 및 로컬 사용자 관계형 데이터베이스에 대해서도 마찬가지입니다. 어플리케이션 서버(AS)로만 활동하는 분산 관계형 데이터베이스(RDB) 네트워크의 시스템은 해당 디렉토리에 다른 리모트 관계형 데이터베이스의 관계형 데이터베이스명을 포함할 필요는 없습니다.

로컬 관계형 데이터베이스에 할당된 관계형 데이터베이스명은 유일한 것이어야 합니다. 다시 말해, 그것은 네트워크 내의 다른 관계형 데이터베이스와 달라야 합니다. 디렉토리에서 다른 관계형 데이터베이스로 할당된 이름은 리모트 관계형 데이터베이스 또는 로컬 사용자 데이터베이스를 식별합니다. 리모트 RDB의 이름은 AS가 로컬 시스템 데이터베이스 또는 구성이 되어 있는 경우 그 사용자 데이터베이스 중 하나를 식별하는데 사용하는 이름과 일치해야 합니다. AS에서 로컬 시스템 RDB명 항목이 요구 시에 없으면, 자동으로 이 항목 하나가 디렉토리 안에 작성됩니다. 사용되는 이름은 네트워크 속성 표시(DSPNETA) 명령에 의해 표시된 현재의 시스템명이 됩니다.

자세한 정보는 다음 주제를 참조하십시오.

- 관계형 데이터베이스 디렉토리에 대한 작업
- 관계형 데이터베이스 디렉토리 설정 예

관계형 데이터베이스 디렉토리에 대한 작업

다음의 명령은 관계형 데이터베이스 디렉토리에 대한 작업을 실행시킵니다.

ADDRDBDIRE

관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령

CHGRDBDIRE

관계형 데이터베이스 디렉토리 항목 변경(CHGRDBDIRE) 명령

DSPRDBDIRE

관계형 데이터베이스 디렉토리 항목 표시(DSPRDBDIRE) 명령

RMVRDBDIRE

관계형 데이터베이스 디렉토리 항목 제거(RMVRDBDIRE) 명령

WRKRDBDIRE

관계형 데이터베이스 디렉토리 항목에 대한 작업(WRKRDBDIRE) 명령

SNA 사용법을 위한 항목 추가

ADDRDBDIRE(RDB 디렉토리 항목 추가) 화면이 아래에 나와 있습니다. 이 화면에서 프롬프트를 사용하거나 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령을 사용하여 관계형 데이터베이스 디렉토리에 항목을 추가할 수 있습니다.

ADDRDBDIRE (REC 디렉토리 항목 추가)

선택사항을 입력하고 Enter를 누르십시오.

관계형 데이터베이스	MP311	이름
관계형 데이터베이스 별명	*NONE	
리모트 위치:		
주소명	MP311	이름, *LOCAL, *ARDPGM
유형	*SNA	*SNA, *IP
텍스트	'Oak Street	판매 대리점'

이 예에서, 리모트 위치명 MP311을 갖고 있는 서버에 대하여 MP311라는 이름의 관계형 데이터베이스를 로컬 서버의 관계형 데이터베이스 디렉토리에 추가하도록 입력되어 있습니다. SNA 연결의 경우 관계형 데이터베이스 별명 필드는 디폴트 값 *NONE으로 남아 있어야 합니다. 로컬 위치명은 이것을 사용하여 관계형 데이터베이스 디렉토리 항목이 작성되기 전에 정의될 필요는 없습니다. 그러나, 리모트 위치명은 관계형 데이터베이스 디렉토리 항목이 어플리케이션에서 사용되기 전에 정의되어야 합니다. 관계형 데이터베이스(RDB)명 매개변수와 리모트 위치명(RMTLOCNAME) 매개변수는 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령에 요구됩니다. RMTLOCNAME 매개변수의 두 번째 요소는 *SNA에 디폴트 값으로 됩니다. 설명적인 텍스트(TEXT) 매개변수 선택형입니다. 이 예에서 보는 바와 같이, 관계형 데이터베이스명을 사용자의 네트워크 구성에서 이 서버를 위해 지정된 서버명이나 위치명과 같게 만드는 것이 좋습니다. 이것은 데이터베이스명을 식별하도록 사용자를 도와주고, 특히 사용자의 네트워크가 복잡한 경우, 이것을 사용자의 분산 관계형 데이터베이스(RDB) 네트워크에 있는 특정의 서버와 상관되게 할 수 있습니다.

이 명령에 관한 다른 선택형 매개변수를 참조하려면 ADDRDBDIRE(RDB 디렉토리 항목 추가) 화면에서 F10 키를 누르십시오. 이 선택형 매개변수는 아래에 나와 있습니다.

ADDRDBDIRE(REC 디렉토리 항목 추가)

선택사항을 입력하고 Enter를 누르십시오.

```

관계형 데이터베이스 . . . . . MP311
관계형 데이터베이스 별명 . . . *NONE
리모트 위치:
이름 또는 주소 . . . . . MP311
유형 . . . . . *SNA *SNA, *IP
텍스트 . . . . . 'Oak Street 판매 대리점'

장치:
APPC 장치 설명 . . . . . *LOC 이름, *LOC
로컬 위치 . . . . . *LOC 이름, *LOC, *NETATR
리모트 네트워크 ID . . . . . *LOC 이름, *LOC, *NETATR, *NONE
모드 . . . . . *NETATR 이름, *NETATR
트랜잭션 프로그램 . . . . . *DRDA 문자 값, *DRDA
    
```

서버는 추가의 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령 매개변수의 디폴트 *SNA 값을 제공합니다.

- DEV(장치)
- LCLLOCNAME(로컬 위치명)
- RMTNETID(리모트 네트워크 ID)
- MODE(모드)
- TNSPGM(트랜잭션 프로그램)

주:

1. SNA 연결의 경우 관계형 데이터베이스 별명 필드는 디폴트 값 *NONE으로 남아 있어야 합니다.
2. iSeries 서버의 트랜잭션 프로그램명 매개변수는 TNSPGM입니다. SNA에서는 TPN입니다.
3. APPC(advanced program-to-program communications)로 디폴트를 사용하는 경우, 서버는 사용될 장치, 로컬 위치 및 리모트 네트워크 ID를 결정합니다. 네트워크 속성에 정의되는 모드명이 사용되며, 분산 관계형 데이터베이스 구조(DRDA)(DRDA) 지원에 대한 트랜잭션 프로그램명이 사용됩니다.
4. APPN(Advanced Peer-to-Peer Networking)으로 디폴트를 사용하는 경우, 서버는 DEV(장치) 매개변수를 무시하며 로컬 위치명, 리모트 네트워크 ID 및 네트워크 속성에 정의된 모드명을 사용합니다.

관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령에서 이 디폴트값을 변경할 수 있습니다. 예를 들어, DB2 UDB for VM 서버와 통신하기 위해 TNSPGM 매개변수를 변경해야 할 수도 있습니다. DB2 UDB for VM 지원의 디폴트에 의해, TNSPGM은 사용자가 연결하려는 DB2 UDB for VM 데이터베이스명이 됩니다. DRDA(*DRDA)의 디폴트 TNSPGM 매개변수 값은 X'07F6C4C2'입니다. QCNTEDDM과 DB2DRDA도 X'07F6C4C2'로 맵핑됩니다. 트랜잭션 프로그램명에 대한 자세한 정보는 다음을 참조하십시오.

- 188 페이지의 『iSeries용 DB2 UDB 어플리케이션 리퀘스터 상에 TPN으로서 QCNTSRVC 설정』.
- 188 페이지의 『VM 어플리케이션 리퀘스터를 위한 DB2 UDB 상에 TPN으로서 QCNTSRVC 설정』.
- 189 페이지의 『z/OS 어플리케이션 리퀘스터를 위한 DB2 UDB 상에 TPN으로서 QCNTSRVC 설정』.
- 189 페이지의 『DB2 연결 어플리케이션 리퀘스터 상에서 TPN으로서 QCNTSRVC 설정』.

TCP/IP 사용법을 위한 항목 추가

아래에 나와 있는 ADDRDBDIRE(RDB 디렉토리 항목 추가) 화면은 사용자가 *IP를 RMTLOCNAME의 두 번째 요소로 입력하면 패널이 어떻게 변경되는지와 TCP/IP를 사용하는 RDB에 대해 일반적인 입력은 어떤 것과 비슷한지를 나타내고 있습니다. 관계형 데이터베이스 별명 필드를 TCP/IP를 사용하는 연결에 사용할 수 있지만 이 첫 번째 TCP/IP 예는 별명을 지원하지 않습니다.

```

ADDRDBDIRE (REC 디렉토리 항목 추가)

선택사항을 입력하고 Enter를 누르십시오.

관계형 데이터베이스 . . . . . > MP311
관계형 데이터베이스 별명 . . . > *NONE
리모트 위치:
주소명 . . . . . > MP311.spiffy.com

유형 . . . . . > *IP          *SNA, *IP
텍스트 . . . . . > 'Oak Street 판매 대리점'

포트 번호나 서비스 프로그램 . . . > *DRDA
리모트 인증 메소드:
우선되는 방법 . . . . . > *ENCRYPTED *USRID, *USRIDPWD...
하위 인증 허용 . . . > *ALWLOWER *ALWLOWER, *NOALWLOWER
    
```

관계형 데이터베이스 별명 지정

다음 예에서는 RDB 별명을 지정하는 디렉토리 항목을 추가합니다. 그러므로 같은 이름의 관계형 데이터베이스가 있는 네트워크는 DRDA 환경에서 각각의 관계형 데이터베이스를 고유하게 식별하게 할 수 있습니다. 별명을 사용하는 항목이 RDB 디렉토리에 추가되면 항목은 별명으로 식별됩니다. 항목을 표시하거나 삭제하려면 별명 이름을 지정해야 합니다.

아래의 화면은 RDBALS를 관계형 데이터베이스 별명으로 지정한 것입니다.

```

선택사항을 입력하고 Enter를 누르십시오.

관계형 데이터베이스 . . . . . > TEST 문자 값
관계형 데이터베이스 별명 . . . RDBALS
리모트 위치:
이름 또는 주소 . . . . . MP311.spiffy.com

유형 . . . . . *IP          *SNA, *IP
텍스트 . . . . . 'Oak Street 판매 대리점'
    
```

WRKRDBDIRE와 옵션 1을 사용하여 별명에 대한 항목을 추가할 경우 먼저 실제 RDB 이름을 항목 필드에 넣고 Enter를 눌러야 합니다. 그런 다음 관계형 데이터베이스 별명 필드에 별명을 입력하고 다른 필드를 채우면 별명이 RDB 항목 리스트의 항목 필드에 있는 실제 RDB 이름을 대체합니다. 리모트 위치명 유형을 *SNA에서 *IP로 변경해야 합니다.

별명에 대한 항목을 제거하면 별명 항목을 지정하는 데 실제 RDB 이름이 아니라 별명이 사용됩니다.

별명을 통해 리모트 데이터베이스를 식별할 경우 같은 디렉토리의 실제 이름을 통해 리모트 데이터베이스를 참조할 수 없습니다.

RMTLOCNAME에 대해 MP311.spiffy.com을 지정하는 대신에 IP 주소를 지정할 수 있습니다(예를 들어, '9.5.25.176'). IP를 또다른 iSeries 서버에 연결하는 경우, 포트 447을 사용해야만 하는 경우가 아니라면, 디폴트에서 설정된 PORT 매개변수 값, *DRDA를 그대로 두십시오. 예를 들어, IP 보안(IPSec)을 사용한 전송을 위해 포트 447을 구성할 수도 있습니다. 예를 들어, 다른 플랫폼의 IBM Universal Database(UDB) 서버에 연결하는 경우, 포트를 50000과 같이 하나의 숫자로 설정해야 할 수 있습니다. 사용하고 있는 서버에 대한 제품 문서를 참조하십시오. 어떤 위치에서 DRDA 포트에 대해 유효한 서비스명을 정의하는 경우, 숫자 대신에 이것을 사용할 수 있습니다. 하지만 iSeries에서는 *DRDA가 'drda' 서비스명을 사용하는 것보다 우선됩니다.

어플리케이션 리퀘스터 드라이버(ARD)에 항목 추가

관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령 프롬프트에 통신 정보와 ARD 프로그램을 지정하려면 F9 키와 PgDn을 누르십시오. ARD 프로그램이 ADDRDBDIRE 명령에 지정된 통신 정보를 사용하지 않을 경우(일반적인 경우임), RMTLOCNAME 매개변수에 특수값 *ARDPGM을 사용하십시오. ARD 프로그램은 시스템 데이터베이스의 라이브러리에 상주해야 합니다(ASP 번호 1-32).

(WRKRDBDIRE) 명령 사용

RDB 디렉토리 항목 표시에 대한 작업은 사용자가 관계형 데이터베이스 디렉토리 항목을 추가, 변경, 표시 또는 제거할 수 있도록 하는 옵션을 제공합니다.

RDB 디렉토리 항목에 대한 작업

위치 대상

옵션을 입력한 후 Enter를 누르십시오.
 1=추가 2=변경 4=제거 5=세부사항 표시 6=세부사항 인쇄

옵션	관계형 데이터베이스	리모트 위치	텍스트
	KC000	KC000	캔자스 시티 지역 데이터베이스
	MP000	*LOCAL	미니애폴리스 지역 데이터베이스
	MP101	MP101	판매 대리점 데이터베이스 MP101
	MP102	MP102	판매 대리점 데이터베이스 MP102
	MP211	MP211	판매 대리점 데이터베이스 MP211
	MP215	MP215	판매 대리점 데이터베이스 MP215
	MP311	MP311	판매 대리점 데이터베이스 MP311

화면에서 보는 바와 같이, 옵션 4는 로컬 서버의 관계형 데이터베이스 디렉토리로부터 항목을 제거하는 데 사용될 수 있습니다. 항목을 제거하면, 사용자는 지정된 항목에 대해 제거 요구를 확인하거나 다른 관계형 데이터베이스 디렉토리 항목을 선택할 수 있게 해주는 또다른 화면을 수신합니다. 관계형 데이터베이스 디렉토리 제거(RMVRDBDIRE) 명령을 사용하는 경우, 특정의 관계형 데이터베이스명, 총칭명, 모든 디렉토리 항목 또는 리모트 항목만을 지정하는 옵션을 가집니다.

사용자는 항목의 세부사항을 표시하는 RDB 디렉토리 항목 화면에 대한 작업에 관한 옵션을 가집니다. RDB 항목 화면에 대한 작업의 출력은 화면 그대로입니다. 그러나, 관계형 데이터베이스 디렉토리 항목 표시(DSPRDBDIRE) 명령을 사용하는 경우, 관계형 데이터베이스 디렉토리는 iSeries 오브젝트가 아니므로, 출력

파일의 사용은 관계형 데이터베이스 디렉토리에 대하여 백업 수단을 제공합니다. 관계형 데이터베이스 디렉토리 백업에 대하여 출력 파일을 가지고 (DSPRDBDIRE) 명령을 사용하는 방법에 대한 자세한 정보는 138 페이지의 『관계형 데이터베이스 디렉토리 저장 및 복원』을 참조하십시오.

RDB 디렉토리 항목 화면에 대한 작업에 관한 옵션을 갖고 관계형 데이터베이스 디렉토리에 있는 항목을 변경합니다. 또한, 사용자는 관계형 데이터베이스 디렉토리 항목 변경(CHGRDBDIRE) 명령을 사용하여 디렉토리에 있는 항목을 변경할 사용자는 선택형 명령 매개변수와 서버의 리모트 위치명은 무엇이든 변경할 수 있습니다. 디렉토리 항목에 대한 관계형 데이터베이스명을 변경할 수는 없습니다. 디렉토리에서 관계형 데이터베이스명을 변경하려면, 관계형 데이터베이스에 대한 항목을 제거하고 새로운 데이터베이스명에 대한 항목을 추가하십시오.

주: 관계형 데이터베이스 디렉토리 항목에서 리모트 위치가 변경된 경우, 리모트 저널은 리모트 저널 제거 (RMVRMTJRN) 명령 또는 QjoRemoveRemoteJournal API를 사용하여 제거한 다음, 리모트 저널 추가 (ADDRMTJRN) 명령 또는 QjoAddRemoteJournal API를 사용하여 다시 추가해야 합니다. 리모트 위치 유형(또는 인증) 또는 그 외 다른 것이 변경된 경우, 리모트 저널링은 리모트 저널 변경(CHGRMTJRN) 명령 또는 QjoChangeJournalState API를 사용하고 종료하고, 리모트 저널 변경(CHGRMTJRN) 명령 또는 qjoChangeJournalState API를 사용하여 재시작해야 합니다. 분산 파일에 변경이 일어난 경우, 노드 그룹을 삭제하고 다시 작성한 다음, 해당 파일을 다시 작성해야 합니다.

*LOCAL 디렉토리 항목

*LOCAL이 들어 있는 디렉토리 항목은 디렉토리에 그런 항목이 하나뿐이라는 점에서 고유하며 이는 로컬 시스템 데이터베이스의 이름을 지정합니다. 연관된 RDB 이름은 SQL CONNECT문에서 로컬 데이터베이스¹에 연결하는데 사용될 수 있습니다. 이 결과는 CONNECT RESET SQL문을 사용하는 것과 유사하지만 보통은 이런 식으로 사용할 필요가 없습니다.

그러나 로컬 RDB 항목명을 변경해야 하는 경우, 이 프로시저는 이전의 문단에서 설명한 대로 제거와 추가 실행을 포함합니다. 그러나 로컬 입력의 제거에 따르는 특별한 고려사항이 있습니다. 이것은 해당 항목이 전조직에 걸치는 DRDA 속성 정보를 어느 정도 포함하고 있기 때문입니다. 항목을 제거하려고 시도하면, 메시지 CPA3E01(*LOCAL 디렉토리 항목이 구성 자료(CG)의 손실을 야기시킬 수도 있음)을 얻게 되고, 이 작업을 취소하거나 계속할 기회를 얻게 됩니다. 메시지 텍스트는 해당 항목이 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령으로 입력된 구성 자료를 저장하는 *LOCAL 항목이 제거되면 구성 자료가 파손될 수도 있고, 디폴트 구성 값이 실행됩니다. 디폴트 값이 만족스럽지 못하면, 구성 자료는 CHGDDMTCPA 명령으로 재입력되어야 합니다. 입력을 제거하기 전에, 사용자는 *LOCAL 항목이 제거되고 올바른 로컬 RDB명으로 추가된 후에 저장될 수 있도록 CHGDDMTCPA 명령에 지정된 값을 기록하려고 할 수도 있습니다.

로컬 사용자 데이터베이스의 디렉토리 항목

1. 프로그램 테스트 등의 목적으로 DRDA를 로컬 서버 데이터베이스에 연결하려고 한다면 이 용도로 사용할 수 있는 특수 RDB 이름이 두 개 있습니다. 사용법에 관한 예는 프로그래머가 디렉토리 항목을 ME라는 RDB 이름으로, *IP 유형으로 그리고 LOOPBACK이라는 리모트 위치 이름으로 추가하는 것입니다. 그런 다음, 프로그램에서 SQL CONNECT TO ME를 실행하고 로컬 시스템에 소켓 DRDA 연결을 설정합니다. 하지만 RDB 이름을 일반적으로 사용하는 것은 바람직하지 않고 어떤 경우에는 이를 사용하여 뜻밖의 작동이 발생할 수 있다는 것을 경고하기 위한 용도로만 문서에 언급되어 있습니다.

하나의 데이터베이스만 있는 서버에서(예를 들어, 구성된 독립형 보조 기억장치 풀이 없을 때), *LOCAL 항목은 하나의 로컬 데이터베이스를 가리킵니다. 복수의 데이터베이스가 있는 서버에서(하나의 시스템 데이터베이스 및 하나 이상의 사용자 데이터베이스), *LOCAL 항목은 시스템 데이터베이스를 가리킵니다. 로컬 사용자 데이터베이스는 리모트 *IP 항목과 유사한 항목으로 표시됩니다. 가장 큰 차이점은 리모트 위치 필드입니다. 데이터베이스가 다른 서버로 전환될 수 없는 경우에 이 필드는 보통 LOOPBACK이라는 단어를 포함하게 됩니다. LOOPBACK은 호스트 서버의 IP 주소를 나타냅니다. 데이터베이스가 전환되는 경우, 사용자는 특정 IP 주소가 접속하는 서버에 관계없이 데이터베이스와 연관되도록 서버를 구성할 것을 권장합니다. 전용 IP 주소를 구성하는 방법에 관한 설명은 iSeries Information Center의 시스템 관리 아래 클러스터 주소에서 관리 어플리케이션 CRG IP 주소 내용을 참조하십시오. 이 경우, IP 주소는 리모트 위치 필드에서 사용될 것입니다.

전환 가능한 데이터베이스에 LOOPBACK이 사용될 경우, 로컬 서버에서 전환될 때마다 사용자는 수동으로 디렉토리 항목을 변경하여 LOOPBACK을 접속된 새 서버의 IP 주소로 대체하고 데이터베이스가 다시 원래대로 전환될 때는 LOOPBACK으로 다시 변경해야 합니다.

관계형 데이터베이스 디렉토리 설정 예

Spiffy사 네트워크는 관계형 데이터베이스 디렉토리가 분산 관계형 데이터베이스(RDB) 네트워크에서 서버에 사용되는 방법을 설명하고 각 디렉토리를 설정하는 방법을 나타내는 예를 제공합니다. 예에서, TCP/IP와는 대조적인 것으로서 설정은 비슷하게 하되 APPC의 사용을 통신용으로 가정합니다. 그러나, 예의 일부 요소는 프로토콜 의존성입니다. APPC 사용에 필요한 RDB 디렉토리 항목은 TCP/IP 네트워크에도 요구되는 것이지만, 해당 매개변수는 다릅니다. 호스트명이나 IP 주소 및 포트 식별은 LU명, 장치 설명, 모드, TPN 등을 대체합니다.

고려해야 할 간단한 관계는 아래에서 보는 바와 같이 두 지역 오피스간의 관계입니다.

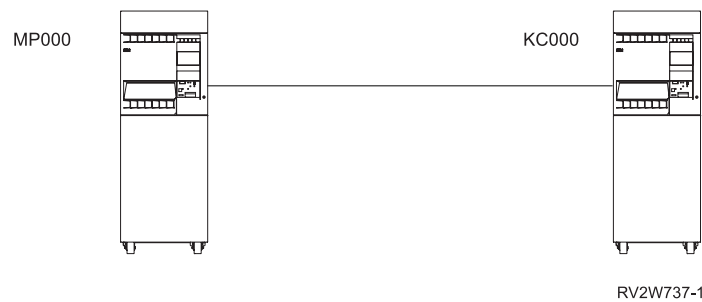


그림 8. 두 서버에 대한 관계형 데이터베이스 디렉토리 설정

각 지역 오피스에 대한 관계형 데이터베이스 디렉토리는 로컬 관계형 데이터베이스에 대한 항목과 리모트 관계형 데이터베이스에 대한 항목을 포함하여야 합니다. 이것은 각 시스템이 어플리케이션 리퀘스터(AR)이고 동시에 어플리케이션 서버(AS)이기 때문입니다. MP000 서버에 대한 관계형 데이터베이스 디렉토리를 작성하는 명령은 다음과 같습니다.

```
ADDRDBDIRE    RDB(MP000) RMTLOCNAME(*LOCAL) TEXT('Minneapolis region database')
ADDRDBDIRE    RDB(KC000) RMTLOCNAME(KC000) TEXT('Kansas City region database')
```

위의 예에서, MP000 서버는 RMTLOCNAME 매개변수에 대해 *LOCAL을 지정함으로써 그 자체를 로컬 관계형 데이터베이스로서 식별합니다. iSeries 서버에는 하나의 관계형 데이터베이스만이 존재합니다. 디렉토리에서 관계형 데이터베이스명을 로컬 서버에 대해서 서버명 및 로컬 위치명과 동일하게 만들고, 리모트 서버에 대해서는 리모트 위치명과 동일하게 만들면, 사용자의 네트워크 관계형 데이터베이스의 식별을 단순화시킬 수 있습니다.

주: 서버명은 네트워크 속성 변경(CHGNETA) 명령의 SYSNAME 매개변수에 지정되어 있습니다. 로컬 서버는 통신 구성 중에 CHGNETA 명령의 LCLLOCNAME 매개변수에서 식별됩니다. SNA(APPC)를 사용하는 리모트 위치는 통신 구성 중에 제어기 설명((APPC) 작성(CRTCTLAPPC)의 RMTCPNAME 매개변수로 식별됩니다. 이 명령을 사용하는 방법에 관한 예는 예: 분산 관계형 데이터베이스를 위한 APPN 구성을 참조하십시오. 서버명으로 같은 이름을 사용하면, 특히 복잡한 네트워크에서 네트워크 위치와 데이터베이스명의 혼동을 피할 수 있습니다.

KC000 서버 관계형 데이터베이스 디렉토리에 대한 통신 관계의 항목은 다음과 같습니다.

```
ADDRDBDIRE RDB(KC000) RMTLOCNAME(*LOCAL) TEXT('Kansas City region database')
```

```
ADDRDBDIRE RDB(MP000) RMTLOCNAME(MP000) TEXT('Minneapolis region database')
```

이보다 더 복잡한 것으로서 고려해야 할 예는 지역의 오피스에서 이것의 판매 대리점 간의 관계입니다. 예를 들어, 아래에서 보는 바와 같이 네트워크에서 관계형 데이터베이스에 액세스하려면, MP000 서버의 관계형 데이터베이스 디렉토리가 해당 판매 대리점 중 각각에 대한 항목을 포함하도록 확장되어야 합니다.

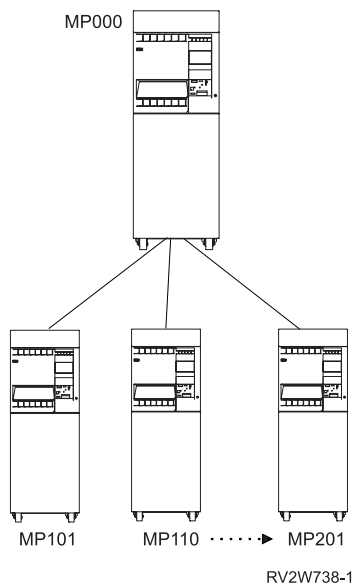


그림 9. 복수 서버에 대한 관계형 데이터베이스 디렉토리 설정

판매 대리점 데이터베이스 전체를 포함하도록 MP000 관계형 데이터베이스 디렉토리를 완료하는 데 사용되는 명령의 샘플은 다음과 같습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

PGM
ADDRDBDIRE   RDB(MP000) RMTLOCNAME(*LOCAL) +
TEXT('Minneapolis region database')
ADDRDBDIRE   RDB(KC000) RMTLOCNAME(KC000)
TEXT('Kansas City region database')
ADDRDBDIRE   RDB(MP101) RMTLOCNAME(MP101)
TEXT('Dealer database MP101')
ADDRDBDIRE   RDB(MP002) RMTLOCNAME(MP110)
TEXT('Dealer database MP110')
.
.
.
ADDRDBDIRE   RDB(MP215) RMTLOCNAME(MP201)
TEXT('Dealer database MP201')
ENDPGM

```

위의 예에서, 각 지역 판매 대리점은 리모트 관계형 데이터베이스와 같이 미니애폴리스 관계형 데이터베이스 디렉토리 안에 포함됩니다.

각 판매 대리점은 MP000과 다른 판매 대리점 어플리케이션 서버에 대한 AR로서 사용될 수 있기 때문에, 각 판매 대리점은 로컬 관계형 데이터베이스와 지역 오피스로서 그 자체에 대한 항목과 리모트 관계형 데이터베이스로서 다른 모든 판매자에 대한 항목이 있는 관계형 데이터베이스 디렉토리를 갖고 있어야 합니다. 데이터베이스 관리자는 각 판매 대리점 서버에서 관계형 데이터베이스 디렉토리를 작성하는 여러 옵션을 갖고 있습니다.

대부분의 시간을 사용하고 오류가 생기기 쉬운 메소드는 MP000 분산 관계형 데이터베이스(RDB) 네트워크의 일부인 모든 서버에 각 디렉토리 항목을 작성하는 데 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령을 사용하여 각 서버에서 관계형 데이터베이스 디렉토리를 작성하는 것입니다.

이보다 나은 대안은 MP000에 대한 위의 예에서 보여주는 것과 같이 CL(제어 언어) 프로그램을 작성하는 것입니다. 분산 관계형 데이터베이스(RDB) 관리자는 판매 대리점 서버 중 각각에 대하여 이 CL 프로그램을 복사할 수 있습니다. 각 판매 대리점에 대해 이 프로그램을 사용자 정의하기 위해 데이터베이스 관리자는 MP000 서버의 리모트 위치명을 MP000으로 변경하고, 로컬 판매 대리점의 리모트 위치명을 *LOCAL로 변경합니다. 분산 관계형 데이터베이스(RDB) 관리자는 사용자 정의된 CL 프로그램을 각 판매 대리점으로 분배하여 고유한 관계형 데이터베이스 디렉토리를 빌드하도록 해당 서버에서 실행될 수 있습니다.

세 번째 메소드는 관계형 데이터베이스 디렉토리 항목 표시(DSPRDBDIRE) 명령을 사용한 결과로서 출력 파일에 송신된 관계형 데이터베이스 디렉토리를 읽는 프로그램을 기록하는 것입니다. 이 프로그램은 MP000 서버에 대한 관계형 데이터베이스 디렉토리 항목을 포함하는 출력 파일과 함께 판매 대리점으로 분배될 수 있습니다. 각 서버는 MP000 출력 파일을 읽어 로컬 관계형 데이터베이스 디렉토리를 작성할 수도 있습니다. 이 때, 관계형 데이터베이스 디렉토리 항목 변경(CHGRDBDIRE) 명령은 로컬 서버에 대한 MP000 서버 디렉토리를 사용자 정의하는 출력 파일을 사용하여 관계형 데이터베이스 디렉토리 항목 작성에 대한 자세한 정보는 138 페이지의 『관계형 데이터베이스 디렉토리 저장 및 복원』을 참조하십시오.

DRDA 보안 설정

분산 관계형 데이터베이스 구조™(DRDA) 보안은 41 페이지의 제 4 장 『iSeries 분산 관계형 데이터베이스에 대한 보안』에서 다루고 있으나 완벽을 기하기 위해 여기서는 DRDA를 사용하기 전 또는 APPC 사용에서 TCP/IP로 사용자 네트워크를 변환할 때 고려해야 할 사항을 설명합니다.

TCP/IP에 대한 보안 설정은 APPC에 요구되는 것과는 상당한 차이가 있습니다. 단 하나 알고 있는 차이는 APPC가 갖고 있는 ‘보안 위치’ 개념이 없다는 것입니다. TCP/IP 서버는 클라이언트 서버가 말하는 그대로를 완전하게 믿을 수 없기 때문에, 연결 요구 시 암호 사용은 더욱 중요합니다. 연결 요구 시 암호 송신을 보다 더 쉽게 할 수 있도록, 특정의 사용자 프로파일과 연관된 서버 권한 리스트 사용에 관해 TCP/IP 지원과 더불어 이미 소개하였습니다. 서버 권한 부여 리스트 항목은 xxxSVRAUTHE 명령(xxx는 ADD, CHG 및 RMV)을 사용하여 유지보수됩니다. xxxSVRAUTHE 명령에 대한 설명은 41 페이지의 제 4 장 『iSeries 분산 관계형 데이터베이스에 대한 보안』 및 iSeries Information Center의 제어 언어(CL) 주제를 참조하십시오. 서버 권한 항목의 사용에 대한 대안은 SQL CONNECT 명령문의 USER/USING 양식을 사용하여 연결 요구 시 암호를 송신하는 것입니다.

V5R2에서 Kerberos 지원이 추가되었고 사용자가 TCP/IP를 사용하는 경우, 다른 보안 옵션을 제공합니다. Kerberos를 위한 구성 방법에 관한 정보는 iSeries Information Center의 네트워크 인증 서비스 주제를 참조하십시오.

서버측에서의 설정은 인바운드 연결 요구에 대해 어떤 레벨의 보안이 필요한지를 결정하고 지정하는 것을 포함합니다. 예를 들어, 암호화되지 않은 암호를 받아들여야 합니까? 디폴트 설정은 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령의 사용에 의해 변경될 수 있습니다.

DRDA용 TCP/IP 서버 설정

TCP/IP 프로토콜을 사용하고 있는 DRDA 어플리케이션 서버(AS)를 소유하는 경우, DDM TCP/IP 서버를 설정해야 합니다. 이것은 요구 시에 시작된다는 것을 보증하는 것 만큼이나 간단한 일이 될 수 있으며, 사용자가 언제나 활성 상태로 이를 유지하기를 원하는 경우, 다음의 명령을 실행함으로써 실행될 수 있습니다.

```
CHGDDMTCPA AUTOSTART(*YES)
```

그러나, 사용자의 환경에 대해 서버를 동조시키기 위해 조정하려 할 수 있는 다른 매개변수가 존재합니다. 이것은 초기의 사전시작 작업의 수, 작업의 최대수 및 더 많은 작업을 시작할 때에 임계값 등을 포함합니다. 이 주제에 대한 자세한 정보는 113 페이지의 『TCP/IP 서버 관리』를 참조하십시오.

사용자는 연결할 때 모든 클라이언트가 사용하는 공통된 사용자 프로파일, 또는 다른 클래스의 리모트 사용자에게 서로 다른 레벨의 보안을 가지는 서로 다른 사용자 프로파일 세트를 설정하려 할 수도 있습니다. 이 때, 사용자는 어플리케이션 리퀘스터(AR)에서 서버 인증 항목 추가(ADDSVRAUTE) 명령을 사용하여 AS에서 그 소속하에 실행할 사용자 프로파일에 대하여 AR에서 각 사용자의 프로파일명을 맵핑할 수 있습니다. 자세한 정보는 인증 방법 협상 주제를 참조하십시오.

대화식 SQL(ISQL)용 SQL 패키지 설정

이 섹션은 비iSeries 어플리케이션 서버에만 적용됩니다.

다음 중 하나가 참인 경우, SQL 패키지가 서버에서 작성되었다는 것을 확인해야 합니다.

- DB2 UDB 조회 관리자 및 SQL 개발 킷을 가지고 있고 제품의 대화식 SQL(STRSQL) 기능을 사용하고 하는 경우
- V5R1 이전의 iSeries 클라이언트에서 TCP/IP를 사용하는 비iSeries DRDA 서버 또는 2단계 확약 기능이 없는 서버에 연결하려는 경우

STRSQL iSeries 서버 SQL 패키지를 요구하지 않습니다. 일반적으로 SQL 패키지는 STRSQL의 사용자에게 대해 비iSeries 어플리케이션 서버(AS)에서 자동으로 작성됩니다. 그러나, STRSQL에 대한 초기 연결이 로컬 서버와 이루어지고, 이 연결이 2상의 확약 프로토콜에 의해 보호되므로 문제가 발생할 수 있습니다. 만일 그 다음의 연결이 단일 위상의 확약 기능을 하는 서버에 대해 이루어지거나 TCP/IP가 이전 V5R1 iSeries 클라이언트로부터 사용된다면 이 때 이 연결은 읽기 전용이 됩니다. 그 연결을 통해 자동으로 패키지를 작성하려고 시도하면 실패하게 됩니다. 이것은 패키지 작성은 갱신으로 간주되며 읽기 전용 연결을 통해 실행될 수 없기 때문입니다.

이에 대한 솔루션은 리모트 AS로 연결하기 전에 로컬 데이터베이스로 연결을 없애는 것입니다. 이것은 COMMIT에 의해 이어지는 RELEASE ALL 명령을 실행함으로써 할 수 있습니다. 이 때 리모트 서버에 연결될 수 있고, 이것이 첫 번째 연결이 되므로 이를 통해 갱신이 이루어질 수 있습니다.

대화식 SQL을 시작할 때, 사용자는 *NONE 이외의 것에 관한 확약 제어 레벨을 지정해야 합니다. 또한, 사용자가 연결에 사용하는 사용자 ID는 어플리케이션 서버에서 SQL 패키지를 작성하기 위한 적당한 권한을 갖고 있어야 합니다. 연결 시도에서 SQLSTATE of 42501을 수신하면, 사용자는 패키지 작성 권한을 가질 수 없습니다.

자세한 정보는 256 페이지의 『서로 다른 어플리케이션 서버에 대화식 SQL 및 조회 관리자 설정』을 참조하십시오.

DDM 파일 설정

iSeries 서버에 대한 DRDA 지원 구현에는 통신용 분산 자료 관리(DDM) 대화를 사용합니다. 이 때문에, 사용자는 분산 관계형 데이터베이스(RDB) 처리와 연계하여 DDM을 사용할 수 있습니다. 사용자는 DDM을 사용하여 리모트 명령을 어플리케이션 서버(AS)에 전달하고, iSeries 서버로부터 또다른 서버로 표를 복사하고, 또다른 서버에서 비분산 관계형 데이터베이스(RDB) 작업을 처리할 수 있습니다.

분산 관계형 데이터베이스(RDB)를 사용하면 어플리케이션 리퀘스터(AR)가 데이터베이스에 연결하는 데 필요한 정보가 관계형 데이터베이스 디렉토리에 제공됩니다. DDM을 사용할 때 어플리케이션 서버(AS)에서 작업 대상이 되는 각 파일에 대해서 별도의 DDM 파일을 작성하여야 합니다. DDM 파일은 어플리케이션 리퀘스터(AR)에서 어플리케이션에 의해 사용됨으로써 어플리케이션 서버(AS)의 리모트 파일과 어플리케이션 서버(AS)로의 통신 경로를 식별합니다.

또, V5R2에서는 RDB 디렉토리 항목에 대한 참조를 가진 DDM 파일을 작성할 수도 있습니다. 제 6 장 『분산 관계형 데이터베이스(DRDB) 관리 및 조작 TASK』에서 다루는 일부 데이터베이스 관리 TASK는 DDM을 사용하여 리모트 파일에 액세스합니다. DDM 파일은 분산 자료 관리 파일 작성(CRTDDMF) 명령을 사용하여 작성됩니다. 파일과 파일 안에 명명된 통신 경로를 작성하기 전에 DDM 파일을 작성할 수 있습니다. 그러나, DDM 파일 안에 명명된 파일과 통신 정보는 DDM 파일이 어플리케이션에 의해 사용되기 전에 작성되어야 합니다.

다음 예는 DDM 파일이 작성되는 한 방법을 보여줍니다.

```
CRTDDMF FILE (TEST/KC105TST) RMTLOCNAME(KC105)
          RMTFILE(SPIFFY/INVENT)
```

이 예에서 DDM 파일 액세스가 TCP/IP를 통해 이루어지려면, RMTLOCNAME 매개변수의 두 번째 요소에 *IP를 지정하여야 합니다.

이 명령은 KC105TST라는 이름의 DDM 파일을 작성하고 이것을 어플리케이션 리퀘스터(AR)의 TEST 라이브러리에 저장합니다. 이 DDM 파일은 리모트 위치 KC105를 사용하여 목표 iSeries 서버의 SPIFFY 라이브러리 안에 저장된 INVENT라는 이름의 리모트 파일에 액세스합니다.

사용자는 DDM 파일에 대한 작업 화면에서 옵션을 사용하여 DDM 파일을 변경, 삭제, 표시 또는 작성할 수 있습니다.

DDM 파일에 대한 자세한 내용은 iSeries Information Center의 분산 자료 관리 주제를 참조하십시오.

분산 관계형 데이터베이스에서 자료를 표에 로딩

분산 관계형 데이터베이스(RDB) 환경에서 어플리케이션은 표에 저장된 자료에 대해 조작합니다. 일반적으로 어플리케이션은 정보의 표를 조회하거나, 하나 또는 여러 개의 표에서 행을 삽입, 갱신, 삭제하거나 또는 새로운 표를 작성하는 데에 사용됩니다. 서버의 자료를 또다른 서버로 이동시켜야 할 때에는 다른 상황이 발생합니다.

이 섹션은 다음 TASK에서 사용할 수 있는 많은 방법에 대해 논의합니다.

- 신규 자료를 분산 관계형 데이터베이스 표에 로딩
- iSeries 서버 간의 자료 이동
- 비iSeries 서버에서 iSeries 서버로 데이터베이스 이동

신규 자료를 분산 관계형 데이터베이스 표에 로딩

사용자는 각 자료 항목을 표에 입력함으로써 자료를 표 안에 로드합니다. iSeries 서버에서 사용자는 SQL, DB2 UDB for iSeries 조회 관리 기능 또는 iSeries 어플리케이션 개발 툴의 자료 파일 유틸리티 부분을 사용하여 자료를 표에 입력하는 어플리케이션을 작성할 수 있습니다.

자료를 SQL을 사용한 표에 로딩

자료를 표에 로드하는 간단한 메소드는 SQL 어플리케이션과 SQL INSERT 작업을 사용하는 것입니다.

지역 센터에서 판매 대리점으로 정기적인 재고 출하를 시행하므로, Spiffy 지역 센터에서는 재고 항목을 주기적으로 판매 대리점의 재고표로 추가해야 하는 상황을 생각해 보십시오.

```
INSERT INTO SPIFFY.INVENT
(PART, DESC, QTY, PRICE)
VALUES
('1234567', 'LUG NUT', 25, 1.15 )
```

위의 명령문은 한 행의 자료를 SPIFFY라는 이름의 SQL 컬렉션에 있는 INVENT라고 불리는 표 안에 삽입합니다.

정규의 출하에 대한 각 항목에서, SQL INSERT 명령문은 판매 대리점의 재고표 안에 한 행을 배치합니다. 위의 예에서, 서로 다른 15개의 항목이 판매 대리점으로 출하되면, 지역 오피스의 어플리케이션은 호스트 변수를 사용하여 15개의 SQL INSERT 명령문 또는 하나의 SQL INSERT 명령문을 포함할 수 있습니다.

이 예에서, 지역 센터는 어플리케이션 서버(AS)에서 자료를 표에 로드하는 데 SQL 어플리케이션을 사용하고 있습니다. SQL에 대한 실행 시간 지원은 OS/400 사용권 프로그램에 제공되어 있습니다. 따라서 AS는 IBM iSeries용 DB2 조회 관리자 및 SQL 개발 킷 사용권 프로그램을 요구하지 않습니다. 그러나, IBM iSeries용 DB2 조회 관리자 및 SQL 개발 킷 사용권 프로그램은 어플리케이션을 기록하는 데 요구됩니다. SQL 프로그래밍 언어에 대한 자세한 정보는 SQL 프로그래밍 개념 및 SQL 참조서 서적을 참조하십시오.

iSeries 조회 관리 기능을 사용한 표 및 파일의 자료 조작

OS/400 사용권 프로그램은 사용자가 표와 파일에 있는 자료를 조작할 수 있도록 하는 DB2 UDB for iSeries 조회 관리 기능을 제공합니다. 조회는 SQL 조회 명령문을 사용하여 작성됩니다. 사용자는 CL 명령이나 사용자의 어플리케이션 프로그램에서 조회 호출가능 인터페이스를 통해 조회를 실행할 수 있습니다. 사용자는 조회 관리 기능을 사용하여 다음과 같이 이전의 섹션에서 설명한 재고 갱신에 관한 표에 한 행의 자료를 삽입할 수 있습니다.

소스 실제 파일 INVLOAD와 SQL 명령문에 소스 멤버 INVLOAD를 작성하십시오.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```
INSERT INTO SPIFFY/INVENT
(PART, DESC, QTY, PRICE)
VALUES
(&PARTVALUE, &DESCVALUE, &QTYVALUE, &PRICEVALUE)
```

CL 명령을 사용하여 조회 관리 조회 오브젝트를 작성하십시오.

```
CRTQMQR Y QMQR(INVLOAD) SRCFILE(INVLOAD) SRCMBR(INVLOAD)
```

다음의 CL 명령은 INSERT SQL 명령문의 결과를 SPIFFY 컬렉션의 INVENT 표 안에 배치합니다. 조회에서 변수((&PARTVALUE, &DESCVALUE 등)를 사용하면, 사용자가 각 행에 대해 다시 조회 관리 조회를 작성하도록 요구하지 않고 STRQMQR 호출의 일부로 원하는 값을 입력할 수 있게 됩니다.

```
STRQMQR Y QMQR(INVLOAD) RDB(KC000)
SETVAR((PARTVALUE '1134567') (DESCVALUE '''Lug Nut''')
(QTYVALUE 25) (PRICEVALUE 1.15))
```

조회 관리 기능은 동적이고, 이것은 프로그램이 컴파일되는 시간이 아니라 실행 시간에 해당 기능의 액세스 경로가 빌드된다는 의미입니다. 이 이유로 DB2 UDB for iSeries 조회 관리 기능은 자료를 표에 로드하는 데 있어서 SQL 어플리케이션만큼 효율적이지 못합니다. 그러나, 어플리케이션을 기록하는 데는 IBM iSeries용 DB2 조회 관리자 및 SQL 개발 킷 제품이 필요합니다. 또한 SQL에 대한 실행 시간 지원과 조회 관리는 OS/400 사용권 프로그램의 일부입니다.

조회 관리 기능에 대한 자세한 내용은 *Query Management Programming* 서적을 참조하십시오.

자료 파일 유틸리티를 사용한 자료 입력, 표 갱신 및 조회 작성

자료 파일 유틸리티(DFU)는 IBM에서 제공하는 iSeries 어플리케이션 개발 툴(ADT) 패키지로서 사용자가 자료를 입력하고 표를 갱신하고 조회를 만드는 프로그램을 작성하도록 하는 프로그램 빌더입니다. DFU를 사용하는 데 프로그래밍 언어가 필요하지 않습니다. 사용자의 자료 입력, 유지보수 또는 조회 프로그램은 사용자가 화면 시리즈에 대해 응답할 때 작성됩니다. DFU 사용 상의 장점은 이것의 총체적인 특성의 활용을 통해 사용자가 SQL과 같이 프로그래밍 언어를 사용함으로써 가능한 한 빨리 자료를 표에 로드하는 데이터베이스 갱신 프로그램을 작성할 수 있다는 것입니다. 사용자는 DDM 파일을 갖고 있는 DFU를 사용하거나 표시장치 pass-through를 사용하여 어플리케이션 소스(AS)에서 DFU를 실행함으로써 리모트 서버에서 자료에 대한 작업을 할 수 있습니다.

DFU 프로그램 생성기에 관한 자세한 정보는 V5R1 보충 매뉴얼 웹 사이트의 *ADTS/400: Data File Utility*



매뉴얼을 참조하십시오.

iSeries 서버 간의 자료 이동

하나의 iSeries 서버로부터 또다른 시스템으로 자료를 이동시키도록 요구하는 기업망 조작에서 수많은 상황이 발생합니다. 예를 들어, 어떤 지역에서 신규 판매 대리점이 문을 열 수도 있고, 다른 판매 대리점 하나 또는 두 곳에서 일부 클라이언트가 클라이언트 주소에 의해 판별되어 신규 판매 대리점으로 이전될 수도 있습니다. 아마도 판매 대리점이 문을 닫거나 더이상 Spiffy사의 판매 및 서비스를 대행하지 않을지도 모릅니다. 이 판매 대리점의 재고 정보 및 서비스 정보가 지역 사무실이나 다른 지역 판매대리점으로 할당되어야 합니다. 아마도 판매 대리점이 서버를 업그레이드하는 데 필요한 한도까지 성장했고, 전체의 데이터베이스가 새로운 서버로 이동되어야 할지도 모릅니다.

하나의 iSeries 서버로부터 또다른 서버로 자료를 이동시키기 위한 몇 가지 대안은 다음과 같습니다.

- 사용자 작성 어플리케이션 프로그램
- 대화식 SQL(ISQL)
- DB2 UDB for iSeries 조회 관리 기능
- 테이프나 디스켓 장치로부터 및 테이프나 디스켓 장치로 복사
- DDM으로 파일 명령 복사
- 네트워크 파일 명령
- iSeries 서버 저장 및 복원 명령

사용자 작성 어플리케이션 프로그램 작성

DUW 연결 관리로 컴파일된 프로그램은 리모트 데이터베이스와 로컬 데이터베이스에 연결하여 하나로부터 불러오기(FETCH)하여 다른 하나에 끼워넣기(INSERT) 식으로 자료를 이동시킬 수 있습니다. 복수 행 FETCH와 복수 행 INSERT를 사용함으로써 레코드 블록을 한 번에 처리할 수 있습니다. 확약 제어를 사용하면 자료 이동 중에 검사 항목이 철저하게 실행될 수 있으며 장애 발생 시 복사 시작을 피할 수 있습니다.

대화식 SQL을 사용한 데이터베이스 조회

SQL SELECT 명령문과 대화식 SQL을 사용하여 로컬 서버에서 표를 작성하거나 갱신하는 데 필요한 자료에 대해 또다른 iSeries 서버의 데이터베이스를 조회할 수 있습니다. SELECT 명령문은 사용자가 표 이름과 원하는 자료를 포함하는 열 및 검색하는 자료의 행을 판별하는 선택 범주나 필터를 지정할 수 있게 합니다. SELECT 명령문이 통과되면, 지정된 표의 하나 또는 그 이상의 행이 결과로 생깁니다.

하나의 표에서 자료를 얻는 데 더하여, SQL은 사용자가 결합 조작을 사용하여 같은 데이터베이스에 들어 있는 두 개 또는 그 이상의 표에 포함된 열에서 정보를 얻도록 해줍니다. SELECT 명령문이 통과되면, 지정된 표의 하나 또는 그 이상의 행이 결과로 생깁니다. 리턴된 행렬에 위치한 자료 값은 지정된 표 안에 포함된 합성 자료 값을 나타냅니다.

대화식 SQL 조회를 사용하면, 조회 결과는 로컬 서버의 데이터베이스 파일 안에 배치될 수 있습니다. 대화식 SQL 프로세스에 대해서 확약 제어 레벨이 지정되는 경우, 그것은 어플리케이션 서버(AS)에 적용되며 또한 로컬 서버의 데이터베이스 파일은 *NONE이라는 확약 제어 레벨하에 있게 됩니다.

대화식 SQL에 의해 사용자는 다음과 같이 행할 수 있습니다.

- 선택 결과에 대해 새로운 파일 작성.
- 대체 및 기존의 파일.
- 파일에 신규 멤버 작성.
- 멤버 대체.
- 결과를 기존 멤버에 추가.

KC105 판매 대리점이 자체에서 전체의 재고 부품번호 '1234567'을 KC110으로 전송하려는 상황을 고려해 보십시오. KC110은 KC105로부터 확보한 부품에 대한 KC105 데이터베이스를 조회합니다. 이 재고 조회의 결과는 KC110 서버에 이미 존재하는 데이터베이스 파일로 리턴됩니다. 이것은 이 작업을 완료하기 위해 사용자가 사용할 수 있는 프로세스입니다.

STRSQL(SQL 시작) 명령을 사용하여 대화식 SQL 화면을 받으십시오. 신규 데이터베이스에 대해 SQL 명령문(CONNECT는 제외) 하나를 입력하기 전에 이 조작의 결과가 다음의 단계를 실행함으로써 로컬 서버의 데이터베이스 파일에 송신되도록 지정하십시오.

1. SQL 명령문 입력 화면으로부터 서비스 옵션을 선택하십시오.
2. 서비스 화면으로부터 세션 속성 변경 옵션을 선택하십시오.
3. 세션 속성 화면으로부터 출력장치 선택 옵션을 입력하십시오.

4. 출력 장치 필드에 있는 데이터베이스 파일에 대해 3을 하나 입력하고 Enter를 누르십시오. 다음 화면이 나타납니다.

파일 변경

선택사항을 입력하고 Enter를 누르십시오.

파일	QSQLSELECT	이름	
라이브러리	QGPL	이름	
멤버	*FILE	이름, *FILE, *FIRST	

옵션	1		1=새로운 파일 작성
			2=파일 대체
			3=신규 멤버 작성
			4=멤버 대체
			5=멤버 추가

새로운 파일에서:

권한	*LIBCRTAUT	*LIBCRTAUT, *CHANGE, *ALL	
		*EXCLUDE, *USE	

권한 부여 리스트명.

텍스트

F3=종료 F5=화면정리 F12=취소

5. 결과를 수신 받은 데이터베이스 파일명을 지정하십시오.

데이터베이스명이 지정되어 있을 때, 아래의 예에서 보는 바와 같이 사용자의 대화식 SQL 처리를 시작할 수 있습니다.

SQL 명령문 입력

SQL 명령문을 입력하고 Enter를 누르십시오.

현재의 연결 지점은 관계형 데이터베이스 KC000입니다.

CONNECT TO KC105

현재의 연결 지점은 관계형 데이터베이스 KC105입니다.

====> SELECT * FROM INVENTORY

WHERE PART = '1234567'

맨 아래

F3=종료	F4=프롬트	F6=행 삽입	F9=검색	F10=행 복사
F12=취소		F13=서비스	F24=추가 키	

SQL 프로그래밍 언어 및 대화식 SQL에 대한 자세한 정보는 iSeries Information Center의 SQL 프로그래밍 개념 및 SQL 참조서 서적을 참조하십시오.

DB2 UDB for iSeries 조회 관리 기능을 사용한 리모트 서버 조회

DB2 UDB for iSeries 조회 관리 기능은 리모트 서버를 조회하고 로컬 서버로 출력 파일에 있는 결과를 리턴하기 위해 대화식 SQL과 같은 지원을 제공합니다.

대화식 SQL과 조회 관리 기능 모두 표(또는 파일)가 이미 컬렉션에 존재 해야 한다는 요구없이도 파일이나 표에 대해 자료의 조작 작업(INSERT, DELETE, SELECT 등)을 실행할 수 있습니다. 또한 조회 관리는 SQL CREATE TABLE 명령을 사용하여 새로운 표가 조회의 결과로서 서버에 작성될 때 자료 정의를 제공합니다. 조회 관리 기능으로부터 작성된 표는 SQL을 사용하여 작성된 표에 적용하는 동일한 지침과 제한사항을 따릅니다.

그러나, 조회 관리 기능은 사용자가 파일이나 표에 결과를 추가하려고 할 때 멤버를 지정하지 못하도록 합니다. 조회 기능의 결과는 사용자가 조회 관리 기능을 시작하기 전에 데이터베이스 파일 대체(DVRDBF) 명령을 사용하여 다른 멤버를 지정하지 않는 한, 첫 번째 파일 멤버에 위치합니다.

조회 관리 기능에 대한 자세한 내용은 Query Management Programming 서적을 참조하십시오.

테이프 또는 디스켓 간의 파일 복사

사용자는 iSeries 서버에서 테이프를 복사(CPYTOTAP) 및 디스켓으로 복사(CPYTDDKT) 명령을 사용하여 표나 파일을 테이프나 디스켓으로 복사할 수 있습니다.

테이프나 디스켓의 자료는 테이프로부터 복사(CPYFRMTAP) 및 디스켓으로부터 복사(CPYFRMDKT) 명령을 사용하여 또다른 서버에 로드될 수 있습니다. 이들 명령 사용에 대한 자세한 정보는 기억장치 솔루션 주제를 참조하십시오.

또한 파일 복사(CPYF) 명령을 사용하여 테이프의 자료를 iSeries용 DB2 UDB 안에 로드할 수 있습니다. 이것은 특히 DB2 UDB for z/OS 또는 DB2 UDB Server for VM(SQL/DS™)으로부터 언로드되었던 자료를 로드할 때 유용합니다. 널(null)이 가능한 자료는 단일 바이트 플래그가 각 널(null) 가능한 필드와 연관될 수 있도록 하는 방법으로 이 서버로부터 언로드될 수 있습니다. FMTOPT 매개변수에 대해 지정된 *NULLFLAGS 옵션을 갖고 있는 CPYF는 널(null) 플래그를 인식하고, 테이프에서 인접 필드에 있는 자료를 무시하고, iSeries용 DB2 UDB에 필드 널(null)을 만들 수 있습니다. IBM 주 프레임으로부터 자료를 반입하기 위한 또 하나의 유용한 FMTOPT 매개변수 값은 *CVTFLOAT입니다. 이것은 System/390 형식으로 테이프에 저장된 부동 소수점 자료가 iSeries용 DB2 UDB에 의해 사용된 IEEE 형식으로 변환될 수 있도록 합니다.

파일 복사 명령을 사용한 iSeries 서버 간의 자료 이동

하나의 iSeries 서버로부터 다른 서버로 자료를 이동시키는 또 하나의 방법은 DDM으로 파일 복사 명령을 사용하여 자료를 복사하는 것입니다. 사용자는 파일 복사(CPYF), 소스 파일 복사(CPYSRCT) 및 조회 파일로부터 복사(CPYFRMQRYF) 명령을 사용하여 소스와 어플리케이션 소스(AS)의 파일 간에 자료를 복사할 수 있습니다. 사용자는 리모트 데이터베이스 파일로부터(또는 이 파일로) 로컬 관계형 데이터베이스나 장치 파일을 복사할 수 있고, 또한 리모트 파일이 리모트 파일로 복사될 수 있습니다.

예를 들어, 한 판매 대리점이 문을 닫게 되면, 분산 관계형 데이터베이스(RDB) 관리자는 리모트 서버로부터 로컬 지역 서버로 클라이언트와 재고표를 복사할 수 있습니다. 관리자는 표에 액세스하고 복사하기 위해 어플리케이션 소스(AS)에서 정당한 권한을 부여받은 사용자 프로파일을 필요로 하며, 복사되는 각 표나 파일에 대해 어플리케이션 리퀘스터(AR)에 DDM 파일을 작성하여야 합니다. 다음의 예는 데이터베이스 관리자가 KC105라는 리모트 위치명을 가지는 서버로부터 KC000라는 지역 센터 서버로 SPIFFY라는 컬렉션에 INVENT라는

명령표를 복사하는 데 사용한 명령을 나타내고 있습니다. 어플리케이션 리퀘스터(AR) KC000에서 TEST라는 라이브러리에 있는 INCOPY라는 DDM 파일은 파일 액세스에 사용됩니다. 이 명령은 KC000 서버에서 실행됩니다.

```
CRTDDMF FILE(TEST/INCOPY) RMTFILE(SPIFFY/INVENT)
      RMTLOCNAME(KC105)
CPYF FROMFILE(TEST/INCOPY) TOFILE(TEST/INVENTDDM)
      MBROPT(*ADD)
```

이 예에서, 관리자는 KC000 서버에서 이 명령을 실행합니다. 관리자가 KC000 서버에 없는 경우에 KC000 서버에서 이 명령을 실행하는 데 pass-through를 사용하여야 합니다. 리모트 명령 제출(SBMRMTCMD) 명령은 iSeries 서버가 같은 작업에 대해 어플리케이션 리퀘스터(AR)와 어플리케이션 소스(AS)가 될 수 없기 때문에 위의 명령을 실행하는 데 사용될 수 없습니다.

DDM으로 이 명령을 사용할 때 다음의 항목을 고려하십시오.

- DDM 파일은 파일 복사(CPYF) 명령과 소스 파일 복사(CPYSRFC) 명령을 위해 FROMFILE과 TOFILE 매개변수에 지정될 수 있습니다.

주: 조회 파일로부터 복사(CPYFRMQRYF), 디스켓으로부터 복사(CPYFRMDKT) 및 테이프로부터 복사(CPYFRMTAP) 명령에 대하여 DDM 파일명은 TOFILE 매개변수에만 지정될 수 있습니다. 또한 디스켓으로 복사(CPYTODKT) 및 테이프 복사(CPYTOTAP) 명령에 대하여 DDM 파일명은 FROMFILE 매개변수에만 지정될 수 있습니다.

- 삭제 가능(delete-capable) 파일이 비삭제가능 파일로 복사될 때 사용자는 COMPRESS(*YES)를 지정하여야 하며, 그렇지 않으면 오류 메시지가 송신되고 작업이 종료됩니다.
- DDM 파일에서 리모트 파일명이 멤버명을 지정하는 경우, 파일 복사(CPYF) 명령에서 해당 파일에 지정된 멤버명은 DDM 파일에서 리모트 파일의 멤버명과 같아야 합니다. 게다가, 데이터베이스 파일 대체(OVRDBF) 명령은 DDM 파일에서 리모트 파일명의 멤버명과 다른 멤버명을 지정할 수 있습니다.
- DDM 파일이 멤버명을 지정하지 않은 경우와 데이터베이스 파일 대체(OVRDBF) 명령이 해당 파일에 대해 멤버명을 지정하는 경우, CPYF(파일 복사) 명령은 OVRDBF 명령에 지정된 멤버명을 사용합니다.
- TOFILE 매개변수가 존재하지 않는 파일에 관련하는 DDM 파일인 경우, DPYF는 해당 파일을 작성합니다. 다음은 파일 복사(CPYF) 명령으로 작성된 리모트 파일에 대해 특별한 고려사항입니다.
 - 목표 DDM 작업에 대한 사용자 프로파일은 어플리케이션 소스(AS)에서 실제 파일(PF) 작성(CRTPF) 명령에 대한 권한을 부여받아야 합니다.
 - iSeries 목표를 위해, TOFILE 매개변수는 iSeries Information Center의 파일 관리 주제에 설명된 속성을 제외한 모든 FROMFILE 속성을 지니고 있습니다.
- TCP/IP를 사용할 때, 분산 자료 관리 파일 작성(CRTDDMF) 명령의 RMTLOCNAME 매개변수에서 두 번째 요소는 *IP이어야 합니다.

서버간 복사에 사용되는 파일 복사 명령에 대한 자세한 정보는 iSeries Information Center의 분산 자료 관리 주제를 참조하십시오.

네트워크 파일 명령을 사용하여 네트워크에 자료 전송

자료는 SNADS를 지원하는 네트워크 프로토콜을 통해 전송될 수 있습니다. 분산 관계형 데이터베이스(RDB) 처리에 의해 사용되는 APPC와 APPN 프로토콜에 더하여, SNADS는 BSC(EL(binary synchronous equivalence link)과 SNUF(SNA Upline Facility) 프로토콜과 함께 사용될 수 있습니다. SNADS에 의해 지원되는 iSeries 서버는 네트워크 파일 송신(SNDNETF) 명령으로 또다른 서버에 자료를 송신할 수 있고, 네트워크 파일 수신(RCVNETF) 및 네트워크 파일에 대한 작업(WRKNETF) 명령으로 또다른 서버로부터 네트워크 파일을 수신할

서버 저장 및 복원 명령을 사용한 표 이동

사용자는 오브젝트 저장(SAVOBJ) 및 오브젝트 복원(RSTOBJ) 명령을 사용하여 또 하나의 iSeries 서버로부터 표를 이동시킬 수 있습니다. 저장 명령은 테이프, 디스켓 또는 저장 파일에 데이터베이스 파일을 저장합니다. 저장 파일은 통신을 통해 또다른 서버로 분배됩니다.

표나 파일을 저장하고 복원하는 데 사용되는 저장 및 복원 명령은 다음을 포함합니다.

- 라이브러리 저장(SAVLIB) 명령은 하나 또는 그 이상의 컬렉션이나 라이브러리를 저장합니다.
- 오브젝트 저장(SAVOBJ) 명령은 하나 또는 그 이상의 오브젝트(데이터베이스 표와 뷰를 포함하여)를 저장합니다.
- 변경된 오브젝트 저장(SAVCHGOBJ) 명령은 컬렉션이나 라이브러리가 저장되었던 최종 시간이나 지정된 날짜로부터 변경된 오브젝트를 저장합니다.
- 라이브러리 복원(RSTLIB) 명령은 컬렉션이나 라이브러리를 복원합니다.
- 오브젝트 복원(RSTOBJ) 명령은 하나 또는 그 이상의 오브젝트(데이터베이스 표와 뷰를 포함하여)를 복원합니다.

예를 들어 두 판매 대리점이 합병되었다면, 저장 및 복원 명령을 사용하여 관계형 데이터베이스 하나에 대한 컬렉션과 표를 저장할 수 있으며, 컬렉션과 표들은 남은 서버의 관계형 데이터베이스에 복원됩니다. 이 작업을 수행하려면 관리자는 다음을 실행해야 합니다.

1. 서버 A에서 라이브러리 저장(SAVLIB) 명령을 사용하여 컬렉션을 저장하거나 서버 A에서 오브젝트 저장(SAVOBJ) 명령을 사용하여 표를 저장하십시오.
2. 자료가 저장 파일에 저장되어 SNADS를 사용하여 분배되거나 테이프나 디스켓에 저장될 수 있도록 지정하십시오.
3. 저장 파일을 서버 B에 분배하거나 테이프나 디스켓을 서버 B로 송신하십시오.
4. 서버 B에서 라이브러리 복원(RSTLIB) 명령을 사용하여 컬렉션을 저장하거나 서버 B에서 오브젝트 복원(RSTOBJ) 명령을 사용하여 표를 복원하십시오.

저장 및 복원 명령을 사용할 때 고려해야 할 사항은 복원된 오브젝트에 대한 소유권과 권한입니다. 현재의 오브젝트 소유자에 대한 유효 사용자 프로파일은 오브젝트가 복원되는 서버에 존재하여야 합니다. 현재 소유자의 프로파일이 이 서버에 없는 경우, 오브젝트는 QDFTOWN 디폴트 사용자 프로파일 하에서 복원됩니다. 오브젝트에 대한 사용자 권한은 디폴트 사용자 프로파일 매개변수에 의해 제한됩니다. QSECOFR 권한을 갖고 있

는 사용자는 이 서버에서 원래 소유자의 프로파일을 작성하고 복원된 오브젝트 소유권을 변경하거나, 로컬 및 리모트 사용자들 모두에 대하여 이 오브젝트에 대한 새로운 권한을 지정해야 합니다.

저장 및 복원 명령에 대한 자세한 정보는 iSeries Information Center의 백업 및 복구 주제를 참조하십시오.

비iSeries 서버에서 iSeries 서버로 데이터베이스 이동

사용자는 또다른 IBM 서버에서 iSeries 서버로 또는 비IBM 서버에서 iSeries 서버로 파일을 이동시킬 필요가 있습니다. 이 섹션은 비iSeries 서버에서 iSeries 서버로 자료를 이동시키기 위한 대안을 나열합니다. 그러나, 사용자는 다른 서버에 대해 공급되거나 사용에 있어서 특정의 지침에 대한 어플리케이션용으로 구분되는 매뉴얼을 참조하십시오.


또다른 IBM 서버에서 자료 이동

또다른 IBM 서버에서 iSeries 서버로 자료를 이동시키는 데 사용할 수 있는 여러 가지의 메소드가 있습니다. 이 메소드는 다음과 같습니다.

- 고급 언어(HLL) 프로그램은 또다른 서버에서 자료를 추출하기 위해 작성될 수 있습니다. 서버의 통신 프로그램은 이 서버로 자료를 로드하는데 사용될 수 있습니다.
- 다른 DRDA 실행을 지원하는 서버에서 자료를 이동시키는 데 SQL 기능을 사용할 수 있습니다. 예를 들어, 작업의 분산 장치를 사용하여 자료의 소스에 대한 조회를 열 수 있으며, 이와 같은 작업 단위에서, 자료를 서버에 있는 표 안에 삽입할 수 있습니다. 최선의 실행을 위해서, 조회에 블록 표시를 사용하여야 하며, 복수 열 삽입은 서버에서 행해야 합니다. 추가 정보에 대한 내용은 18 페이지의 『추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계』를 참조하십시오.
- 자료는 다른 서버에 있는 표와 파일에서 추출될 수 있으며, 테이프나 디스켓 상으로 또는 통신 회선을 통해 iSeries 서버로 송신될 수 있습니다.
 - DB2 UDB for z/OS 데이터베이스에서 데이터베이스 관리자에 의해 공급되는 DSNTIAUL이라는 샘플 프로그램은 파일이나 표에서 자료를 추출하는 데 사용될 수 있습니다.
 - DB2 UDB Server for VM(SQL/DS) 데이터베이스에서, 데이터베이스 관리자의 데이터베이스 서비스 유틸리티 부분은 자료를 추출하는 데 사용될 수 있습니다.
 - DB2 UDB for z/OS 또는 DB2 UDB Server for VM 데이터베이스에서 DXT*(자료 추출)는 자료를 추출하는 데 사용될 수 있습니다. 그러나, 널 자료의 DXT™ 처리는 아래에 설명되어 있는 널 자료의 파일 복사 처리와 호환될 수 없습니다. 따라서, DXT는 iSeries 서버로 마이그레이션하기 위해 관계형 자료를 언로드하는 데 사용하는 것은 좋지 않습니다.
 - IMS/DB 계층적 데이터베이스에서, DXT는 자료를 추출하는 데 사용될 수 있습니다.
- 사용자는 표준 테이프 관리 기술을 사용하여 DB2 UDB for z/OS 또는 DB2 UDB Server for VM 데이터베이스에서 테이프나 디스켓으로 자료를 복사할 수 있습니다. iSeries 서버는 테이프로부터 복사(CPYFRMTAP) 명령을 사용하여 테이프에서 자료를 로드합니다. 그러나, 파일 복사(CPYF) 명령은 IBM 주 프레임 컴퓨터에서 자료를 마이그레이트하도록 특별한 지원을 제공합니다. CPYF는 테이프 파일에 의한 대체(OVRTAPF) 명령을 사용하여 테이프 자료에 사용될 OVRTAPF 명령은 사용자가 iSeries 서버 이외의 서버에서 자료를 내보낼 때 필요할 수도 있는 특별한 테이프 특정 매개변수를 지정하게 합니다.

특별한 CPYF 지원은 널(null) 가능한 자료와 부동 소수점 자료를 내보내도록 합니다. 널 가능한 자료는 단일 바이트 플래그가 각 널 가능한 필드와 연관될 수 있는 방법으로 주 프레임에서 언로드될 수 있습니다. FMTOPT 매개변수에 대해 지정된 *NULLFLAGS 옵션으로, 파일 복사(CPYF) 명령은 널 플래그를 인식하고, 테이프에서 인접 필드에 있는 자료를 무시하고, iSeries용 DB2 UDB 에 필드 널을 만들 수 있습니다. IBM 주 프레임에서 자료를 반입하는 데 유용한 다른 FMTOPT 매개변수 값은 *CVTFLOAT입니다. 이것은 System/390 형식으로 테이프에 저장된 부동 소수점 자료가 iSeries용 DB2 UDB에 의해 사용된 IEEE 형식으로 변환될 수 있도록 합니다.

iSeries 서버에서의 테이프 장치 및 디스켓 장치 사용에 대한 자세한 정보는 기억장치 솔루션 주제를 참조하십시오. 서버간 복사에 사용되는 파일 복사 명령에 대한 자세한 정보는 iSeries Information Center의 분산 자료 관리 주제와 제어 언어(CL) 주제를 참조하십시오.

- 통신 회선을 통해 송신된 자료는 iSeries 서버에서 SNADS 지원을 통해 처리될 수 있습니다. SNADS 지원은 분산 관계형 데이터베이스(RDB) 처리에 사용되는 APPC나 APPN 프로토콜에 더하여 BSCCL과 SNUF 프로토콜에 대해서도 네트워크 파일을 전송합니다.
 - MVS™ 시스템에서, 자료는 TSO XMIT 기능을 사용하여 iSeries 서버로 송신될 수 있습니다. 서버는 네트워크 파일에 대한 작업(WRKNETF) 또는 네트워크 파일 수신(RCVNETF) 명령을 사용하여 네트워크 파일을 수신합니다.
 - VM 시스템에서, 자료는 SENDFILE 기능을 사용하여 서버로 송신될 수 있습니다. 서버는 네트워크 파일에 대한 작업(WRKNETF) 또는 네트워크 파일 수신(RCVNETF) 명령을 사용하여 네트워크 파일을 수신합니다.
- Microsoft® Windows로부터 클라이언트 자료가 별도로 주문된 IBM 제품인 iSeries Access를 사용하여 iSeries 서버로 송신될 수 있습니다.
- 다양한 워크스테이션 클라이언트에서, 사용자는 DB2 Connect IMPORT 및 EXPORT 유틸리티를 사용하여 iSeries 서버로 또는 iSeries 서버로부터 자료를 복사할 수 있습니다. IMPORT는 자료를 기존 표로만 가져올 수 있습니다. IMPORT 및 EXPORT 유틸리티에 대한 예는 레드북 Advanced Functions and Administration on DB2 Universal Database for iSeries을 참조하십시오. 이 레드북에는 IMPORT 및 EXPORT 유틸리티와 함께 사용할 수 있는 파일 유형과 자료 형식에 대한 정보도 있습니다.
- 또한, 비동기식 통신과 같이 SNADS를 지원하지 못하는 통신 회선을 통해 자료를 송신할 수도 있습니다. OS/400 사용권 프로그램의 일부인 FTS(파일전송 지원) 유틸리티는 자료를 송신하고 수신하는 데 사용될 수 있습니다. 통신과 통신 파일에 대한 작업에 관한 자세한 정보는 *ICF Programming*  서적을 참조하십시오.

비IBM 서버로부터의 자료 이동

사용자는 다른 서버로부터 테이프나 디스켓으로 파일이나 표를 복사하고 iSeries 서버에 이 파일을 로드할 수 있습니다. 가져오기 파일로부터 복사(CPYFRMIMPF) 명령을 사용하여 이를 실행하십시오.

또한, 공급자 독립형 통신 기능은 2개의 라이선스가 부여된 분리된 iSeries 프로그램을 통해 지원됩니다.

로컬 및 광역 네트워크 양쪽에 대해 대등시스템간 연결성(Peer-to-peer connectivity) 기능은 TCP/IP(Transmission Control Protocol/Internet Protocol)에 의해 제공됩니다. iSeries TCP/IP Connectivity Utilities/400 사용권 프로그램의 FTP(파일 전송 프로토콜) 기능은 사용자가 리모트 서버의 기능에 따라 많은 유형의 파일을 수신할 수 있게 합니다. 자세한 정보는 iSeries Information Center의 TCP/IP 설치 주제를 참조하십시오.

OSIFS/400(OSI File Services/400) 사용권 프로그램은 OSI(개방 서버 상호연결) 네트워크에 대하여 파일 관리 및 전송 서비스를 제공합니다. 요구사항 사용권 프로그램 OSI Communications Subsystem/400을 갖고 있는 OSIFS/400은 iSeries 서버를 OSI 파일 전송, 액세스 및 관리(FTAM) 표준과 일치하는 리모트 IBM 또는 비IBM 서버에 연결합니다.

OSIFS/400은 리모트 서버에서 로컬 iSeries 서버로 파일을 복사하거나 이동시킬 수 있도록 대화식 인터페이스나 어플리케이션 프로그래밍 인터페이스(API)를 제공합니다. 이에 대한 자세한 정보는 *OSI Communications Subsystem Programming and Concepts Guide*를 참조하십시오.

제 6 장 분산 관계형 데이터베이스(DRDB) 관리 및 조작 task

사용자는 분산 관계형 데이터베이스의 관리자로서, 여러 서버에서 실행하는 작업에 대해 책임이 있습니다. 어플리케이션 리퀘스터(AR)로서 사용자의 로컬 시스템에서 기원하는 작업은 어떤 것이든 다른 작업이 iSeries 서버에서 모니터되는 것과 같은 방법으로 모니터될 수 있습니다. 사용자가 어플리케이션 서버(AS)로서 로컬 시스템에서 실행 중인 작업 단위를 추적하고 있을 때, 사용자는 툴을 사용하지만 다른 종류의 정보를 찾게 됩니다.

이 장에서는 사용자가 네트워크 전체를 통해 실행 중인 분산 관계형 데이터베이스 작업을 관리할 수 있는 방법에 대해 다루고 있습니다. 여기에서 거론되는 명령, 프로세스 및 다른 자원의 대부분은 분산 관계형 데이터베이스(RDB) 사용을 위한 것만은 아닙니다. 이 모두가 iSeries 서버의 조작을 위해 제공됩니다. 여기에서 거론하는 모든 관리 명령, 처리 및 자원은 전체 iSeries용 DB2 UDB 기능과 함께 OS/400 프로그램과도 관련됩니다.

iSeries 서버의 작업 관리 기능은 다음을 실행하여 몇몇 서버 상의 작업을 추적하는 효과적인 방법을 제공합니다.

- 관계형 데이터베이스 활동 모니터
- 리모트 iSeries 서버 조작
- DDM 대화 제어
- 프로그램에 의해 사용된 오브젝트 표시
- 분산 관계형 데이터베이스에서 컬렉션 드롭(Drop)
- 분산 관계형 데이터베이스에서 작업 계정
- TCP/IP 서버 관리
- 관계형 데이터베이스 디렉토리 감사

관계형 데이터베이스 활동 모니터

제어 언어(CL) 명령에 따라, iSeries 서버에서 작업의 뷰가 사용자에게 주어질 수 있습니다. 이 명령은 모두 비슷한 정보를 제공하지만 그 방식은 다릅니다. 명령에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 분산 관계형 데이터베이스에서 작업에 대한 작업

작업에 대한 작업(WRKJOB) 명령은 사용자가 작업명이나 WRKJOB를 입력하는 작업에 대해 알고 있는 경우, 작업에 대해 특정한 정보를 줍니다.

- 분산 관계형 데이터베이스에서 사용자 작업에 대한 작업

사용자 작업에 대한 작업(WRKUSRJOB) 명령은 사용자가 작업을 실행하도록 하는 사용자 프로파일을 알고 있는 경우, 작업에 관한 보다 자세한 정보를 제공합니다(TCP/IP 환경에서 WRKUSRJOB QUSER *ACTIVE를 사용).

- 분산 관계형 데이터베이스에서 활동 작업에 대한 작업

활동 작업에 대한 작업(WRKACTJOB) 명령을 사용하면 서버에서 진행되고 있는 작업을 가장 일반적으로 관찰할 수 있습니다. 이것은 서버에서 현재 실행 중인 모든 작업과 각 작업에 대한 일부의 통계를 나타내고 있습니다.

- 분산 관계형 데이터베이스에서 확약 정의에 대한 작업

확약 정의에 대한 작업(WRKCMTDFN) 명령은 확약 정의를 표시합니다. 이것은 확약 제어가 확약 제어 시작(STRCMTCTL) 명령에 의해 시작될 때, 확약 제어에 대한 정보를 저장하는 데 사용됩니다.

이 명령을 사용하여 서버의 작업을 보는 것 외에 정보를 추적하거나 특정 작업을 찾으려 할 수도 있습니다. 자세한 정보는 다음 주제를 참조하십시오.

- 분산 관계형 데이터베이스의 작업 기록부를 사용한 요구 정보 트래킹
- 분산 관계형 데이터베이스 작업 찾기

분산 관계형 데이터베이스에서 작업(Job)에 대한 작업

작업에 대한 작업(WRKJOB) 명령은 작업 메뉴에 대한 작업을 나타냅니다. 이 메뉴를 사용하여 지정된 작업에 대한 작업이나 이것에 관련되는 정보에 대한 변경을 위한 옵션을 선택할 수 있습니다. 현재 사용 중인 작업에 대한 정보를 얻으려면 어떠한 매개변수도 가질 필요없이 이 명령을 입력하십시오. 작업을 지정하고 다음과 같은 명령에 작업명을 입력하여 작업에 속하는 동일한 정보를 얻으십시오.

WRKJOB JOB(job-number/user-ID/job-name)

메뉴의 옵션에 의해 해당 작업이 작업 대기행렬에 있는지, 출력 대기행렬에 있는지 또는 활동 중에 있는지를 알려주는 정보를 얻을 수 있습니다. 그러나, 작업의 항목 전체를 완전하게 읽은 후에야 비로소 작업이 서버에 존재하게 됩니다. 이 때, 항목은 작업 대기행렬에만 위치합니다. 작업 정보에 대한 옵션은 다음과 같습니다.

- 작업 상태 속성
- 작업 정의 속성
- 스폴 파일 정보

다음의 옵션에 대한 내용은 작업이 활동 중일 때에만 보여줄 수 있습니다.

- 작업 실행 속성
- 작업 로그 정보
- 프로그램 스택 정보
- 작업 잠금 정보
- 라이브러리 리스트 정보
- 열린 파일 정보
- 파일 대체 정보
- 확약 제어 상태
- 통신 상태

- 활성 그룹
- 뮤텍스(Mutexes)

옵션 10(작업 로그 표시)은 사용자에게 활동 중인 작업이나 작업 대기행렬에 있는 작업에 대한 정보를 줍니다. 종료한 작업에 대하여, 보통 옵션 4(스폴 파일에 대한 작업)를 사용하여 같은 정보를 찾을 수 있습니다. 사용자가 옵션 5를 사용하여 QPJOBLOC로 명명되는 파일(리스트에 있을 경우)을 표시하는 경우, 이것은 스폴 파일에 대한 작업 화면을 나타냅니다.

분산 관계형 데이터베이스에서 사용자 작업(Job)에 대한 작업

작업에 의해 사용되고 있는 사용자 프로파일(사용자명)을 알고 있는 경우, 사용자 작업에 대한 작업(WRKUSRJOB) 명령을 사용하여 작업 정보를 표시하거나 변경할 수 있습니다. 사용자 프로파일을 사용하여 서버에 있는 작업 리스트를 얻으려면 어떠한 매개변수도 필요없이 이 명령을 입력하십시오. 사용자와 작업 상태를 지정하여 이와 같이 명령 안에 작업명을 입력함으로써 작업 리스트를 짧게 단축할 수 있습니다.

WRKUSRJOB USER(KCDBA)

사용자 작업(Jobs)에 관한 작업 화면은 서버(*ACTIVE), 작업 대기행렬(*JOBQ) 또는 출력 대기행렬(*OUTQ)에서 실행 중인 사용자 작업의 이름과 상태에 대한 정보를 나타냅니다. 다음의 화면은 KCDBA라는 사용자의 활동 중 및 끝난 작업을 나타내고 있습니다.

```

                사용자 작업(Jobs)에 대한 작업                KC105
03/29/92 16:15:33
옵션을 입력한 후 Enter를 누르십시오.
  2=변경  3=보류  4=종료  5=작업 대상  6=해제  7=메세지 표시
  8=스폴 파일에 대한 작업  13=단절

옵션  작업          사용자          유형          -----상태-----          기능
---  KC000          KCDBA          CMNEVK          OUTQ
---  KC000          KCDBA          CMNEVK          OUTQ
---  KC000          KCDBA          CMNEVK          OUTQ
---  KC000          KCDBA          CMNEVK          OUTQ
---  KC000          KCDBA          CMNEVK          ACTIVE
---  KC0001         KCDBA          CMNEVK          ACTIVE          * -PASSTHRU
---  KC0001         KCDBA          INTER          ACTIVE          CMD-WRKUSRJOB

                                                                맨 아래

매개변수 또는 명령
====>
F3=종료  F4=프롬트  F5=화면정리  F9=검색  F11=스케줄 표시
F12=취소  F21=지원 레벨 선택

```

이 화면은 사용자의 서버에 들어 있는 모든 작업을 나열하며, 지정된 상태(이 경우, *ALL)를 보여 주며, 작업 유형을 나타냅니다. 또한, 이것은 사용자에게 선택된 작업을 위해 명령을 입력하는 데 8가지 옵션(2 - 8 및 13)을 제공합니다. 옵션 5는 위에서 설명하는 작업(Job)에 대한 작업 화면을 나타냅니다.

사용자 작업에 대한 작업(WRKUSRJOB) 명령은 사용자의 서버가 TCP/IP를 사용하고 있는 경우, DDM TCP/IP의 상태를 보려고 할 때 유용하게 쓰입니다. 다음 명령을 실행하십시오.

WRKUSRJOB QUSER *ACTIVE

문자, QRWT로 시작하는 작업이 나타날 때까지 PgDn을 누르십시오. 서버가 활동 중이면, QRWTLSTN라는 이름을 갖는 하나의 작업을 찾고(사전시작 DRDA 작업이 서버에서 실행되지 않을 경우에) QRWTSRVR이라는 이름을 가지는 하나 또는 그 이상의 작업을 찾아야 합니다. QRWTSRVR 작업은 사전시작 작업입니다. RWTLSTN 작업을 찾지 못했다면, 다음의 명령을 실행하여 시작하십시오.

```
STRTCPSVR *DDM
```

QRWTLSTN 작업은 찾고, QRWTSRVR 작업은 찾지 못하고, 또한 DRDA 사전시작 작업은 작동 불가능한 상태이면, 다음의 명령을 실행하여 사전시작 작업을 시작하십시오.

```
STRPJ subsystem QRWTSRVR
```

V5R2 이전에는 QRWTSRVR이 정상적으로 실행되는 서브시스템은 QSYSWRK였습니다. V5R1 이후 QRWTSRVR은 QUSRWRK에서 실행됩니다.

분산 관계형 데이터베이스에서 활동 작업(Job)에 대한 작업

여러 사용자에게 대해 실행되는 작업을 모니터링하거나 작업을 찾거나 작업명 또는 사용자 ID를 알지 못할 경우 WRKACTJOB(활동 작업에 대한 작업) 명령을 사용하십시오. 이 명령을 입력할 때, 활동 중인 작업(Jobs)에 대한 작업 화면이 나타납니다. 이것은 서버에서 현재 활동 중인 작업에 대한 실행과 상태에 대한 정보를 나타내고 있습니다. 모든 정보는 한 개의 작업 단위로 모아 서브시스템별로 그룹을 만듭니다.

아래의 화면은 KC105 시스템에서 활동 중인 작업(Job)에 대한 작업 화면을 나타내고 있습니다.

```

              활동 중인 작업(Job)에 대한 작업              KC105
03/29/92 16:17:45
CPU %:   41.7   경과 시간:   04:37:55   활동 중인 작업: 42

옵션을 입력한 후 Enter를 누르십시오.
  2=변경  3=보류  4=종료  5=작업 대상  6=해제  7=메세지 표시
  8=스플 파일에 대한 작업  13=단절 ...

옵션  서브시스템/작업  사용자  유형  CPU %  기능  상태
---  ---  ---  ---  ---  ---  ---
---  QBATCH  QSYS  SBS  .0  .0  DEQW
---  QCMN  QSYS  SBS  .0  .0  DEQW
---  QINTER  QSYS  SBS  .0  .0  DEQW
---  DSP01  CLERK1  INT  .0  CMD-STRSQL  DSPW
---  DSP02  CLERK2  INT  .0  * -CMDENT  DSPW

                                                    계속...

매개변수 또는 명령
====>
F3=종료  F5=화면정리  F10=통계 재시작  F11=경과 자료 표시
F12=취소  F23=추가 옵션  F24=추가 키
  
```

F11 키(경과 자료 표시)를 누르면, 다음의 화면이 나타나 자세한 상태 정보를 나타내고 있습니다.

```

          활동 중인 작업(Job)에 대한 작업          KC105
03/29/92 16:17:45
CPU %: 41.7   경과 시간: 04:37:55   활동 중인 작업: 42

옵션을 입력한 후 Enter를 누르십시오.
2=변경 3=보류 4=종료 5=작업 대상 6=해제 7=메세지 표시
8=스플 파일에 대한 작업 13=단절 ...
-----경과된-----
옵션  서버시스템/작업  유형  풀(pool)  Pty  CPU  Int  Rsp  AuxIO  CPU %
--  --
--  QBATCH             SBS   2       0     4.4  --   --   108    .0
--  QCMN               SBS   2       0    20.7  --   --   668    .0
--  KC000              EVK   2      50     .1    --   --    9     .0
--  KC0001             EVK   2      50     .1    --   --    9     .0
--  MP000              EVK   2      50     .1    --   --   14    .0
--  QINTER             SBS   2       0     7.3  --   --    4     .0
--  DSP01              INT   2      20     .1    --   --    0     .0
--  DSP02              INT   2      20     .1    --   --    0     .0

                                          계속...

매개변수 또는 명령
====>
F3=종료      F5=화면정리      F10=통계 재시작   F11=상태 표시
F12=취소    F23=추가 옵션   F24=추가 키

```

활동 중인 작업(Job)에 대한 작업 화면은 사용자 작업(Job)에 대한 작업에서 얻은 사용자 및 유형 정보 뿐만 아니라 작업 우선순위와 서버 사용에 대한 정보를 나타내고 있습니다. 또한, 옵션 5를 포함하여 작업에 관한 11개의 옵션(2 - 11 및 13) 중 어느 것이든 사용할 수 있고, 해당 옵션은 선택한 작업에 대해서 작업(Job)에 대한 작업의 화면을 제시합니다.

작업 우선순위와 서버 사용법에 관한 정보를 보는 다른 방법은 iSeries Navigator를 사용하는 것입니다. 이를 수행하려면 다음 단계를 수행하십시오.

1. iSeries Navigator 인터페이스에서 데이터베이스를 선택하십시오.
2. 정보를 보려는 리모트 데이터베이스를 선택하십시오.
3. 마우스의 오른쪽 버튼을 클릭하고 등록 정보를 선택하십시오. 이는 정보가 표시된 등록 정보 창을 엽니다.

분산 관계형 데이터베이스에서 확약 정의에 대한 작업

서버에서 확약 정의에 대한 작업을 하려면, 확약 정의에 대한 작업(WRKCMTDFN) 명령을 사용하십시오. 확약 정의는 확약 제어가 확약 제어 시작(STRCMTCTL) 명령에 의해 시작될 때, 확약 제어에 대한 정보를 저장하는이 확약 정의는 활동 중인 작업과 연관되거나 연관되지 않을 수도 있습니다. 활동 중인 작업과 연관되지 않은 정의는 종료되었지만, 하나 또는 그 이상의 논리적 작업 단위는 아직 완료되지 않았습니다.

확약 정의에 대한 작업(WRKCMTDFN) 명령은 작업명, 상태 또는 해당 확약 정의에 대한 논리적 작업 단위 ID에 기초하여 확약 정의에 대한 작업에 사용될 수 있습니다.

사용자는 STATUS 매개변수에 모든 작업 또는 *RESYNC나 *UNDECIDED의 상태값을 가지는 작업만을 지정할 수 있습니다. *RESYNC는 동기화(SYN) 점을 재설치하려고 할 때 자원을 재동기화하는 데 관련하는 작업만을 보여 주며, 동기화(SYN) 점은 모든 자원이 일정한 상태에 있을 때의 지점입니다.

*UNDECIDED는 확약 결정이나 롤백 자원의 값이 알 수 없는 작업만을 나타내고 있습니다.

사용자는 LUWID 매개변수에 또다른 서버에서 확약 정의에 대한 작업 중인 확약 정의를 표시할 수 있습니다. 이 확약 정의를 포함하는 작업은 APPC로 보호되는 대화를 사용하여 통신하고 있습니다. LUWID는 하나의 서버에 확약 정의를 표시함으로써 찾을 수 있고, 그 다음에 이것을 확약 정의에 대한 작업(WRKCMTDFN) 명령에 입력으로서 사용하여 통신의 확약 정의를 찾을 수 있습니다.

확약 정의에 대한 작업(WRKCMTDFN) 명령을 사용하여 결정되지 않은 작업에 들어 있는 로컬 자원을 사용 가능하게 할 수 있습니다. 그러나, 확약 정의가 PRP(Prepared) 또는 LAP>Last Agent Pending: 최종 에이전트 지연 중) 상태에 있는 경우에만 해당합니다. 확약 정의를 강제하여 확약 또는 롤백할 수 있으며, 이 방식으로 보류된 자료를 자유롭게 할 수 있습니다. 반면, 제어는 개시자가 확약 정의에 취한 활동에 대해 알게 될 때까지 원래의 확약을 제공한 프로그램으로 리턴하지 않습니다.

또한, 사용자는 확약 정의에 대한 작업(WRKCMTDFN) 명령을 사용하여 또다른 서버에 대한 재동기화가 결국 완료되지 않는다는 판별을 하는 경우에 재동기화를 끝낼 수 있습니다.

확약 제어 및 재동기화에 대한 자세한 정보는 iSeries Information Center의 트랜잭션 및 확약 제어 문제점 해결주제를 참조하십시오.

분산 관계형 데이터베이스의 작업 기록부를 사용한 요구 정보 트래킹

iSeries 서버에서 모든 작업은 작업에 대해 입력된 요구에 관련하는 정보를 포함하는 작업 기록부를 갖고 있습니다. 작업 기록부에는 다음의 정보가 들어 있습니다.

- 작업에 의해 사용된 정보
- 프로그램 메시지 대기행렬로부터 송신되고 제거되지 않은 메시지
- 프로그램이 LOGCLPGM(*JOB)으로 작성되었고, 작업이 LOGCLPGM(*YES)을 지정하거나 CL 프로그램이 LOGCLPGM(*YES)으로 작성된 경우에 CL 프로그램에 들어 있는 명령

작업 종료 시, 이 작업 기록부가 QPJOBLOC라는 스푼 파일로 기록될 수 있고 원래의 작업 기록부는 삭제됩니다. 작업 설명의 LOC 매개변수를 지정함으로써 작업 기록부에 기록되어 있는 정보를 제어할 수 있습니다.

작업 기록부를 표시하는 방법은 작업 상태에 따라 달라집니다. 작업이 끝나고 이 작업 기록부가 아직 인쇄되지 않은 상태이면, 사용자 작업에 대한 작업(WRKUSRJOB) 명령을 사용하여QPJOB LOC라는 스푼 파일을 찾고 옵션 5(작업 기록부 표시)를 선택하십시오. 또한, 작업에 대한 작업(WRKJOB) 명령과 작업 화면에 대한 작업에서 다른 옵션을 사용하여 작업 기록부를 표시할 수도 있습니다.

일괄처리 또는 대화식 작업이 아직 활동 중이거나 작업 대기행렬에 있을 경우, WRKUSRJOB 명령을 사용하여 해당 작업을 찾으십시오. 활동 작업에 대한 작업(WRKACTJOB) 명령은 활동 중인 작업 기록부를 표시하는 데 사용되며, 작업 대기행렬 상의 작업을 보여 주지는 않습니다. 옵션 5(작업(job)에 대한 작업)를 선택한 다음, 옵션 10(작업 기록부 표시)을 선택하십시오.

사용자 소유의 대화식 작업에 관한 작업 기록부를 표시하려면, 다음 중 하나를 실행하십시오.

- 작업 기록부 표시(DSPJOBLOG) 명령을 입력하십시오.
- 작업(Job)에 대한 작업(WRKJOB) 명령을 입력하고 작업(Job) 화면에 대한 작업에서 옵션 10(작업 기록부 표시)을 선택하십시오.

- 작업 기록부에 나타나는 메시지를 표시하려면 명령 입력 화면에서 F10 키(상세 메시지 표시)를 누르십시오.

작업 기록부 표시(DSPJOBLOC) 명령을 사용할 때에 작업 기록부 화면을 참조하십시오. 이 화면은 다음과 같이 특수 기호를 포함하는 프로그램명을 나타내고 있습니다.

- >> 실행 중인 명령 또는 다음에 실행될 명령. 예를 들어, CL 또는 고급 언어 프로그램이 호출되었다면 그 프로그램에 대한 호출을 나타내고 있습니다.
- > 명령이 처리를 완료하였습니다.
- . . 명령이 아직 처리되지 않았습니다.
- ? 응답 메시지. 이 기호는 응답을 요구하는 메시지와 응답 받은 메시지를 표시합니다.

분산 관계형 데이터베이스 작업 찾기

어플리케이션 리퀘스터(AR)에서 분산 관계형 데이터베이스(RDB) 작업에 대한 정보를 구하고 있고 사용되는 사용자 프로파일을 알고 있을 때, 사용자 작업에 대한 작업(WRKUSRJOB) 명령을 사용하여 해당 작업을 찾을 수 있습니다. 또한, 사용자는 어플리케이션 서버(AS)에서 이 명령을 사용할 수도 있으나, AS 상의 사용자 프로파일이 AR에 의해 사용된 것과 다를 수도 있다는 사실을 알고 있어야 합니다. TCP/IP 서버에 대하여 작업명을 규정하는 사용자 프로파일은 항상 QUSER이 되며 이것의 작업명은 항상 ORWTSRVR이 됩니다. 기록부 표시(DSPLOG) 명령은 완료된 서버 작업명을 찾으려 하는 데 사용될 수 있습니다. 메시지는 다음과 같은 형식입니다.

```
DDM job 031233/QUSER/QRWTSRVR servicing user XY on 10/02/97 at
22:06
```

지정된 사용자 프로파일에 대해 여러 개의 작업이 나열되어 있고 관계형 데이터베이스가 DRDA를 사용하여 액세스되는 경우, 작업(Job)에 대한 작업 화면을 얻으려면 옵션 5(작업(job)에 대한 작업)를 입력하십시오. 이 화면에서, 작업 기록부를 참조하려면 옵션 10(작업 기록부 표시)을 입력하십시오. 작업 기록부는 이것이 분산 관계형 데이터베이스(RDB) 작업인지를 보여 주고, 이 작업이 맞을 경우에는 이것을 어느 리모트 서버로든 연결합니다. 작업 기록부를 대충 훑어 보면서 다음과 같은 메시지 중 하나를 찾아보십시오(연결이 APPC를 사용하느냐 TCP/IP를 사용하느냐에 따라 메시지가 달라짐).

CPI9150

DDM 작업 시작

CPI9160

TCP/IP 또는 로컬 소켓을 통해 시작되는 데이터베이스 연결

메시지 CPI9150 및 CPI9160에 대한 두 번째 레벨 텍스트는 AS 작업에 대한 작업명을 포함합니다.

AS에 존재하는 상태에서 작업명을 알지 못하지만,² 사용자명을 알고 있는 경우, 사용자 작업에 대한 작업(WRKUSRJOB) 명령을 사용하십시오. 사용자를 지정하지 않은 경우, 이 명령은 사용하고 있는 사용자 프로

2. 사용자가 DDM TCP/IP를 사용하고 있으면, 위에서 설명한 대로 기록부 표시(DSPLOG) 명령으로 작업명을 찾을 수 있습니다.

3. TCP/IP에 대하여, 작업명에서 사용자 프로파일은 항상 QUSER이 됩니다.

파일³하에 있는 작업의 리스트를 리턴합니다. 사용자 작업(Jobs)에 대한 작업 화면에서, 이 열을 사용하여 APPC 연결에 서비스를 제공 중인 AS 작업을 식별하게 하십시오.

- 1** **작업 유형 열은 APPC 통신 작업에 대해 CNNEVK로서 나열되어 있는 유형으로 작업을 나타내고 있습니다.**
- 2** **상태 열은 작업이 활동 중인지 완료되었는지를 나타내고 있습니다. 작업 기록은 서버가 설정되어 있는 방법에 따라 좌우되며, 사용자는 활동 중인 작업만을 참조할 수 있습니다.**
- 3** **작업 열은 작업명을 제공합니다. AS에서 작업명은 사용되고 있는 장치와 동일합니다.**

```

          사용자 작업(Jobs)에 대한 작업                KC105
03/29/92  16:15:33
옵션을 입력한 후 Enter를 누르십시오.
  2=변경  3=보류  4=종료  5=작업 대상  6=해제  7=메세지 표시
  8=스플 파일에 대한 작업  13=단절

옵션  작업      사용자      유형      -----상태-----   기능
---  KC000      KCDBA      CMNEVK   OUTQ
---  MP000      KCDBA      CMNEVK   OUTQ
---  MP000      KCDBA      CMNEVK   OUTQ
---  KC000      KCDBA      CMNEVK   OUTQ
---  KC000      KCDBA      CMNEVK   ACTIVE
---  KC0001     KCDBA      INTER    ACTIVE                    CMD-WRKUSRJOB
3
    
```

활동 중인 AS 작업을 찾고 있는데 사용자명을 알지 못하는 경우, 활동 작업에 대한 작업(WRKACTJOB) 명령은 서버에서 활동 중인 서브시스템에 대하여 해당 작업의 리스트를 제공합니다. 다음의 예는 찾으려는 몇 가지 항목에 대해 나타내고 있습니다.

```

          활동 중인 작업(Job)에 대한 작업                KC105
03/29/92  16:17:45
CPU %:   41.7   경과 시간:   04:37:55   활동 중인 작업:   102

옵션을 입력한 후 Enter를 누르십시오.
  2=변경  3=보류  4=종료  5=작업 대상  6=해제  7=메세지 표시
  8=스플 파일에 대한 작업  13=단절

옵션  서브시스템/작업  사용자      유형  CPU %   기능  상태
4  QBATCH            QSYS        SBS   .0      .0    DEQW
---  QCMN              QSYS        SBS   .0      .0    WDEQ
---  KC0001           KCCLERK     EVK    .0      *     EVTW
5
    
```

- 4** **설정되어 있는 서브시스템⁴을 탐색하여 AS 작업을 처리하십시오. 이 예에서, AS 작업용 서브시스템은 QCMN입니다.**
- 5** **APPC AS 작업에 대하여, 작업명은 AS 사용을 위해 작성되어 있는 장치의 장치명입니다.**
- 6** **나열되어 있는 작업 유형⁵은 일반적으로 EVK이며, 프로그램 시작 요구에 의해 시작됩니다.**

4. TCP/IP 서버 작업을 위한 서브시스템은 V5R2 이전에는 QSYSWRK였으며 V5R1 이후에는 QUSRWRK입니다.

5. TCP/IP AS 작업에 대하여 작업 유형은 PJ입니다(DRDA 사전시작 작업이 서버에서 비활동 상태가 아닌 경우와 비활동 상태인 경우에 작업 유형은 BCI입니다.).

후보처럼 보이는 작업을 찾았을 때, 옵션 5를 입력하여 해당 작업(job)에 대한 작업을 실행하십시오. 그런 다음, 작업에 대한 작업 메뉴에서 옵션 10을 선택하여 작업 기록부를 표시하십시오. iSeries용 DB2 Universal Database 어플리케이션 리퀘스터로부터 AS를 액세스 중인 작업의 분산 데이터베이스 작업 기록부에는 맨 위쪽에 하나의 명령문이 들어 있습니다.

CPI3E01

(시스템명)에 의해 액세스된 로컬 관계형 데이터베이스.

AS에서 작업 중인 작업(job)을 찾은 후에도 AR이 iSeries 서버인 경우에 AR로 그것을 역추적할 수 있습니다. 다음의 메시지들 중 하나가 사용자의 작업 로그에서 나타나고 사용자가 수신한 메시지에 커서가 위치합니다.

CPI9152

목표 DDM 작업이 어플리케이션 리퀘스터(AR)에 의해 시작되었습니다.

CPI9162

TCP/IP를 통해 어플리케이션 리퀘스터에 의해 시작된 DDM 연결을 처리하기 위해 할당된 목표 작업.

도움말 키를 누르면, 명령문에 대한 자세한 메시지가 나타납니다. 이름이 주어진 어플리케이션 리퀘스터(AR) 작업은 이 작업을 야기시킨 AR 상의 작업입니다.

리모트 iSeries 서버 조작

사용자는 때때로 분산 관계형 데이터베이스에서 관리자로서 리모트 iSeries 서버 시스템을 조작해야 할지도 모릅니다.

예를 들어, 사용자는 리모트 서버를 시작하거나 중단해야 할 수 있습니다. iSeries 서버는 리모트 서버가 필요 시에 조작 중이라는 사실을 확인하도록 하는 옵션을 제공합니다. 물론, 리모트 서버가 조작 중임을 보증하는 가장 간단한 방법은 리모트 위치가 해당 서버를 공급 분산 관계형 데이터베이스(RDB) 요구사항을 충족시킬 수 있도록 하는 것입니다. 그러나, 이것이 언제나 가능하지는 않습니다. 사용자는 전원 공급과 전원 차단 스케줄을 자동으로 설정하거나 리모트 서버에 대해 리모트 전원 공급을 작동할 수 있습니다. 전원 공급 및 전원 차단 스케줄에 대한 자세한 정보는 자동 전원 공급 및 전원 차단 스케줄 설정 주제를 참조하십시오. IPL 및 리모트 IPL을 제어하는 서버 값에 대한 자세한 정보는 IPL을 제어하는 시스템 값 주제를 참조하십시오.

서버는 실제 시간이나 이전에 스케줄된 시간에 이를 실행할 여러 가지의 방법을 제공합니다. 좀더 자주, 리모트 서버의 조작 중에 특정 타스크를 실행해야 할 수도 있습니다. 사용자가 이 일을 할 수 있는 3가지의 1차 방법은 표시장치 pass-through, 리모트 명령 제출(SBMRMTCMD) 명령 또는 저장 프로시듀어를 사용하는 것입니다.

리모트 명령 제출(RBMRMTCMD) 명령은 분산 자료 관리(DDM) 지원을 사용하여 CL 명령을 제출하고 어플리케이션 서버(AS)에서 실행합니다. 사용자는 가장 먼저 DDM 파일을 작성해야 합니다. DDM 파일의 리모트 위치 정보는 사용되어야 할 통신 회선을 판별하는 데 사용됩니다. 이 방식으로, DDM 파일은 제출된 명령을 수신하는 어플리케이션 서버(AS)를 식별합니다. DDM 파일과 연관되는 리모트 파일은 DDM 파일이 명령

을 제출하여 어플리케이션 서버(AS)에서 실행하도록 하는 데 사용될 때에는 관련되지 않습니다. DDM 파일 작성에 관한 자세한 정보는 85 페이지의 『DDM 파일 설정』을 참조하십시오.

리모트 명령 제출(SBMRMTCMD) 명령은 일괄처리 환경과 QCAEXEC 시스템 프로그램 경유로 실행될 수 있는 CL 명령은 어느 것이든지 제출할 수 있습니다. 다시 말해, 이 명령은 ALLOW 속성에 대해 지정되는 *BPGM 및 *EXEC라는 값을 갖고 있습니다. 사용자는 명령 표시(DSPCMD) 명령을 사용하여 ALLOW 속성을 표시할 수 있습니다.

리모트 명령 제출(SBMRMTCMD) 명령의 1차 목적은 어플리케이션 리퀘스터(AR) 사용자나 프로그램이 어플리케이션 서버(AS)에 위치하는 오브젝트에 대하여 파일 관리 조작과 파일 권한 활동을 실행할 수 있도록 하는 것입니다. 이 명령의 2차 목적은 사용자가 메시지 대기행렬과 같은 비파일 조작을 수행하거나 어플리케이션 서버(AS)에서 실행될 사용자 작성 명령을 제출할 수 있도록 하는 것입니다. CMD 매개변수는 어플리케이션 서버(AS)에서 실행될 명령을 나타내는 문자 스트링이 2000자까지 지정 가능하도록 합니다.

사용자는 제출되는 CL 명령과 이 명령이 조작하는 오브젝트에 대해 어플리케이션 서버(AS)에서 정당한 권한을 가져야 합니다. 어플리케이션 리퀘스터(AR) 사용자가(어플리케이션 서버(AS) 사용자 프로파일에서 관별된 대로) 실행하는 데 올바른 권한을 갖고 있는 경우, 다음과 같은 활동은 리모트 명령 제출(SBMRMTCMD) 명령을 사용하여 리모트 파일에서 실행될 수 있는 것에 관한 예입니다.

- 리모트 표에 대한 오브젝트 권한 부여 또는 취소
- 표나 다른 오브젝트 확인
- 표나 다른 오브젝트 저장 또는 복원

그 명령이 리모트 서버에서 표나 다른 오브젝트를 갖고 많은 일을 실행하는 데 사용될 수 있다 하더라도, 일부 TASK에 이 명령을 사용하는 것은 iSeries 서버에서 다른 메소드만큼 효율적이지 못합니다. 예를 들어, 사용자는 이 명령을 사용하여 리모트 파일에 관한 파일 설명이나 필드 속성을 표시하거나, 파일이나 다른 오브젝트를 덤핑할 수도 있지만, 출력은 어플리케이션 서버(AS)에 남아 있습니다. 어플리케이션 리퀘스터(AR)에서 리모트 파일 설명과 필드 속성을 표시하기 위해 보다 좋은 메소드는 지정된 SYSTEM(*RMT)으로 파일 설명 표시(DSPFD) 및 파일 필드 설명 표시(DSPFFD) 명령을 사용하여 리모트 파일과 연관되는 DDM 파일의 이름을 지정하는 것입니다.

제출할 수 있는 CL 명령 리스트 및 이 명령 사용에 대한 제한사항은 분산 자료 관리 책을 참조하십시오. 이에 더하여, DDM이 대화를 공유하는 방법에 대한 자세한 정보는 『DDM 대화 제어』를 참조하십시오.

DDM 대화 제어

주: 대화라는 용어는 SNA APPC 전문용어로 특정한 기술적인 의미를 갖고 있습니다. 이것은 공식적인 의미에서 TCP/IP 전문용어까지 연장되지 않습니다. 그러나, TCP/IP 안에 유사한 개념(다른 책의 해당 주제에서 '네트워크 연결')이 있습니다. 이 책에서, 해당 단어는 TCP/IP 네트워크 연결에도 적용된다는 의미로 사용되고 있습니다. 이 책의 다른 섹션에서, 해당 용어는 특정의 APPC 의미를 보유하고 있지만 READER가 문맥에서 올바른 의미를 분별할 수 있어야 합니다.

이 책의 이 섹션에서 연결이란 용어는 SQL 연결의 개념에 관련합니다. SQL 연결은 명시성 또는 내재성의 SQL CONNECT를 행하는 시간부터 논리 SQL 연결이 SQL DISCONNECT와 같은 수단에 의해 종료되거나 RELEASE 다음에 COMMIT이 이어질 때까지 지속됩니다. 복수 SQL 연결은 단일 네트워크 연결이나 대화를 통해 연쇄적으로 발생할 수 있습니다. 다른 말로 표현하면, 하나의 연결이 끝났을 때 이것을 운반하는 대화가 반드시 끝나는 것은 아닙니다.

어플리케이션 리퀘스터(AR)가 DRDA를 사용하여 어플리케이션 서버(AS)에 연결하는 경우, DDM 대화를 사용합니다. 대화는 AR에서 SQL CONNECT 명령문을 사용하여 설치되며 단, 이것은 다음의 경우에만 해당합니다.

- 같은 리모트 위치값을 사용하는 대화가 이미 AR 작업에 대해 존재하지 않을 경우.
- 대화가 같은 활성 그룹을 사용하는 경우.
- DDM에서 시작되었으면, 대화는 해당 활성 그룹으로 스코프된 파일을 가집니다.
- 대화는 동일한 대화 유형을 갖고 있습니다(보호되거나 보호되지 않거나).

DDM 대화는 다음 세 가지 상태 - 활동 중, 사용되지 않음 또는 제거됨 - 중 하나로 존재할 수 있습니다. 분산 관계형 데이터베이스에 의해 사용되는 DDM 대화는 AR이 AS에 연결되어 있는 동안에는 활동 중입니다.

SQL DISCONNECT 및 RELEASE 명령문은 연결을 끝내는 데 사용됩니다. 또한, 연결은 서버에 의해 암시적으로 종료될 수도 있습니다. 이에 더하여, RUW 연결 관리로 실행 중일 때, 이전의 연결은 CONNECT가 실행될 때 종료됩니다. 연결이 끝나는 때에 대한 자세한 정보는 203 페이지의 『명시적 CONNECT』를 참조하십시오. 연결이 끝난 후에, DDM 대화는 사용되지 않는 상태 또는 제거 상태가 됩니다. DDM 대화가 사용되지 않는 경우, 리모트 데이터베이스 관리 시스템에 대한 대화는 DDM 통신 관리자에 의해 유지보수되고 사용되지 않는 것으로 표시됩니다. DDM 대화가 제거되는 경우, DDM 통신 관리자는 대화를 종료합니다. DDMCNV 작업 속성은 더이상 활동 중이 아닌 연결에 대해 DDM 대화가 사용되지 않는 상태인지 제거 상태인지 판별합니다. 작업 속성 값이 *KEEP이고 또 하나의 iSeries 서버에 연결되어 있을 경우, 대화는 사용되지 않게 됩니다. 작업 속성 값이 *DROP이거나 다른 iSeries 서버에 연결되어 있지 않은 경우, 대화가 제거됩니다.

DDMCNV 작업 속성 *KEEP을 사용하는 것은 리모트 관계형 데이터베이스로의 연결이 자주 변경될 때 바람직한 방법입니다.

속성 값 *DROP은 다음과 같은 상황에서 바람직합니다.

- 대화를 유지보수하는 비용이 높고 그 대화가 상대적으로 곧 사용되지 못하게 될 때.
- RUW 연결 관리로 컴파일된 프로그램과 DUW 연결 관리로 컴파일된 프로그램의 혼합 형태로 실행할 때. 리모트 위치에 대해 RUW 연결 관리로 컴파일된 프로그램을 실행하려고 시도하는 경우, 보호되는 대화가 존재할 때에는 장애가 발생합니다.
- DDM 또는 DRDA로 보호되는 대화를 실행할 때. 추가 오버헤드는 사용되지 않는 보호 상태의 대화에 대한 확약과 롤백에서 초래됩니다.

또한, DDM 대화가 DDM을 통해 리모트 파일에서 조작하는 데 사용되고 있는 경우, 해당 대화는 다음과 같은 조건이 충족될 때까지 활동 중으로 남아 있습니다.

- 대화에 사용되는 모든 파일이 닫힌 상태로 풀립니다.
- 다른 DDM에 관련된 기능이 전혀 실행되지 않습니다.
- 어떠한 DDM 관련 기능도 인터럽트되지 않았습니다(예를 들어, 구분 프로그램에 의해).
- 보호되는 대화에 대하여, 확약이나 롤백은 모든 SQL 프로그램을 끝낸 후와 모든 DDM 관련 기능이 완료된 후에 실행되었습니다.
- 이제는 AR 작업이 AS에 연결되지 않습니다.

DDMCNV 작업 속성 값에 관계없이, 대화는 작업 라우팅의 마지막 단계, 작업의 마지막에 또는 작업이 재라우트 작업(RRTJOB) 명령을 개시할 때에 제거됩니다. 또한, 활동 중인 작업 내에서 사용되지 않는 대화는 RCLDDMCNV(DDM 대화 재생) 또는 RCLRSC(자원 재생) 명령에 의해 제거될 수도 있습니다. 자세한 정보는 DDM 자원 재생을 참조하십시오. 또한, 통신 회선 장애와 같이 오류가 발생하여 대화가 제거될 수도 있습니다.

DDMCNV 매개변수는 작업 변경(CHGJOB) 명령에 의해 변경되고 OPTION(*DFNA)을 사용하여 작업 표시(DSPJOB) 명령에 의해 표시됩니다. 또한, 작업 속성 검색(RTVJOBA) 명령을 사용하여 이 매개변수 값을 얻고 이것을 CL 프로그램 내에서 사용할 수 있습니다.

DDM 자원 재생

분산 자료 관리 대화 재생(RCLDDMCNV) 명령은 작업의 DDMCNV 속성 값이 *KEEP일 경우에도 현재 자원 작업에 의해 사용되고 있지 않은 모든 어플리케이션 대화를 재생합니다. 명령으로 열린 파일을 모두 닫거나 자원 재생(RCLRSC) 명령에 의해 실행되는 어떤 다른 기능을 실행하지 않고도 사용되지 않은 DDM 대화를 재생할 수 있습니다.

분산 자료 관리 대화 재생(RCLDDMCNV) 명령은 이것이 입력되어 있는 어플리케이션 리퀘스터(AR) 상의 작업에 대한 DDM 대화에 적용됩니다. AR 작업에 의해 사용되는 DDM 대화에 대해 연관되는 AS 작업이 있습니다. AS 작업은 연관되는 DDM 대화가 끝나면 자동으로 끝납니다⁶.

이 명령이 작업에 의해 사용되는 모든 DDM 대화에 적용된다 하더라도, 이것을 사용하는 것이 전체의 대화가 재생된다는 것을 의미하지 않습니다. 대화는 활발히 사용되고 있지 않을 경우에만 재생됩니다. 확약 제어가 사용되는 경우, COMMIT나 ROLLBACK 조작은 DDM 대화가 재생될 수 있기 전에 실행되어야 할 수도 있습니다.

프로그램에 의해 사용된 오브젝트 표시

프로그램 참조 표시(DSPPGMREF) 명령을 사용하여 프로그램이나 SQL 패키지에 의해 사용되는 표, 자료 영역 및 다른 프로그램을 판별할 수 있습니다. 이 정보는 SQL 패키지와 컴파일된 프로그램에만 사용할 수 있습니다. 정보는 데이터베이스 출력 파일에 표시, 인쇄 또는 기록될 수 있습니다.

6. 종료하는 TCP/IP 대화에 대하여 어플리케이션 서버(AS) 작업은 일반적으로 사전시작 작업이며, 보통 종료되지 않고 재순환됩니다.

프로그램이나 패키지가 작성되어 있을 때, 이 프로그램이나 패키지 안에 사용되는 특정 오브젝트에 대한 정보가 저장됩니다. 이 때, 이 정보는 프로그램 참조 표시(DSPPGMREF) 명령으로 사용하도록 제공됩니다. 검색되는 정보는 다음을 포함할 수 있습니다.

- 프로그램이나 패키지의 이름과 텍스트 설명
- 프로그램이나 패키지를 포함하는 라이브러리나 컬렉션의 이름
- 프로그램 패키지에 의해 참조되는 오브젝트 수
- 서버 오브젝트의 규정된 이름
- 정보 검색 날짜
- 참조한 오브젝트의 오브젝트 유형

파일과 표에 대하여, 해당 레코드에는 다음의 추가 필드가 포함되어 있습니다.

- 프로그램이나 패키지에 들어 있는 파일이나 표의 이름(프로그램이나 패키지를 작성할 때 대체가 실행되었으면 가능한 한 서버 오브젝트명과 다른 이름으로)

주: 모든 대체는 어플리케이션 리퀘스터(AR)에만 적용됩니다.

- 파일이나 표에 관한 프로그램이나 패키지의 사용(입력, 출력, 갱신, 특기하지 않음 또는 이 4가지의 조합)
- 해당 사항이 있는 경우, 참조한 레코드 형식의 수
- 파일이나 표에 의해 사용된 레코드 형식명 및 이것의 레코드 형식 레벨 ID
- 각 형식에 대해 참조된 필드의 수

오브젝트를 프로그램 안에서 보여 줄 수 있기 전에 사용자는 해당 프로그램에 대해 *USE를 갖고 있어야 합니다. 또한 라이브러리 규정자에 의해 지정되는 라이브러리 중에서 해당 프로그램에 대해 사용자가 읽기 권한을 갖고 있는 라이브러리만을 탐색합니다.

표 5는 고급 언어와 유틸리티가 정보를 저장하는 오브젝트를 나타내고 있습니다.

표 5. 고급 언어가 오브젝트에 대한 정보를 저장하는 방법

언어	파일	프로그램	자료 영역	주 참조
CL	예	예	예	1
COBOL/400* 언어	예	예	아니오	2
PL/I	예	예	N/A	2
RPG/400* 언어	예	아니오	예	3
DB2 UDB SQL	예	N/A	N/A	4

주:

1. 파일, 프로그램 또는 자료 영역에 관련하는 모든 서버 명령은 정보가 CL 프로그램 안에 컴파일되어 있을 때 정보가 저장되어 있어야 하는 명령 정의에서 지정합니다. 변수가 사용되면 변수명은 오브젝트명으로 사용됩니다(예를 들어, &FILE). 표현식이 사용되면 오브젝트명이 *EXPR로서 저장됩니다. 또한 사용자가 정의한 명령은 해당 명령에 지정된 파일, 프로그램 또는 자료 영역에 대한 정보를 저장할 수 있습니다. iSeries Information Center의 CL 프로그래밍 주제에 있는 FILE, PGM 및 PARM이나 ELEM 명령의 DTAARA 매개변수에 관한 설명을 참조하십시오.
2. 프로그램명은 리터럴이 프로그램명에 사용될 때에만 저장되고(이것은 정적 호출입니다. 예를 들어, CALL 'PGM1'), COBOL/400® ID가 프로그램명에 사용될 때에는 저장되지 않습니다(이것은 동적 호출입니다. 예를 들어, CALL PGM1).
3. 로컬 자료 영역(LDA)의 사용은 저장되지 않습니다.
4. SQL 패키지에 대한 정보

저장된 파일 정보는 사용의 유형에 대한 항목(수)을 포함합니다. 프로그램 참조 표시(DSPPGMREF) 명령 (OUTFILE 매개변수를 사용하여 빌드됨)의 데이터베이스 파일 출력에서, 오브젝트 당 하나의 입력만 가능하므로 조합이 사용됩니다. 예를 들어, 7로 코드화된 파일은 입력, 출력 및 갱신에 사용됩니다.

코드 의미

- 1 입력
- 2 출력
- 3 입력 및 출력
- 4 갱신
- 8 지정하지 않음

자세한 정보는 예: 프로그램 참조 표시를 참조하십시오.

예: 프로그램 참조 표시

어플리케이션 리퀘스터(AR) 프로그램에 의해 사용되는 오브젝트를 참조하기 위해 다음과 같이 명령을 입력할 수 있습니다.

```
DSPPGMREF PGM(SPIFFY/PARTS1) OBJTYPE(*PGM)
```

리퀘스터에서, 프로그램에 의해 사용되는 모든 콜렉션과 표에 관한 리스트를 얻을 수 있으나 이것이 위치하는 관계형 데이터베이스에서 참조할 수는 없습니다. 이것은 복수 관계형 데이터베이스에 위치할 수도 있습니다. 명령으로부터 출력은 데이터베이스 파일이나 표시된 스폴 파일을 찾아갈 수 있습니다. 출력은 다음과 같습니다.

```
File . . . . . : QPDSPPGM                   Page/Line  1/1
Control . . . . .                   Columns     1 - 78
Find . . . . .
```

```
3/29/92                   Display Program References
DSPPGMREF Command Input
Program . . . . . : PARTS1
Library . . . . . : SPIFFY
Output . . . . . : *
Include SQL packages . . . . . : *YES
Program . . . . . : PARTS1
Library . . . . . : SPIFFY
Text 'description'. . . . . : Check inventory for parts
Number of objects referenced . . . . . : 3
Object . . . . . : PARTS1
Library . . . . . : SPIFFY
Object type . . . . . : *PGM
Object . . . . . : QSROUTE
Library . . . . . : *LIBL
Object type . . . . . : *PGM
Object . . . . . : INVENT
Library . . . . . : SPIFFY
Object type . . . . . : *FILE
File name in program . . . . . :
File usage . . . . . : Input
```


어플리케이션 서버(AS) SQL 패키지에 의해 사용되는 오브젝트를 참조하기 위해 다음과 같이 명령을 입력할 수 있습니다.

```
DSPPGMREF PGM(SPIFFY/PARTS1) OBJTYPE(*SQLPKG)
```

명령으로부터 출력은 데이터베이스 파일이나 표시된 스푼 파일을 찾아갈 수 있습니다. 출력은 다음과 같습니다.

```
File . . . . . : QPDSPPGM                               Page/Line  1/1
Control . . . . .                               Columns   1 - 78
Find . . . . .
```

```
3/29/92                Display Program References
DSPPGMREF Command Input
Program . . . . . : PARTS1
Library . . . . . : SPIFFY
Output . . . . . : *
Include SQL packages . . . . . : *YES
SQL package . . . . . : PARTS1
Library . . . . . : SPIFFY
Text 'description'. . . . . : Check inventory for parts
Number of objects referenced . . . . . : 1
Object . . . . . : INVENT
Library . . . . . : SPIFFY
Object type . . . . . : *FILE
File name in program . . . . . :
File usage . . . . . : Input
```

분산 관계형 데이터베이스에서 콜렉션 드롭(Drop)

저널 리시버를 포함하는 콜렉션을 삭제하려고 시도하면 조회 메시지가 어플리케이션 서버(AS) 작업에 대한 QSYSOPR 메시지 대기행렬로 송신되는 원인이 될 수 있습니다. AS와 어플리케이션 리퀘스터(AR) 작업은 이 조회가 응답될 때까지 대기합니다.

메시지 대기행렬에 나타나는 메시지는 다음과 같습니다.

CPA7025

(라이브러리) 안에 리시버(이름)는 완전히 저장되지 않습니다. (I C)

AR 작업은 대기 중일 때에 어플리케이션이 보류되어 있는 것처럼 나타날 수 있습니다. 사용자의 AR 작업이 예상보다 오랜 시간 동안 대기 중일 때 다음을 고려하십시오.

- 조회 메시지가 QSYSOPR 메시지 대기행렬로 송신되었고 응답처리가 필요한지 알아 보십시오.
- AS가 서버 응답 리스트를 사용하여 메시지에 응답하게 하십시오.

주: 일단 어플리케이션이 명백한 '보류'(hung)상태에 있으면 정지된 것입니다. 이는 저널 리시버를 오브젝트 이동(MOVOBJ) 명령으로 다른 라이브러리로 이동시킬 수 없기 때문입니다. 또 다른 라이브러리에 저장되거나 복원될 수 없습니다. 사용자가 할 수 있는 일은 저널 리시버 작성(CRTJRNRCV) 명령으로 다른 라이브러리에 새로운 저널 리시버를 작성하고, 저널 변경(CHGJRN) 명령으로 이를 저널에 접속시키는 것입니다. JRNRCV(*GEN) 매개변수와 저널 변경(CHGJRN) 명령으로 시스템에서 작성된 어떠한 새 저널 리시버도 새 라이브러리에 작성됩니다. 저널이 저장될 때 접속된 리시버가 다른 라이브러리에 있으면, 저장되어 있는 버

전의 저널이 복원될 때 새 저널 리시버도 다른 라이브러리에 작성됩니다. 저널링에 대한 자세한 정보는 iSeries Information Center의 저널 관리 주제를 참조하십시오.

서버 응답 리스트를 사용하여 메시지에 대한 AS 응답을 받는 것은 현재 보류되어 있다고 나타나는 작업을 변경하거나 서버에서 실행 중인 모든 AS 작업에 대해 작업 설명을 변경함으로써 이루어질 수 있습니다. 그러나, 가장 먼저 응답 리스트 입력 추가(ADDRPYLE) 명령을 사용하여 메시지 CPA7025에 대해 어플리케이션 서버(AS) 응답 리스트에 입력을 추가해야 합니다.

```
ADDRPYLE SEQNBR(...) MSGID(CPA7025) RPY(I)
```

현재 AS에서 실행 중인 작업에 대해 작업 설명을 변경하려면 리모트 명령 제출(SBMRMTCMD) 명령을 사용하십시오. 다음의 예는 캔자스 시티 지역에서 한 서버의 데이터베이스 관리자가 KC105(TEST/KC105TST) DOM 파일에 의해 주소지정된 서버에서 작업 설명을 변경하는 방법을 나타내고 있습니다.

```
SBMRMTCMD CMD('CHGJOB JOB(KC105ASJOB) INQMSGRPY(*SYSRPLY)')  
DDMFILE(TEST/KC105TST)
```

작업 설명 변경(CHGJOB) 명령을 사용하여 이 상황이 AS에서 보다 영구적으로 일어나지 않도록 예방함으로써 해당 작업 설명을 사용하는 작업이 서버 응답 리스트를 사용할 수 있습니다. 다음의 예는 이 명령이 같은 AS에서 입력되는 방법을 나타내고 있습니다.

```
CHGJOB JOB(KC105ASJOB) INQMSGRPY(*SYSRPLY)
```

이 메소드는 주의하여 사용되어야 합니다. CPA7025를 서버에 추가하면, 응답 리스트는 서버 작업 리스트를 사용하는 모든 작업에 영향을 줍니다. 또한 작업 리스트를 변경하면 특정의 작업 설명을 사용하는 모든 작업에 영향을 줍니다. AS 작업에 대해 별도의 작업 설명을 작성하려고 할 수도 있습니다. 작업 설명 작성에 대한 추가 정보는 iSeries Information Center의 작업 관리 주제를 참조하십시오.

분산 관계형 데이터베이스에서 작업 계정

서버를 사용하고 있는 사용자와 이 사용자가 사용하고 있는 서버 자원을 판별할 수 있도록 iSeries 서버에서 작업 사용 계정 기능이 자료를 수집합니다. 일반적인 작업 사용 계정은 서버에서 실행 중인 작업과 처리 장치, 프린터, 표시장치 및 데이터베이스와 통신 기능과 같이 사용되는 자원을 자세히 설명합니다.

작업 사용 계정은 선택 가능하며 서버에서 설정되어야 합니다. 서버에서 자원 사용 계정을 설정하려면, 다음을 수행하십시오.

1. 저널 리시버 작성(CRTJRNRCV) 명령을 사용하여 저널 리시버를 작성하십시오.
2. 저널 작성(CRTJRN) 명령을 사용하여 QSYS/QACGJRN이라는 이름의 저널을 작성하십시오. 사용자는 QSYS/QACGJRN이라는 이름을 사용해야 하고 이 저널을 작성하려면 QSYS에 항목을 추가할 권한을 갖고 있어야 합니다. 이 명령에서 이전 단계에 작성된 저널 리시버명을 지정하십시오.
3. 시스템 값에 대한 작업(WRKSYSVAL) 또는 시스템 값 변경(CHGSYSVAL) 명령을 사용하여 계정 레벨 서버 값 QACGLVL을 변경하십시오.

시스템 값 변경(CHGSYSVAL) 명령에서 VALUE 매개변수는 작업 사용 계정 저널 항목이 생산되는 때를 판별합니다. *NONE 값은 서버가 작업 사용 계정 저널에서 어떠한 항목도 생산하지 못한다는 뜻입니다.

다. *JOB 값은 서버가 작업(JB) 저널 항목을 생산한다는 뜻입니다. *PRINT 값은 인쇄된 각 파일에 대한 직접 인쇄(DP) 또는 스푼 인쇄(SP) 저널 항목을 생산합니다.

작업이 시작될 때 작업 설명이 작업에 기록됩니다. 작업 설명 오브젝트는 ACGCDE(계정 코드) 매개변수에 대한 값을 포함하며, 이 값은 계정 코드나 디폴트 값 *USRPRF일 수도 있습니다. *USRPRF를 지정하면, 작업의 사용자 프로파일에 있는 계정 코드가 사용됩니다.

사용자 프로파일 작성(CRTUSRPRF) 명령이나 사용자 프로파일 변경(CHGUSRPRF) 명령에서 계정 코드 매개변수 ACGCDE를 사용하여 사용자 프로파일에 계정 코드를 추가할 수 있습니다. 작업 설명 작성(CRTJOBDD) 명령이나 작업 설명 변경(CHGJOBDD) 명령에서 ACGCDE 매개변수에 대해 원하는 계정 코드를 지정함으로써 특정 작업에 대해 계정 코드를 변경할 수 있습니다.

작업 사용 계정 저널이 설정되어 있을 때, 시스템 값 변경(CHGSYSVAL) 명령이 실행된 후에 작업 사용 계정 항목은 서버에 입력한 그 다음 작업을 시작함으로써 저널 리시버에 위치지정됩니다.

저널 표시(DSPJRN) 명령에서 OUTFILE 매개변수를 사용하여 사용자가 처리할 수 있습니다.

작업 사용 계정에 대한 자세한 정보는 iSeries Information Center의 작업 관리 주제를 참조하십시오.

TCP/IP 서버 관리

이 섹션은 TCP를 통한 소켓을 사용하여 통신하는 DRDA/DDM 서버 작업을 관리하는 방법에 대해서 설명합니다. 이것은 서버가 실행하는 서브시스템, 서버에 영향을 주는 오브젝트 및 오브젝트 자원을 관리하는 방법에 대해서 설명합니다.

OS/400 프로그램으로 출하되는 DRDA/DDM TCP/IP 서버는 일반적으로 올바르게 작업하기 위해 사용자의 기존 시스템 구성에 대해서 어떠한 변경도 요구하지 못합니다. 이 서버는 사용자가 OS/400을 설치할 때 설정되고 구성됩니다. 언젠가 사용자는 시스템이 사용자의 요구를 보다 잘 만족시키고, 문제점을 해결하고, 서버의 성능을 향상시키고 또는 서버에서 간단하게 작업을 볼 수 있도록 서버 작업을 관리하는 방법을 변경하려고 할 수도 있습니다. 그와 같이 변경하고 사용자의 처리 요구사항을 만족시키려면, 시스템의 어느 부분이든 영향을 주는 오브젝트와 해당 오브젝트를 변경하는 방법에 대해서 알아야 합니다.

이 섹션에서는 상급 레벨에서 서버 작업(jobs)에 대한 작업을 위해 이해해야 하는 몇 가지 작업 관리 개념 및 해당 개념과 오브젝트가 서버에 관련하는 방법에 대해서 설명합니다. iSeries 서버 관리 방법을 완전히 이해하려면, 이 섹션을 계속하기 전에 iSeries Information Center의 작업 관리 주제를 주의깊게 검토해야 합니다. 그 다음으로, 이 섹션은 TCP/IP 서버를 관리할 수 있는 방법과 서버를 시스템의 나머지 부분과 일치시키는 방법에 대해서 나타내고 있습니다.

자세한 정보는 다음 주제를 참조하십시오.

- DRDA TCP/IP 서버 전문 용어
- DDM용 TCP/IP 통신 지원 개념
- DRDA/DDM 서버 작업
- DDM 서버 작업 서브시스템 구성

- 서버 작업 식별

DRDA TCP/IP 서버 전문 용어

같은 서버 소프트웨어가 iSeries용 DB2 UDB에 대한 DDM 및 DRDA TCP/IP 액세스에 사용됩니다. 간결하게 하기 위해, 다음의 논의에서 *DRDA/DDM 서버*라는 용어 대신에 *DDM 서버*라는 용어를 사용하겠습니다. 그러나, 때로는 문맥에서 규정자의 사용이 불필요할 때에 *TCP/IP 서버*, *DRDA 서버* 또는 간단하게 *서버*로서 언급할 수도 있습니다.

DDM 서버는 두 개 또는 그 이상의 작업으로 구성되며, 그 중 하나는 다른 작업(jobs)으로의 연결 요구와 작업 지정 작업에 대해서 청취하기 때문에 말하자면 DDM 리스너입니다. 초기에 구성되는 다른 작업은 초기 연결이 이루어진 후에 DRDA 또는 DDM 클라이언트로부터 받은 요구에 대해 서비스를 제공하는 사전시작 작업입니다. 연관되는 모든 작업, 리스너 및 서버 작업의 집합을 공동으로 *DDM 서버*라고 언급합니다.

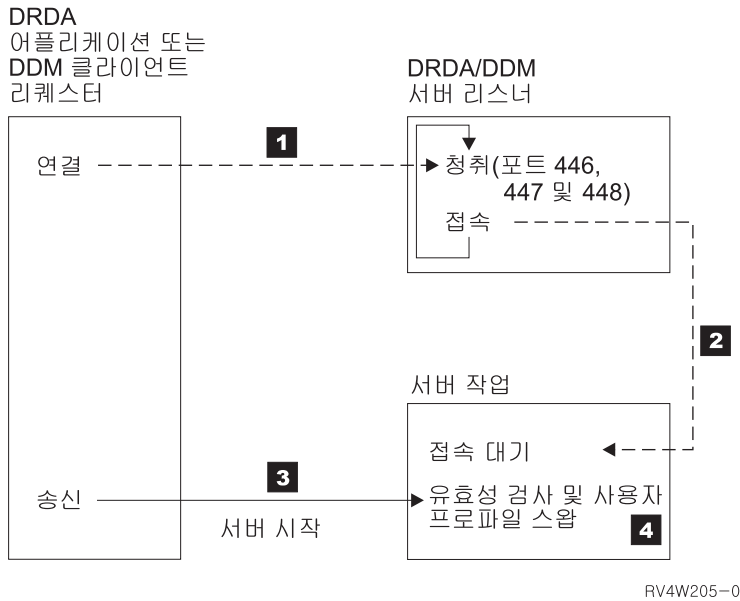
클라이언트라는 용어는 DRDA 어플리케이션 환경에서 *DRDA 어플리케이션 리퀘스터*(또는 AR)와 교환할 수 있게 됩니다. 클라이언트는 DDM(분산파일 관리) 어플리케이션 환경에서 *DDM 소스 시스템*과 교환할 수 있게 사용됩니다.

서버라는 용어는 DRDA 어플리케이션 환경에서 *DRDA 어플리케이션 서버*(또는 AS)와 교환할 수 있게 사용됩니다. 클라이언트는 DDM(분산 파일 관리) 어플리케이션 환경에서 *DDM 목표 시스템*과 교환할 수 있게 사용됩니다. (어떤 문맥에서는 iSeries 시스템(하드웨어)이 서버라고도 불리고 iSeries 서버라고도 불리는 점을 주의하십시오.)

DDM용 TCP/IP 통신 지원 개념

DRDA와 DDM에 의해 사용되는 TCP/IP 통신 지원에 특수하게 속하는 여러 개념이 있습니다. 이 개념은 여기에서 자세하게 설명되어 있습니다.

TCP/IP를 통한 DRDA 또는 DDM 연결 설정



RV4W205-0

그림 10. DRDA/DDM TCP/IP 서버

TCP/IP 통신 지원을 사용하는 DDM 서버를 개시하기 위해, DRDA 어플리케이션 리퀘스터나 DDM 소스 시스템이 잘 알려진 포트 번호인 446 또는 447에 연결됩니다. 또한 DDM 서버는 포트 448에서도 청취하지만 단지 SSL(소스 소켓층) 연결에 사용하기 위한 것일 뿐 아직까지 iSeries용 DB2 UDB 어플리케이션 리퀘스터나 DDM 클라이언트에 의해 지원되지는 않습니다. **1**. DDM 리스너 프로그램은 클라이언트의 연결 요구를 청취하고 받아들이기 위해(TCP/IP 서버 시작(STRTCPSVR SERVER(*DDM))을 사용하여) 시작되었어야 합니다. DDM 리스너는 이 연결 요구를 받아들일 때에 클라이언트의 연결을 DDM 서버 작업 **2**에 접속하도록 내부적인 요구를 보냅니다. 이 서버 작업은 사전시작 작업이거나, 사용자가 QUSRSYS 또는 사용자 정의 서브시스템에서 QRWTSRVR 사전시작 작업 항목을 제거했다면(사전시작 작업이 사용되지 않는 경우), 클라이언트 연결 요구가 처리되었을 때 제출되는 일괄처리 작업일 수도 있습니다. 서버 작업은 클라이언트와 그 이상의 통신을 처리합니다.

초기에 생기는 자료 교환에는 서버 작업이 **3**을 실행하게 하는 사용자 프로파일을 식별하는 요구가 포함되어 있습니다. 일단 사용자 프로파일과 암호가(사용자 프로파일 id로 송신되는 경우) 사용 가능하게 되면, 서버 작업은 CCSID와 같이 사용자 프로파일 **4**에 대해 정의된 속성을 사용하도록 작업을 변경할 뿐만 아니라 이 사용자 프로파일로 교체합니다.

리스너 프로그램으로 연결, 클라이언트 연결을 서버 작업에 접속, 자료 교환 및 사용자 프로파일과 암호의 유효화 등의 기능은 APPC 프로그램 시작 요구가 처리될 때 실행되는 해당 기능과 호환될 수 있습니다.

DRDA/DDM 리스너 프로그램

DDM 리스너 프로그램은 일괄처리 작업에서 실행됩니다. 이 리스너와 실제의 서버작업 사이에 1 대 다수의 관계가 성립합니다. 한 리스너와 잠재적으로 다수의 DDM 서버 작업이 있습니다. 서버 작업은 일반적으로 사전시작 작업입니다. 리스너 작업은 QSYSWRK 서브시스템에서 실행됩니다.

DDM 리스너는 클라이언트 어플리케이션이 인바운드 연결을 처리하고 라우팅함으로써 연관되는 서버 작업과 TCP/IP 연결을 설치할 수 있게 합니다. 일단 클라이언트가 서버 작업과 통신을 설치하면 통신 지속 기간에 클라이언트와 리스너 간에 그 이상의 연관은 없습니다.

DDM 리스너는 DRDA 어플리케이션 리퀘스터와 DDM 소스 시스템이 DDM TCP/IP 서버와 연결을 설치할 수 있도록 활동 중이어야 합니다. 사용자는 DRDA 리스너가 DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령을 사용하거나 iSeries Navigator를 통해 자동으로 시작되도록 요구할 수 있습니다. iSeries Navigator에서 DDM 설정, 네트워크 -> 서버 -> TCP/IP로 탐색하십시오. 이것은 TCP/IP가 시작될 때 리스너가 시작되게 합니다. DRDA 리스너를 시작 중일 때 QSYSWRK 서브시스템과 TCP/IP가 둘 다 활동 중이어야 합니다.

STRTCPSVR(TCP/IP 서버 시작) CL 명령

TCP/IP 서버 시작(STRTCPSVR) 명령은 *DDM 또는 *ALL의 SERVER 매개변수 값을 갖고 있으며 리스너를 시작하는 데 사용됩니다.

DDM 리스너 제한사항: 한 번에 한 DDM 리스너만 활동할 수 있습니다. 리스너가 이미 활동 중일 때 이것을 시작하라는 요구는 명령 제출자에 대한 정보용 메시지로 끝나게 됩니다.

주: DDM 서버는 QUSER 암호가 만료되었으면 시작하지 않습니다. 암호 만기 간격은 QUSER 프로파일에 대해 *NOMAX로 설정되는 것이 좋습니다. 이 값을 사용하면 암호가 만료되지 않습니다.

예: TCP/IP 서버 시작(STRTCPSVR) CL 명령:

예 1: 전체 TCP/IP 서버 시작

```
STRTCPSVR SERVER(*ALL)
```

이 명령은 DDM 서버를 포함하여 TCP/IP 서버 전부를 시작합니다.

예 2: DDM TCP/IP 서버만 시작

```
STRTCPSVR *DDM
```

이 명령은 DDM TCP/IP 서버만 시작합니다.

TCP/IP 서버 종료(ENDTCPSVR) CL 명령

TCP/IP 서버 종료(ENDTCPSVR) 명령은 DDM 서버를 종료합니다.

DDM 리스너가 종료되고 클라이언트 어플리케이션과 활성 연결되어 있는 연관된 서버 작업이 있으면, 해당 서버 작업은 클라이언트 어플리케이션과 통신이 종료될 때까지 활성 상태를 유지합니다. 그러나, 리스너가 다시 시작될 때까지 클라이언트 어플리케이션으로부터 다음의 연결 요구가 실패됩니다.

TCP/IP 서버 제한사항 종료: TCP/IP 서버 종료(ENDTCPSVR) 명령은 DDM 리스너가 활동 중이 아닐 때 이것을 종료하는 데 사용되면 진단 메시지가 발행됩니다. (ENDTCPSVR) SERVER(*ALL) 명령이 발행될 때 리스너가 활동 중이 아니면 이와 같은 진단 메시지는 송신되지 않습니다.

TCP/IP 서버 종료 예:

예 1: 전체 TCP/IP 서버 종료

ENDTCPSVR *ALL

이 명령은 활성 TCP/IP 서버를 전부 종료합니다.

예 2: DDM 서버만 종료

ENDTCPSVR SERVER(*DDM)

이 명령은 DDM 서버를 종료합니다.

iSeries Navigator에서 DDM 리스너 시작

또 DDM 리스너는 iSeries Navigator를 사용하여 관리할 수 있으며 이는 iSeries Access 제품군의 일부입니다. 이는 다음 경로, 네트워크 -> 서버 -> TCP/IP 디렉토리를 따라 수행됩니다.

DRDA/DDM 서버 작업

DDM이 있는 서브시스템 설명과 사전시작 작업 항목

서브시스템 설명은 서브시스템을 입력하는 작업 방법, 작업 장소, 작업량 및 서브시스템이 해당 작업을 실행하는 데 사용하는 자원에 대해 정의합니다. 다음 단락에는 QUSRWRK(또는 V5R2 이전에는 QSYSWRK) 서브시스템 설명에서 사전시작 작업 항목이 DDM 서버에 어떤 영향을 주는지에 대해 설명합니다.

사전시작 작업은 어플리케이션 리퀘스터(AR)가 서버와 통신을 개시하기 전에 실행하기 시작하는 일괄처리 작업입니다. 사전시작 작업은 서브시스템 설명에 있는 사전시작 작업 항목을 사용하여 이 작업이 시작될 때 사용하는 프로그램 클래스와 기억장치 풀을 판별합니다. 사전시작 작업 항목 내에 서브시스템이 사전시작 작업의 풀을 작성하고 관리하는 데 사용하는 속성을 지정하여야 합니다.

사전시작 작업은 서버에 연결을 개시할 때 증진된 성능을 제공합니다. 사전시작 작업 항목은 서브시스템 내에서 정의됩니다. 사전시작 작업은 해당 서브시스템이 시작되거나 사전시작 작업 시작(STRPJ)과 사전시작 작업 종료(ENDPJ) 명령으로 제어될 수 있을 때 활성화됩니다.

DRDA/DDM 사전시작 작업

사전시작 작업(활동 사전시작 작업 표시(DSPACTPJ) 명령과 같이)에 속하는 서버 정보는 이것이 TCP/IP 연결 요구의 결과로서 시작된 사전시작 작업에 속할 수도 있다 하더라도 '프로그램 시작 요구'라는 용어를 독립적으로 사용하여 사전시작 작업을 시작하도록 하는 요구를 나타냅니다.

다음의 리스트에는 DDM TCP/IP 서버에 대한 초기 구성값을 갖고 있는 사전시작 작업 항목 속성이 들어 있습니다. 이 속성은 사전시작 작업 항목 변경(CHGPJE) 명령으로 변경될 수 있습니다.

- 서브시스템 설명. 사전시작 작업 항목을 포함하는 서브시스템은 V5R2의 QUSRWRK입니다. 이전의 릴리스에서는 QSYSWRK였습니다.
- 프로그램 라이브러리와 이름. 사전시작 작업이 시작될 때 호출되는 프로그램은 QSYS/QRWTSRVR입니다.
- 사용자 프로파일. 작업이 소속되어 실행하는 사용자 프로파일은 QUSER입니다. 이것은 바로 작업이 사용자 프로파일로서 보이는 이유입니다. 클라이언트로부터 서버로 연결 요구를 수신할 때 사전시작 작업 기능은 해당 요구 시 수신되는 사용자 프로파일로 교체합니다.

- 작업명. 작업이 시작될 때 작업명은 QRWTSRVR입니다.
- 작업 설명. 사전시작 작업에 사용되는 작업 설명은 *USRPRE입니다. 사용자 프로파일이 QUSER이므로 QUSER의 작업 설명이 무엇이든간에 사용자 프로파일은 QUSER이 됩니다. 그러나, 작업의 속성은 사용자 ID와 암호가(존재하는 경우) 확인된 후에 요구하는 사용자의 작업 설명과 일치하도록 변경됩니다.
- 시작 작업. 이것은 서브시스템이 시작될 때 사전시작 작업이 자동으로 시작되는지 여부를 표시합니다. 이 사전시작 작업 항목은 시작 작업값 *YES로 출하됩니다. 시스템 IPL이 실행될 때 불필요한 작업이 시작되는 것을 막기 위해 *NO로 변경할 수 있습니다. 주: DDM 서버 작업이 실행 중이 아니고 DDM 리스너 작업이 일괄처리라도 즉각적인 DDM 서버 작업은 QSYSWRK 서브시스템 하에서 실행될 것입니다.
- 초기 작업 수. 초기에 구성된 대로, 서브시스템이 시작될 때 시작되는 작업 수는 1입니다. 이 값은 사용자의 특정 환경과 요구에 따라 조정될 수 있습니다.
- 임계값(threshold). 사전시작 작업 항목에 대해 사용할 수 있는 사전시작 작업 수의 최소값은 1입니다. 이 임계값에 도달했을 때 추가 사전시작 작업은 자동으로 시작됩니다. 이것은 풀 안에 일정한 수의 작업을 유지보수하기 위해 사용됩니다.
- 추가 작업 수. 임계값에 도달할 때 시작되는 추가 사전시작 작업의 수는 초기에 2로 구성됩니다.
- 최대 작업 수. 이 항목에 대해 활동할 수 있는 사전시작 작업의 최대수는 *NOMAX입니다.
- 사용 최대 수. 작업 사용의 최대 수는 200으로 설정됩니다. 이 값은 서버를 시작하도록 200개의 요구가 처리된 후에 사전시작 작업이 종료됨을 나타냅니다. 특정 상황에서, 사용자는 TCP/IP 서버가 제대로 기능할 수 있도록 하기 위해 MAXUSE 매개변수를 1로 설정해야 할 수도 있습니다. 서버가 일정한 ILE 저장 프로시저어를 실행할 때, 파손된 오브젝트에 대한 포인터가 사전시작 작업 환경에 남아 있을 수도 있기 때문에 그 다음의 사전시작 작업 사용에서 MCH3402 예외 발생의 원인이 됩니다. V5R2에서 이 가능성을 최소화하기 위하여 OS/400에 변경이 가해졌습니다.
- 대기 작업. *YES를 설정하면, 작업 최대 수에 도달 하는 경우, 클라이언트 연결 요구가 사용할 수 있는 서버 작업을 기다리게 합니다.
- 풀 IC. 이 사전시작 작업이 실행하는 서브시스템 ID는 1로 설정됩니다.
- 장치 등급. 사전시작 작업이 속하여 실행하는 클래스의 이름과 라이브러리는 QSYS/QSYSCLS20으로 설정됩니다.

사전시작 작업 항목에 대한 시작 작업값이 *YES로 설정되어 있고 나머지 값이 초기의 설정값으로 제공된 그 대로일 때, 각 사전시작 작업 항목에 대해 다음의 결과가 생깁니다.

- 서브시스템이 시작될 때 하나의 사전시작 작업이 시작됩니다.
- 첫 번째 클라이언트 연결 요구가 TCP/IP 서버에 대해 처리될 때 초기의 작업이 사용되고 임계값을 초과합니다.
- 추가 작업은 사전시작 작업 항목 안에 정의된 수에 기초한 서버에 대하여 시작됩니다.
- 사용할 수 있는 작업 수는 1 이하로 내려가지 않습니다.
- 서브시스템은 정기적으로 사용하지 않는 풀 안에 들어 있는 사전시작 작업 수를 검사하고, 초과 작업을 종료합니다. 서브시스템은 언제나 적어도 초기 작업 매개변수에 지정되어 있는 수의 사전시작 작업을 남겨 둡니다.

사전시작 작업 모니터링: 사전시작 작업은 사용 중인 사전시작 작업 표시(DSPACTPJ) 명령을 사용하여 모니터링될 수 있습니다.

(DSPACTPJ) 명령은 다음의 정보를 제공합니다.

- 현재 사전시작 작업 수
- 평균 사전시작 작업 수
- 최대 사전시작 작업 수
- 사용 중인 현재 사전시작 작업 수
- 사용 중인 평균 사전시작 작업 수
- 사용 중인 최대 사전시작 작업 수
- 대기 중인 현재 연결 요구 수
- 대기 중인 평균 연결 요구 수
- 대기 중인 최대 연결 요구 수
- 평균 대기 시간
- 채택된 연결 요구 수
- 거부된 연결 요구 수

사전시작 작업 관리: 활동 중인 사전시작 작업에 대해 제시된 정보는 활동 중인 사전시작 작업 표시 화면에서 F5 키를 누름으로써 화면정리될 수 있습니다. 특히 중요한 정보는 프로그램 시작 요구에 대한 내용입니다. 이 정보는 사용자가 사전시작 작업의 사용 가능한 수에 대한 변경 여부를 사용자에게 알려 줄 수 있습니다. 프로그램 시작 요구가 사용 가능한 사전시작 작업에 대해 대기 중임을 나타내는 정보에 대하여, 사용자는 사전시작 작업 항목 변경(CHGJPJE) 명령을 사용하여 사전시작 작업을 변경할 수 있습니다.

프로그램 시작 요구가 만족할 만큼의 속도로 작용하고 있지 않으면, 사용자는 다음 중 어떤 조합으로든 실행할 수 있습니다.

- 임계값을 증가시키십시오.
- INLJOBS(초기 작업 수) 매개변수 값을 증가시키십시오.
- ADLJOBS(추가 작업 수) 매개변수 값을 증가시키십시오.

중요한 것은 서버 작업을 시작하도록 송신되는 모든 요구에 대해 사용 가능한 사전시작 작업이 있는지 확인하는 일입니다.

사전시작 작업 항목 제거: 사용자가 사전시작 작업 기능을 사용하는 서버를 원하지 않는다고 결정하면, 다음과 같이 실행해야 합니다.

1. 사전시작 작업 종료(ENDPJ) 명령을 사용하여 사전시작된 작업을 종료하십시오.

(ENDPJ) 명령으로 종료된 사전시작 작업은 시작 작업 *YES가 해당 사전시작 작업 항목에 지정되어 있는 경우, 해당 서비스시스템이 시작되고 난 다음에 시작됩니다. 사전시작 작업을 끝내기만 하고 그 다음 단계를 실행하지 않으면, 특정의 서버를 시작하도록 하는 요구는 어느 것이든 장애가 발생합니다.

2. 사전시작 작업 항목 제거(RMVPJE) 명령을 사용하여 서브시스템 설명에 들어 있는 사전시작 작업 항목을 제거하십시오.

(RMVPJE) 명령으로 제거된 사전시작 작업 항목은 서브시스템 설명에서 영구적으로 제거됩니다. 일단 입력이 제거되면 서버에 대한 새로운 요구는 성공적으로 실행되지만 작업 개시 실행에서 오버헤드를 초래합니다.

라우팅 항목: OS/400 작업이 서브시스템으로 라우트될 때, 이것은 서브시스템 설명에 들어 있는 라우팅 입력을 사용하여 실행됩니다. QSYSWRK 서브시스템에서 리스너 작업에 대한 라우팅 입력은 OS/400이 설치된 후에 있습니다. 이 작업은 QUSER 사용자 프로파일하에서 시작되고 QSYSNOMXA 작업 대기행렬이 사용됩니다.

V5R2 이전에는 서버 작업은 QSYSWRK 서브시스템에서 실행되었습니다. V5R2에서는 서버 작업이 QUSRWRK에서 디폴트로 실행됩니다. 서버 작업의 특성은 OS/400을 사용하여 자동으로 구성된 사전시작 작업 입력으로부터 취해집니다. 이 입력이 제거되어 서버에 대해 사전시작 작업이 사용되지 않으면, 서버 작업은 이것과 일치하는 리스너 작업의 특성을 사용하여 시작됩니다.

다음은 리스너 작업에 대하여 QSYSWRK 서브시스템에 초기 구성을 제공합니다.

서브시스템	QSYSWRK
작업 대기행렬	QSYSNOMAX
사용자	QUSER
라우팅 자료	QRWTLSTN
작업명	QRWTLSTN
장치 등급	QSYSCLS20

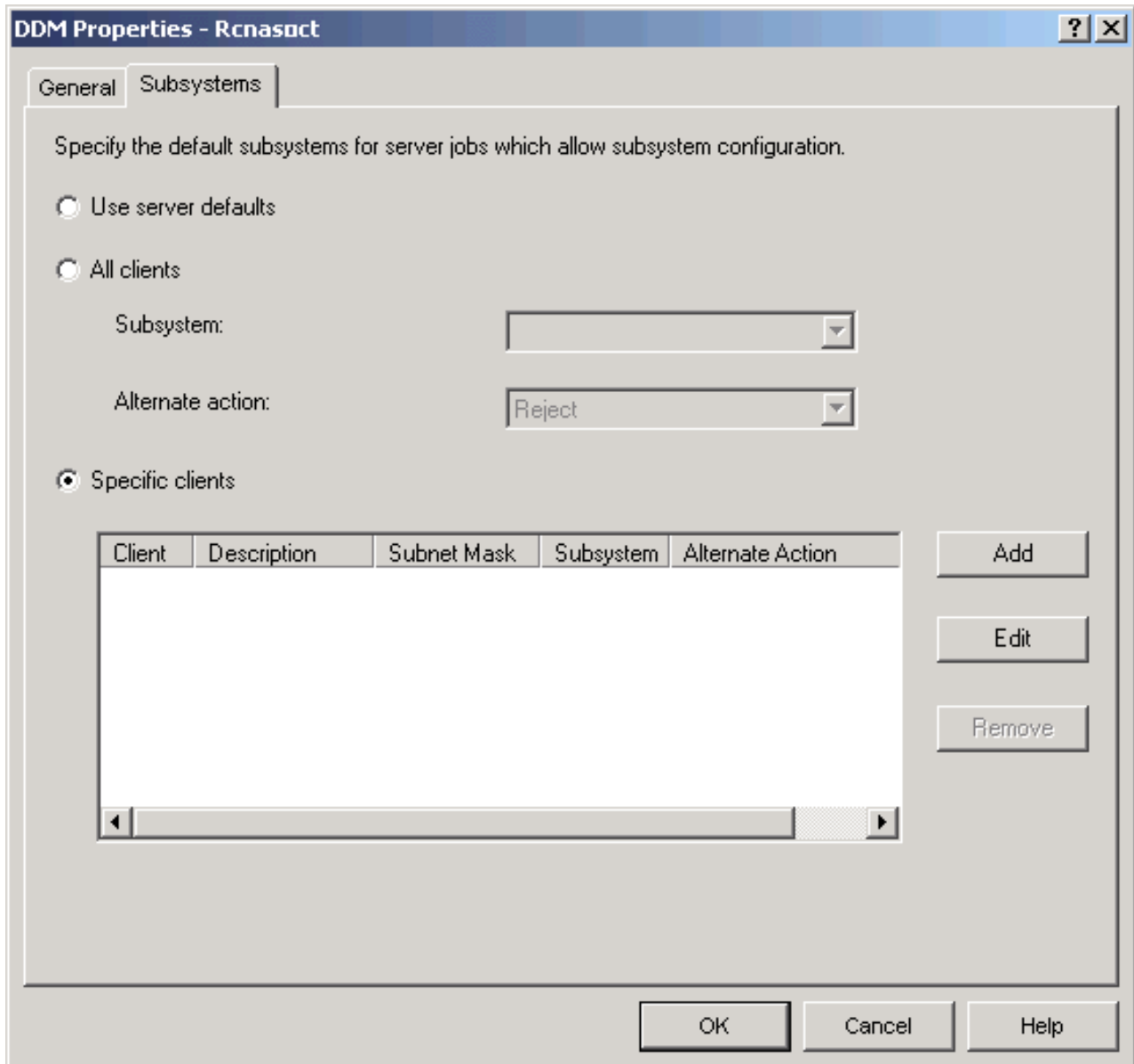
DDM 서버 작업 서브시스템 구성

V5R2 이후부터 DDM TCP/IP 서버 작업은 디폴트로 QUSRWRK 서브시스템에서 실행됩니다. iSeries Navigator를 사용하여 DDM 서버 작업을 모든 또는 특정 서버 작업을 클라이언트의 IP 주소를 기반으로 한 대체 서브시스템에서 실행하도록 구성할 수 있습니다. 구성을 설정하려면 다음을 수행하십시오.

1. 요구되는 서브시스템 각각에 대한 사전시작 작업 항목을 사전시작 작업 항목 추가(ADDPJE) 명령으로 작성하십시오. 사전시작 작업 속성에 대한 자세한 정보는 117 페이지의 『DRDA/DDM 사전시작 작업』을 참조하십시오.
2. 사전시작 작업 시작(STRPJ) 명령으로 작성한 사전시작 작업 항목을 시작하십시오.
3. iSeries Navigator에서 **Network**를 펼치십시오.
4. **Servers**를 펼치십시오.
5. **TCP/IP**를 클릭하십시오.
6. 오른쪽 패널에 표시되는 서버 리스트에서 **DDM**을 마우스 오른쪽 버튼으로 클릭하고 **Properties**를 선택합니다.

7. **Subsystems** 탭에서 특정 클라이언트와 서브시스템명을 추가합니다.

아래 예에서, 네트워크의 다른 서버가 QUSRWRK에 연결하고 실행하는 반면 관리자는 QADMIN 서브시스템에 연결하고 실행합니다. 다른 모든 클라이언트는 거부됩니다.



서버 작업 식별

서버에서 시작된 서버 작업을 살펴 보면 서버 작업이 어떤 정해진 어플리케이션 리퀘스터 작업이나 특정의 PC 클라이언트와 관련되어 있다는 사실을 쉽게 발견하지 못할 수도 있습니다. 특정의 작업을 식별할 수 있는 능력은 문제점을 조사하고 실행 자료를 모으는 일에 대한 요구사항입니다. iSeries Navigator는 작업을 훨씬 쉽게 할 수 있도록 이 타스크에 대하여 지원합니다.

이 섹션에서는 사용자가 iSeries Navigator를 사용하지 않고 있을 때 디버그나 실행 조사를 시작하기 전에 서버 작업을 식별하는 방법에 대한 정보를 제공합니다.

iSeries 작업명

iSeries에서 사용된 작업명은 다음의 세 부분으로 구성됩니다.

- 간단한 작업명
- 사용자 ID
- 작업 번호(오름차순)

DDM 서버 작업은 다음과 같은 규약에 따릅니다.

- 작업명은 QRWTSRVR입니다.
- 사용자 ID
 - 사전시작 작업이 사용되거나 사용되지 않거나 언제나 QUSER입니다.
 - 작업 기록부는 현재 작업을 사용하고 있는 사용자에게 대해 나타내고 있습니다.
- 작업 번호는 작업 관리에 의해 작성됩니다.

서버 작업 표시

서버 작업을 식별하는 데 보조로 사용될 수 있는 세 가지 메소드가 있습니다. 첫 번째는 활동 작업(Job)에 대한 작업(WRKACTJOB) 명령을 사용하는 것입니다. 두 번째는 사용자 작업(Job)에 대한 작업(WRKUSRJOB) 명령을 사용하는 방법입니다. 세 번째는 어느 클라이언트 사용자에게 의해 어느 작업이 사용되고 있는지를 판별하는 이력 기록부를 표시하는 방법입니다.

WRKACTJOB를 사용하여 활동 중인 작업 표시: 활동 작업(Job)에 대한 작업(WRKACTJOB) 명령은 모든 활동 작업을 표시합니다. 서버 작업 전체가 표시될 뿐만 아니라 리스너 작업도 표시됩니다.

다음의 그림은 (WRKACTJOB) 명령을 사용하여 샘플 상태를 나타내고 있습니다. 서버에 관련된 작업만 그림에 나타납니다. 사용할 수 있는 사전시작 작업을 참조하려면 F14 키를 누르십시오.

다음의 작업 유형은 그림에서 볼 수 있습니다.

- **1** - 리스너 작업
- **2** - 사전시작 서버 작업

활동 중인 작업(Jobs)에 대한 작업 AS400597
 04/25/97 10:25:40
 CPU %: 3.1 경과 시간: 21:38:40 활동 중인 작업: 77

옵션을 입력한 후 Enter를 누르십시오.
 2=변경 3=보류 4=종료 5=작업 대상 6=해제 7=메세지 표시
 8=스플 파일에 대한 작업 13=단절 ...

선택	서브시스템/작업	사용자	유형	CPU %	기능	상태
—	QUSRWRK	QSYS	SBS	.0		DEQW
—	1 QRWTLSTN	QUSER	BCH	.0		SELW
—	2 QRWTSRVR	QUSER	PJ	.0		TIMW
—	QRWTSRVR	QUSER	PJ	.0		TIMW
—	QRWTSRVR	QUSER	PJ	.0		TIMW
—	QRWTSRVR	QUSER	PJ	.0		TIMW
—	QRWTSRVR	QUSER	PJ	.0		TIMW

계속...

다음과 같은 작업 유형도 나와 있습니다.

PJ 사전시작 서버 작업

SBS 서브시스템은 작업을 모니터합니다.

BCH 리스너 작업

WRKUSRJOB를 사용하여 활동 중인 사용자 작업 표시: 사용자 작업(Job)에 대한 작업(WRKUSRJOB) 명령 USER(QUSER) STATUS(*ACTIVE)는 QUSER하에서 실행하는 활동의 서버 작업을 전부 표시합니다. 이는 DDM 리스너와 모든 DDM 서버 작업을 포함합니다. 이 명령이 바람직할 수도 있는데, 이것은 사용자가 DDM에 관련된 작업을 찾기 위해 조사해야 할 작업을 보다 적게 나열하기 때문입니다.

이력 기록부 표시

클라이언트 사용자가 서버 작업과의 연결을 성공적으로 설치할 때마다 해당 작업은 해당 클라이언트 사용자의 프로파일하에 실행하도록 교체됩니다. 어떤 작업이 특정의 클라이언트 사용자와 연관되어 있는지 판별하기 위해 사용자는 기록부 표시(DSPLOG) 명령을 사용하여 이력 기록부를 표시할 수 있습니다. 제공되는 정보의 예는 다음의 그림을 참조하십시오.

이력 기록부 내용 표시

```
.  
DDM 작업 036995/QUSER/QRWTSRVR 서비스 사용자 MEL 날짜 08/18/97 시간 15:26:43.  
.  
DDM 작업 036995/QUSER/QRWTSRVR 서비스 사용자 REBECCA 날짜 08/18/97 시간 15:45:08.  
.  
DDM 작업 036995/QUSER/QRWTSRVR 서비스 사용자 NANCY 날짜 08/18/97 시간 15:56:21.  
.  
DDM 작업 036995/QUSER/QRWTSRVR 서비스 사용자 ROD 날짜 08/18/97 시간 16:02:59.  
.  
DDM 작업 036995/QUSER/QRWTSRVR 서비스 사용자 SMITH 날짜 08/18/97 시간 16:48:13.  
.  
DDM 작업 036995/QUSER/QRWTSRVR 서비스 사용자 DAVID 날짜 08/18/97 시간 17:10:27.  
.  
.  
.
```

계속하려면 Enter를 누르십시오.

F3=종료 F10=전체 화면 F12=취소

주: 다음은 MSGID 매개변수와 기록부 표시(DSPLOG) 명령을 사용하여 불필요한 항목을 필터링하는 방법의 예입니다.

```
DSPLOG MSGID(CPI3E34)
```

QRWOPTIONS 자료 영역에 적절한 옵션을 설정하여 이 레코드가 이력 기록부에 쓰여지는 것을 방지할 수도 있습니다. 자세한 정보는 QRWOPTIONS 자료 영역 사용법 주제를 참조하십시오.

관계형 데이터베이스 디렉토리 감사

관계형 데이터베이스 디렉토리에 대한 액세스는 다음의 조건이 충족될 때 보안 감사 저널에 기록됩니다.

- 시스템 QAUDLVL의 값은 *SYSMGT입니다.
- 사용자 AUDLVL의 값은 *SYSMGT입니다.

*SYSMGT 값을 사용하여 서버는 다음의 명령으로 실행된 액세스 전체에 대해 감사를 실시합니다.

- 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령
- 관계형 데이터베이스 디렉토리 항목 변경(CHGRDBDIRE) 명령
- 관계형 데이터베이스 디렉토리 항목 표시(DSPRDBDIRE) 명령
- 관계형 데이터베이스 디렉토리 항목 제거(RMVRDBDIRE) 명령
- 관계형 데이터베이스 디렉토리 항목에 대한 작업(WRKRDBDIRE) 명령

관계형 데이터베이스 디렉토리는 디렉토리 항목 명령을 사용하지 않고 직접 읽을 수 있는 데이터베이스 파일(QSYS/QADBXRDBD)입니다.

V5R2 이전에는 라이브러리 QSYS의 관계형 데이터베이스(RDB) 디렉토리 파일 QADBXRDBD가 *PUBLIC에 부여된 조작 권한으로 빌드되었습니다. 그러나 V5R2부터는 더 이상 해당되지 않습니다. 따라서 이 파일을 사용하여 RDB 디렉토리에 액세스하는 기존의 프로그램은 정확하게 실행되지 않을 수도 있습니다. *ALLOBJ

특수 권한이 없다면, QADBXRDBD에 빌드되는 QADBXRMTNM이라는 이름의 논리 파일에 액세스해야 합니다. 이 파일에 대한 직접 액세스를 감사하려면 오브젝트 감사 변경(CHGOBJAUD) 명령으로 감사를 진행하십시오.

제 7 장 분산 관계형 데이터베이스에 대한 자료 가용성 및 보호

분산 관계형 데이터베이스(RDB) 환경에서 자료 가용성은 네트워크 안에 있는 각 서버의 자료를 보호할 뿐만 아니라 사용자가 해당 네트워크 전역에 있는 자료에 액세스하도록 보장하는 일도 포함합니다.

iSeries 서버는 분산 관계형 데이터베이스 네트워크의 서버에 있는 자료를 사용할 수 있도록 다음과 같은 배열 기능을 제공합니다

- 저장/복구
- 저널 관리 및 액세스 경로 저널링
- 확약 제어
- 보조 기억장치 풀
- 체크섬 보호
- 이중복사 보호 및 무정전 전원 장치

각 서버의 시스템 오퍼레이터가 일반적으로 그 서버 자료의 백업과 회복을 담당하고 있으므로 분산 관계형 데이터베이스를 위한 회복 지원을 참조하십시오. 네트워크 전반의 자료를 최적화하여 사용할 수 있도록 하는 전략을 세울 때 자료 중복성뿐만 아니라 네트워크 중복성이라는 측면도 고려해야 합니다. 특정의 자료가 사용자의 기업망에 대해 중요한 만큼 해당 자료에 액세스하기 위해서는 더 많은 방법을 갖고 있어야 합니다.

분산 관계형 데이터베이스에 대한 회복 지원

컴퓨터 서버에서 발생할 수 있는 장애는 서버 장애(전체 서버가 작동하고 있지 않을 때), 화재, 홍수나 유사한 비상 재해로 인한 사이트 손실 또는 오브젝트의 손상이나 손실 등을 포함합니다. 분산 관계형 데이터베이스(RDB)에 대하여, 네트워크에 있는 하나의 서버에서 발생한 장애는 네트워크 전역의 사용자가 해당 서버의 관계형 데이터베이스에 액세스하지 못하게 만듭니다. 관계형 데이터베이스가 다른 위치에서 매일 매일의 사무 비즈니스 활동에 중요한 역할을 한다면, 하나의 서버가 회복되는 동안 네트워크 전체에 걸친 전사적 조작이 중단될 수 있습니다. 확실히, 장애 발생 후에 자료 보호와 회복에 대한 계획은 분산 관계형 데이터베이스(RDB)에서 특히 중요합니다.

분산 관계형 데이터베이스(RDB)에서 각 서버는 시스템 자체에서 소유하는 자료를 백업하고 회복할 책임을 갖고 있습니다. 또한 네트워크 안에 각 서버는 이상 서버의 종료 후에 회복 프로시디어를 처리합니다. 그러나, 백업과 회복 프로시디어는 조작이 미숙한 조작자가 다루고 있거나 조작자가 전혀 없는 서버에 대해 표시장치 pass-through를 사용하여 분산 관계형 데이터베이스(RDB) 관리자에 의해 실행될 수 있습니다.

가장 일반적인 손실 유형은 오브젝트나 오브젝트 그룹의 손실입니다. 오브젝트는 전원 장애, 하드웨어 장애, 시스템 프로그램 오류, 어플리케이션 프로그램 오류 또는 조작자 오류 등을 포함하여 여러 요소로 인해 손실되거나 손상될 수 있습니다. iSeries 서버는 서버 프로그램, 어플리케이션 프로그램 및 자료가 영원히 없어지지 않도록 예방하기 위한 여러 가지의 메소드를 제공합니다. 장애 유형과 선택된 보호 레벨에 따라서 대부분의 프로그램과 자료가 보호될 수 있고, 회복 시간이 크게 줄어들 수 있습니다.

이 보호 메소드는 다음과 같습니다.

- 오브젝트가 저장되어 있는 곳을 제어하는 보조 기억장치 풀, 보조 기억장치 풀에 대한 체크섬 보호 및 디스크에 관련된 하드웨어 구성요소에 대한 이중복사 보호와 같이 분산 관계형 데이터베이스에 대한 디스크 실패 후 자료 회복
- 관계형 데이터베이스 변경에 관한 보조 레코드와 자료에 대한 저널링 색인에 대한 분산 관계형 데이터베이스에 대한 저널 관리
- 관계형 데이터베이스 트랜잭션이 일정하게 적용되거나 제거될 수 있도록 보장하는 확약 제어를 통한 트랜잭션 회복
- 표, 컬렉션, 패키지 및 관계형 데이터베이스 디렉토리과 같이 SQL 오브젝트가 저장되고 복원될 수 있도록 보장하는 분산 관계형 데이터베이스에 대한 저장 및 복원 처리
- 보조 기억장치에 자료 기록

파일 작성 명령의 FRCRATIO(강제-쓰기 비율) 매개변수는 자료가 보조 기억장치에 기록되도록 강제하는 데 사용될 수 있습니다. 강제-쓰기 비율은 모든 추가, 갱신 및 삭제 요구가 문제의 표에 대하여 즉시 보조 기억장치에 기록되게 합니다. 그러나, 이 옵션을 선택하면 서버 성능을 감소시킬 수 있습니다. 따라서, 사용자의 표와 저널링 표를 저장하는 데는 데이터베이스를 보호하기 위한 1차 메소드가 고려되어야 합니다.

- **실제 보호**

사용자의 시스템이 갑작스런 전원 손실로부터 보호되는지 확인하는 것은 사용자의 어플리케이션 서버(AS)가 어플리케이션 리퀘스터(AR)에 제공되는 것임을 보장하는 중요한 부분입니다. 무정전 전원 장치(uninterruptible power supply)는 별도로 주문될 수 있는 것으로, 전원이 복구될 때까지 서버 장치에 전원을 공급함으로써 전원 장애, 전원 중단 또는 전압 제거로 인한 손실로부터 서버를 보호합니다. 일반적으로, 무정전 전원 장치는 작업 장치 전체에 전원을 공급하지는 못합니다. iSeries 서버를 사용하여 무정전 전원 장치는 서버가 다음을 수행할 수 있게 합니다.

- 짧은 전원 중단 시간이나 순간적인 전압 제거 시간 동안 조작을 계속할 수 있게 합니다.
- 파일을 닫고 오브젝트 무결성을 유지보수하여 조작을 정상적으로 종료할 수 있게 합니다.

분산 관계형 데이터베이스에 대한 디스크 장애 발생 후 자료 회복

디스크 장애가 발생하고 장애 발생 전에 전체 오브젝트가 즉시 테이프나 디스크에 저장되지 않으면 최근에 입력된 자료에 대한 회복은 가능하지 않습니다. 이전에 저장된 오브젝트가 복원된 후에 서버는 조작이 가능하지만 데이터베이스는 행해지지 않습니다.

ASP(보조 기억장치 풀), 체크섬 보호 및 이중복사 보호는 디스크에 관련된 장애가 발생한 후 최근에 입력된 자료를 회복하는 메소드를 제공하는 OS/400 디스크 회복 기능입니다. 이 기능은 추가 서버 자원을 사용하지 않고 분산 관계형 데이터베이스(RDB) 안에 있는 서버에 대해서 상급 레벨의 보호를 제공합니다. 일부 서버는 다른 서버보다 어플리케이션 서버로서 더 중요할 수도 있기 때문에 분산 관계형 데이터베이스(RDB) 관리자는 이 디스크 자료 보호 메소드가 네트워크 내에서 각 서버에 의해 가장 잘 사용될 수 있는 방법인지를 검토해야 합니다.

보조 기억장치 풀(ASP), 체크섬 보호 및 이중복사 보호에 대한 자세한 정보는 iSeries Information Center의 백업 및 회복 주제를 참조하십시오.

보조 기억장치 풀

ASP는 같은 저장 영역에 할당된 하나 또는 그 이상의 실제 디스크 장치입니다. ASP는 사용자가 지정된 실제 디스크 장치에 어떤 정해진 유형의 오브젝트를 분리시킬 수 있게 합니다.

서버 ASP는 서버 프로그램에 의한 처리의 결과로써 작성되는 서버 프로그램과 임시 오브젝트를 분리시킵니다. 사용자 ASP는 라이브러리, SQL 오브젝트, 저널, 저널 리시버, 어플리케이션 및 자료와 같이 일부 오브젝트를 분리시키는 데 사용될 수 있습니다. iSeries 서버는 최대 32개의 기본 사용자 ASP와 223 독립형 사용자 ASP를 지원합니다. 사용자 ASP에서 라이브러리나 오브젝트를 분리시키면 다른 ASP에서 발생하는 디스크 장애로부터 이것을 보호하게 되고 회복 시간도 줄어듭니다.

회복 시간의 경감과 오브젝트의 분리에 더하여, ASP에 오브젝트를 위치지정하면 성능을 향상시킬 수 있습니다. 저널 리시버가 사용자 ASP에서 분리되면 해당 ASP와 연관된 디스크는 해당 리시버의 전용이 됩니다. 데이터베이스 파일에 대해 많은 읽기 및 쓰기 조작을 요구하는 환경에서, 이것은 해당 ASP 내의 디스크에서 암 (arm) 경합을 줄일 수 있고 저널링 성능을 향상시킬 수 있습니다.

분산 관계형 데이터베이스에서의 체크섬 보호

체크섬 보호는 ASP 안에 있는 디스크의 자료가 분실되지 않도록 하는 보호장치입니다. 체크섬 소프트웨어는 해당 ASP 내의 특수한 체크섬 자료 영역에 들어 있는 ASP 자료의 암호화된 사본을 유지보수합니다. 체크섬 보호 ASP에서 영구적인 오브젝트에 대한 변경사항은 체크섬 세트의 체크섬 자료 안에 자동으로 유지보수됩니다. 체크섬 세트에 단일 디스크 장치가 없다면 서버는 해당 세트에서 남아 있는 기능 단위의 체크섬과 자료를 사용하여 분실된 장치의 내용을 재구성합니다. 이 방법으로 어느 장치에서든 장애가 발생하면 해당 장치의 내용이 회복될 수도 있습니다. 재구성된 자료는 장애 발생 시 디스크에 있었던 가장 최근의 정보를 반영합니다. 체크섬 보호는 서버 성능에 크게 영향을 줄 수 있습니다. 분산 관계형 데이터베이스(RDB)에서 이것은 중요한 부분일 수도 있습니다.

분산 관계형 데이터베이스에 대한 이중복사 보호

이중복사 보호는 디스크 제어기, 디스크 I/O 프로세서 또는 버스와 같이 서로 다른 디스크에 관련된 하드웨어 구성요소를 복사함으로써 서버의 가용성을 높여 줍니다. 서버는 장애 발생 후에 사용 가능한 상태를 유지할 수 있으며 장애가 생긴 하드웨어 구성요소에 대한 서비스를 편리한 시간에 제공받을 수 있습니다.

이중복사 보호 레벨은 서버 가용성의 레벨에 따라 차이가 있습니다. 예를 들어, 하나의 서버에 있는 디스크 장치만 이중복사되면 전체 디스크 장치가 디스크 장치의 레벨 보호를 받게되므로, 해당 서버는 단일 디스크 장치의 장애 발생에 대하여 보호됩니다. 이 상황에서 제어기, I/O 프로세서 또는 버스 장애가 발생되면 서버는 장애가 발생한 부품을 수리하거나 교체할 때까지 실행할 수 없습니다. 서버에서 이중복사된 장치 전체는 같은 디스크 장치의 보호 레벨을 갖고 있고 동일한 ASP에 있어야 합니다. ASP에 장치는 이중복사 보호가 시작될 때 서버에 의해 자동으로 한 쌍씩 짝지어집니다.

분산 관계형 데이터베이스에 대한 저널 관리

저널 관리는 관계형 데이터베이스와 색인에 대해 백업 및 회복 전략의 일부로서 사용될 수 있습니다.

저널링에 대한 자세한 정보는 iSeries Information Center의 저널 관리 주제를 참조하십시오.

iSeries 저널 지원은 감사 추적과 정방향과 역방향의 회복을 제공합니다. 정방향 회복은 더 오래된 버전의 표를 받아들이고 저널에 기록된 변경사항을 해당 표에 적용하는 데 사용될 수 있습니다. 역방향 회복은 표에서 저널에 기록된 변경사항을 제거하는 데 사용될 수 있습니다.

컬렉션이 작성되었을 때 저널 리시버라고 일컫는 저널과 오브젝트는 컬렉션에서 작성됩니다. 저널 리시버가 표와 다른 ASP에 있을 때 성능이 개선됩니다. 사용자 ASP에 컬렉션을 두는 것은 표와 저널 그리고 저널 리시버 모두를 같은 사용자 ASP에 두는 것입니다. 성능 상으로는 아무 이득이 없습니다. 다른 ASP(이 저널의 저널 리시버를 위해서만 사용되는)에 새로운 저널 리시버를 작성하여 저널 변경(CHGJRN) 명령으로 이에 접속하면 다음 서버는 모든 저널 리시버가 다른 사용자 ASP에 만들어지도록 생성되며 사용자는 성능이 개선된 것을 알 수 있습니다.

표가 작성되었을 때 이것은 컬렉션 안에 작성된 저널 SQL에 자동으로 저널됩니다. 이 지점부터, 사용자는 저널, 저널 리시버 및 저널에 대한 저널링 표를 관리하기 위해 저널 기능을 사용해야 할 책임이 있습니다. 예를 들어, 표가 컬렉션 안으로 이동되면 저널링 상태로 어떠한 자동 변경도 일어나지 않습니다. 표가 복원되면 일반적인 저널 규칙이 적용됩니다. 다시 말해, 표가 저장될 때 저널되면 이 표는 해당 서버에서 복원될 때 같은 저널로 저널됩니다. 표가 저장 시에 저널되지 않으면 이 표는 복원 시에 저널되지 않습니다. 저널 기능을 사용하여 표에 대한 저널링을 중단할 수 있으나, 그렇게 하면 SQL 조작이 확약 제어하에 실행하지 못하게 방해합니다. SQL 조작은 COMMIT(*NONE)을 지정했으면 계속 실행될 수 있으나, 이것은 저널링과 확약 제어가 제공하는 동일한 레벨의 무결성을 제공하지는 않습니다.

저널링 활동을 사용하여, 데이터베이스에 대한 변경이 생길 때 해당 변경사항은 데이터베이스가 변경되기 전에 저널 리시버에서 저널됩니다. 저널 리시버는 언제나 최근의 데이터베이스 정보를 갖고 있습니다. 모든 활동은 변경된 방법과 관계없이 데이터베이스 표에 대해서 저널됩니다.

저널 리시버는 특정 행(추가, 변경 또는 삭제된) 및 표(열린, 저장된 표나 멤버 등)대한 레코드 활동을 입력합니다. 각 항목에는 활동의 소스, 사용자, 작업, 프로그램, 시간 및 날짜를 식별하는 추가 제어 정보가 들어 있습니다.

서버는 표의 이동 및 표의 이름 변경을 포함하여 일부 파일의 레벨 변경을 저널합니다. 또한 서버는 실제 파일 멤버의 초기설정과 같이 멤버 레벨의 변경과 IPL(초기의 프로그램 로드)과 같이 서버 레벨의 변경을 저널합니다. 사용자는 저널 리시버에 항목을 추가하여(작업 단계가 나중에 재시작될 수 있도록 작업 상태와 서버에 대한 정보가 저널될 수 있는 검사 항목과 같이) 중요한 이벤트를 식별할 수 있습니다.

단일 행에 영향을 주는 변경에 대하여, 제어정보 다음으로 행 이미지가 포함되어 있습니다. 변경된 후에는 언제나 행의 이미지가 포함됩니다. 선택에 따라, 변경되기 전에 행 이미지도 포함될 수 있습니다. 사용자는 저널 실제 파일 시작(STRJRNPF) 명령에 IMAGES 매개변수를 지정하여 행 이미지의 전후로 저널 또는 행 이미지의 이후로 저널 여부에 대해 제어합니다.

저널된 전체 데이터베이스 파일은 서버가 시작될 때(IPL 시간) 또는 독립형 ASP의 연결변환 중에 자동으로 해당 저널과 동기화됩니다. 서버가 이상적으로 종료했거나 독립 ASP가 이상적으로 단절변환했으면 저널에 어떤 데이터베이스 변경이 있을 수도 있으나 아직 데이터베이스 자체에는 반영되지 않습니다. 데이터베이스에도 변경이 생기면 서버는 자동으로 저널에서 데이터베이스를 갱신하고 지금까지의 표를 회생시킵니다.

저널링은 데이터베이스 표가 보다 쉽고 빠르게 저장될 수 있게 합니다. 예를 들어, 사용자는 전체의 표를 매일 저장하는 대신에 표에 대한 변경사항을 포함하는 저널 리시버를 간단하게 저장할 수 있습니다. 사용자는 늘 전체의 표를 정기적으로 저장할 수도 있습니다. 이 메소드는 일상의 저장 작업을 실행하는 데 걸리는 시간을 줄일 수 있습니다.

저널 표시(DSPJRN) 명령은 저널 리시버 입력을 데이터베이스 파일로 변환하는 데 사용될 수 있습니다. 이와 같은 파일은 활동 보고서, 감사 추적, 보안 및 프로그램 디버깅에 사용될 수 있습니다.

색인 회복

색인은 표에서 행을 읽는 순서에 대해 설명합니다. 색인이 저널에 기록되어 있으면 서버는 비정상적인 서버 종료 다음의 IPL 동안 또는 비정상적인 단절변환 후 독립형 ASP의 연결변환 중에 색인을 리빌드하는데 많은 시간을 허비하지 않고 색인을 회복시킬 수 있습니다.

표를 저널할 때 표 안에 행에 대한 변경사항의 이미지가 저널에 기록됩니다. 이 행 이미지는 서버가 이상적으로 종료하는 경우에 표를 회복하는 데 사용됩니다. 그러나, 이상 종료 다음에 해당 서버는 표를 통해 빌드된 색인이 표 안에 자료와 동기화되지 않는다는 사실을 발견할 수도 있습니다. 액세스 경로와 이 경로의 자료가 동기화되지 않으면 서버는 색인을 재설정하여 둘 다 동기화되어 있고 사용 가능하다는 것을 확인해야 합니다.

색인이 저널되었을 때 서버는 저널에 색인의 이미지를 기록하여 해당 색인과 이것의 자료 사이에 알려져 있는 동기화(SYN) 점을 부여합니다. 저널에 있는 해당 정보에 의해, 서버는 자료와 색인을 모두 회복할 수 있고, 또한 둘 다 동기화되어 있음을 확인할 수 있습니다. 이와 같은 경우에, 색인을 재설정하는 데 긴 시간을 피할 수 있습니다.

iSeries 서버는 색인 회복 작업을 보조하기 위해 여러 기능을 제공합니다. 서버의 전체 색인은 해당 색인이 유지보수되는 시간을 지정하는 유지보수 옵션을 갖고 있습니다. SQL 색인은 *INNED 유지보수에 관한 속성으로 작성됩니다.

전원 장애나 이상 서버 장애가 발생한 경우에 변경 처리 중에 있는 색인은 이것이 이 해당 자료와 일치하는지 확인하기 위해 재설정이 필요할 수도 있습니다. 서버의 전체 색인은 필요할 경우 해당 색인이 재설정되어야 할 시간을 지정하는 회복 옵션을 갖고 있습니다. UNIQUE의 속성을 가지는 전체 SQL 색인은 *IPL의 회복 속성으로 작성되며, 이것은 OS/400 라이선스가 부여된 프로그램이 시작되기 전에 이 색인이 재설정된다는 의미입니다. 다른 모든 SQL 색인은 *AFTIPL 회복 속성으로 작성되며, 이것은 이 색인이 오퍼레이팅 시스템이 시작된 후 또는 독립 ASP가 연결변환된 후에 재설정된다는 의미입니다. IPL 시간 또는 독립 ASP 연결변환 동안 사용자는 재설정되는 데 필요한 색인과 이것의 회복 옵션을 나타내는 화면을 참조할 수 있고, 또한 이 회복 옵션을 대체할 수도 있습니다.

SQL 색인은 자동으로 저널되지 않습니다. 사용자는 저널 액세스 경로 시작(STRJRNAP) 명령을 사용하여 SQL 조작에 의해 작성된 색인을 저널할 수 있습니다. 서버 저장과 복원 기능은 사용자가 오브젝트 저장(SAVOBJ)

또는 라이브러리 저장(SAVLIB) 명령의 ACCPTH(*YES)를 사용함으로써 표가 저장될 때 색인을 저장할 수 있게 합니다. 표를 복원해야 하는 경우, 색인을 재설정할 필요가 없습니다. 이전에 저장되고 복원된 색인은 데이터베이스에 의해 자동으로 비동기식으로 재설정됩니다.

색인을 저널링하기 전에 해당 색인과 연관된 표에 대한 저널링을 시작해야 합니다. 이에 더하여, 색인과 이것에 연관된 표에 대해 같은 저널을 사용해야 합니다.

색인 저널링은 추가의 출력 작업을 최소화하도록 설계됩니다. 예를 들어, 서버는 같은 출력 작업에서 변경된 행과 변경된 색인에 대한 저널 자료를 기록합니다. 그러나, 사용자는 색인을 저널링하기 시작할 때 사용자 ASP에서 저널 리시버를 분리시키는 것을 신중하게 고려해야 합니다. 저널 리시버를 이것의 고유 사용자 ASP 안에 위치지정하면 최고의 저널 관리 성능이 부여되고 그리고 동시에 디스크 장애 발생으로부터 저널 리시버를 보호하는 데 도움이 됩니다.

색인 재설정 시간을 줄이는 표 설계

표 설계는 색인 회복 시간을 줄여 줍니다. 예를 들어, 사용자는 큰 마스터 표를 이력 표와 트랜잭션 표로 나눌 수 있습니다. 이 때, 트랜잭션 표는 새로운 자료를 추가하는 데 사용되며 이력 표는 조회용으로만 사용됩니다. 매일 트랜잭션 자료를 이력 표로 병합하고, 다음 날의 자료를 담기 위해 트랜잭션 파일을 비워둘 수 있습니다. 이 설계를 사용하면 서버가 계속 이상적으로 종료되는 경우에 보다 작은 트랜잭션 표로의 색인을 재설정해야 할 수도 있기 때문에 색인을 재설정하는 시간을 줄일 수 있습니다. 그러나, 큰 이력 표로의 색인은 그날의 대부분의 시간 동안 읽기만 가능하므로 이 색인은 이것의 자료와 동기화 가능성이 있을 수도 있고, 반드시 재설정될 필요도 없습니다.

색인 재설정 시간을 줄이는 데 표 설계를 사용하는 방법과 액세스 경로 저널링과 같이 서버가 공급하는 기능을 사용하는 방법 사이에 균형을 고려하십시오. 위에서 설명한 표 설계는 보다 더 복잡한 어플리케이션 설계를 요구할 수도 있습니다. 실제의 상황을 평가한 다음에, 사용자는 더 복잡한 어플리케이션을 설계하기 보다는 오히려 액세스 경로 저널링과 같이 서버가 공급하는 기능을 사용하기로 결정할 수도 있습니다.

SMAPP(시스템 관리식 액세스 경로 보호)

SMAPP(시스템 관리식 액세스 경로 보호)는 액세스 경로에 대해 자동 보호를 제공합니다. SMAPP 지원을 사용하면 사용자는 저널 액세스 경로 시작(STRJRNAP) 명령과 같이 저널링 명령을 사용할 필요없이 액세스 경로 저널링의 장점을 얻을 수 있습니다. SMAPP 지원은 IPL 시간 또는 독립 ASP의 연결변환 동안 액세스 경로를 재설정하기 보다는 이상 시스템 종료 후에 이 경로를 회복합니다.

SMAPP 지원은 출하된 시스템으로 전원이 켜집니다.

이 서버는 사용자에 의해 또는 서버가 제공하는 디폴트 시간을 사용함으로써 부여되는 목표 액세스 경로의 회복에 기초하여 보호해야 할 액세스 경로를 판별합니다. 목표 액세스 경로 회복 시간은 전조직에 미치는 값으로서 또는 ASP 기준에서 지정될 수 있습니다. 사용자 정의의 저널로 저널되어 있는 액세스 경로는 이미 보호되고 있으므로 SMAPP 보호에 대해 바람직하지 않습니다. SMAPP에 대한 자세한 정보는 iSeries Information Center의 시스템이 관리하는 액세스 경로 보호(SMAPP) 주제를 참조하십시오.

확약 제어를 통한 트랜잭션 회복

확약 제어는 iSeries 서버에서 저널 관리 기능의 연장입니다. 서버는 단일 작업 단위(트랜잭션)로서 관계형 데이터베이스의 변경 그룹을 식별하고 처리할 수 있습니다.

SQL COMMIT 명령문은 조작 그룹이 완료되었음을 보증합니다. SQL ROLLBACK 명령문은 조작 그룹이 취소되었음을 보증합니다. 확약되거나 롤백될 수 없는 SQL 명령문은 다음과 같습니다.

- DROP COLLECTION
- 지정된 오브젝트에 대해 권한 보유자가 존재하는 경우, GRANT 또는 REVOKE

확약 제어하에서, 트랜잭션 도중에 사용된 표와 행은 다른 작업에 대해 잠겨 있습니다. 이것은 해당 트랜잭션이 완료될 때까지 다른 작업에서 해당 자료를 사용하지 못하도록 보장합니다. 트랜잭션이 끝났을 때 프로그램은 SQL COMMIT 또는 ROLLBACK 명령문을 발행하여 해당 행을 자유롭게 합니다. 확약 조치가 실행되기 전에 서버나 작업이 이상적으로 끝나면, 확약이나 롤백 조치가 일어났던 최종 시간 이후에 해당 작업에 대한 모든 변경이 롤백됩니다. 이 때, 아직까지 영향을 받아 잠겨있는 행이 모두 풀립니다. 잠금 레벨은 다음과 같습니다.

*NONE

확약 제어를 사용하지 않습니다. 다른 작업에서 확약되지 않은 변경은 참조할 수 있습니다.

*CHG

SQL ALTER, COMMENT ON, CREATE, DROP, GRANT, LABEL ON 및 REVOKE 명령문에서 언급된 오브젝트와 갱신, 삭제 및 삽입된 행은 작업 단위(트랜잭션)가 완료될 때까지 잠겨 있습니다. 다른 작업에서 확약되지 않은 변경은 참조할 수 있습니다.

*CS SQL ALTER, COMMENT ON, CREATE, DROP, GRANT, LABEL ON 및 REVOKE 명령문에서 언급된 오브젝트와 갱신, 삭제 및 삽입된 행은 작업 단위(트랜잭션)가 완료될 때까지 잠겨 있습니다. 선택되어 있으나 갱신되지 않은 행은 그 다음 행이 선택될 때까지 잠겨 있습니다. 다른 작업에서 확약되지 않은 변경은 참조할 수 없습니다.

*ALL SQL ALTER, COMMENT ON, CREATE, DROP, GRANT, LABEL ON 및 REVOKE 명령문에 언급된 오브젝트와 읽기, 갱신, 삭제 및 삽입이 실행된 행은 작업 단위(트랜잭션)가 끝날 때까지 잠겨 있습니다. 다른 작업에서 확약되지 않은 변경은 참조할 수 없습니다.

134 페이지의 표 6은 각각의 잠금 레벨 값에 대해 레코드 잠금 지속 시간을 나타내고 있습니다.

프로그램이 사전검파일되어 있을 때 또는 대화식 SQL이 시작되었을 때 COMMIT(*CHG), COMMIT(*CS) 또는 COMMIT(*ALL)를 요구하면, SQL은 확약 제어 시작(STRCMTCTL) 명령을 암시적으로 호출함으로써 확약 제어 환경을 설정합니다. SQL이 확약 제어를 시작할 때 지정된 LCKLVL 매개변수는 CRTSQLxxx 명령의 COMMIT 매개변수에 지정된 잠금 레벨입니다. NFYOBJ(*NONE)은 SQL이 확약 제어를 시작할 때 지정됩니다. 다른 NFYOBJ 매개변수를 지정하려면 SQL을 시작하기 전에 (STRCMTCTL) 명령을 발행하십시오.

주: 확약 제어로 실행할 때 자료 조작어 명령문에 의해 어플리케이션 프로그램 안에 언급된 표는 저널되어야 합니다. 이 표는 사전검파일 시간에 저널될 필요는 없으나 어플리케이션을 실행할 때 저널되어야 합니다.

리모트 관계형 데이터베이스가 서버에서 자료에 액세스하고 있고 약속 레벨 반복 읽기(*RR)를 요구하고 있으면, 조회가 닫힐 때까지 표가 잠기게 됩니다. 커서가 읽기 전용이면 표가 잠기게 됩니다(*SHRNUP). 커서가 갱신 모드에 있으면 표가 잠기게 됩니다(*EXCLRD).

SQL 콜렉션에서 작성된 저널은 일반적으로 모든 변경사항을 SQL 표에 기록하는 데 사용된 저널입니다. 그러나, 사용자는 서버 저널 기능을 사용하여 SQL 표를 다른 저널에 저널할 수 있습니다.

약속 제어는 작업 단위에서 131 072 고유 행 변경까지 처리할 수 있습니다. COMMIT(*ALL)이 지정되면 전체 읽은 행도 131 072 한계 내에 포함됩니다. 행이 변경되거나 하나의 작업 단위에서 한 번 이상 읽으면, 이 행은 131 072 한계에 대하여 한 번만 계산됩니다. 다수의 잠금을 유지보수하면 서버 성능에 악영향을 미치고 동시에 사용자가 해당 작업 단위가 끝날 때까지 작업 단위에서 잠겨 있는 행에 액세스할 수 없게 됩니다. 따라서, 작업 단위에서 처리되는 행의 수를 적게 유지하는 것이 더 효율적입니다. 약속 제어는 약속 제어하에 열리거나 작업 단위에서 지연 중인 변 경으로 닫혀 있는 표를 512개까지 허용합니다.

COMMIT과 ROLLBACK 명령문에서 HOLD 값은 커서를 열린 상태로 유지시키고 다시 OPEN을 발행하지 않고도 또다른 작업 단위를 시작할 수 있게 합니다. HOLD 값은 프로그램에 대해 릴리스되지 않은 비Series 연결이 있고 SQL이 아직까지 호출 스택에 있을 때 사용할 수 없습니다. ALWBLK(*ALLREAD) 및 COMMIT(*CHG) 또는 COMMIT(*CS)가 프로그램이 사전컴파일되어 있을 때 지정되면, 전체 읽기 전용 커서는 행의 블록 표시를 하고 ROLLBACK HOLD 명령문은 커서 위치를 역방향으로 진행시키지 못하게 됩니다.

SQL 사전컴파일된 프로그램이나 대화식 SQL 세션의 실행이 지연 중인 잠긴 행(레코드)이 있으면, 서버 명령 입력 화면으로부터 COMMIT나 ROLLBACK 명령문이 발행되지 않습니다. 그렇지 않으면, 작업이 끝났을 때 내재성 ROLLBACK 조치가 일어납니다.

사용자는 약속 정의에 대한 작업(WRKCMTDFN) 명령을 사용하여 약속 정의의 상태를 모니터하고 시스템을 통한 약속 제어 활동과 관련된 잠금 및 보류 자원을 자유롭게 합니다. 자세한 정보는 101 페이지의 『분산 관계형 데이터베이스에서 약속 정의에 대한 작업』을 참조하십시오.

약속 제어에 대한 자세한 정보는 iSeries Information Center의 트랜잭션 및 약속 제어 주제를 참조하십시오.

표 6. 레코드 잠금 지속

SQL 명령문	COMMIT 매개변수	레코드 잠금의 지속 시간	잠금 유형
SELECT INTO	*NONE *CHG *CS *ALL(주 2 참조)	잠금이 없음 잠금이 없음 읽고 해제 시 잠긴 행 읽기에서 ROLLBACK 또는 COMMIT까지	READ READ
FETCH(읽기 전용 커서)	*NONE *CHG *CS *ALL(주 2 참조)	잠금이 없음 잠금이 없음 읽기에서부터 그 다음 FETCH까지 읽기에서 ROLLBACK 또는 COMMIT까지	READ READ

표 6. 레코드 잠금 지속 (계속)

SQL 명령문	COMMIT 매개변수	레코드 잠금의 지속 시간	잠금 유형
FETCH(갱신이나 삭제를 할 수 있는 커서) 주 1 참조	*NONE *CHG *CS *ALL	레코드가 갱신되거나 삭제되지 않을 때 읽기에서부터 그 다음 FETCH까지 레코드가 갱신 또는 삭제될 때 읽기에서부터 UPDATE 또는 DELETE까지 레코드가 갱신되거나 삭제되지 않을 때 읽기에서부터 그 다음 FETCH까지 레코드가 갱신 또는 삭제될 때 읽기에서부터 UPDATE 또는 DELETE까지 레코드가 갱신되거나 삭제되지 않을 때 읽기에서부터 그 다음 FETCH까지 레코드가 갱신 또는 삭제될 때 읽기에서부터 UPDATE 또는 DELETE까지 읽기에서 ROLLBACK 또는 COMMIT까지	UPDATE UPDATE UPDATE UPDATE ³
INSERT(목표 표)	*NONE *CHG *CS *ALL	잠금이 없음 삽입에서부터 ROLLBACK 또는 COMMIT까지 삽입에서부터 ROLLBACK 또는 COMMIT까지 삽입에서부터 ROLLBACK 또는 COMMIT까지	UPDATE UPDATE UPDATE ⁴
INSERT(부속 선택된 표)	*NONE *CHG *CS *ALL	잠금이 없음 잠금이 없음 읽는 동안에 잠긴 각 레코드 읽기에서 ROLLBACK 또는 COMMIT까지	READ READ
UPDATE(비-커서)	*NONE *CHG *CS *ALL	갱신 중에 잠긴 각 레코드 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지	UPDATE UPDATE UPDATE UPDATE
DELETE(비-커서)	*NONE *CHG *CS *ALL	삭제 중에 잠긴 각 레코드 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지	UPDATE UPDATE UPDATE UPDATE
UPDATE(커서 사용)	*NONE *CHG *CS *ALL	레코드가 갱신될 때 해제된 잠금 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지	UPDATE UPDATE UPDATE UPDATE
DELETE(커서 사용)	*NONE *CHG *CS *ALL	레코드가 삭제될 때 해제된 잠금 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지 읽기에서 ROLLBACK 또는 COMMIT까지	UPDATE UPDATE UPDATE UPDATE
부속 조회(갱신, 삭제가 가능한 커서, UPDATE 또는 DELETE 비-커서)	*NONE *CHG *CS *ALL(주 2 참조)	읽기에서부터 그 다음 FETCH까지 읽기에서부터 그 다음 FETCH까지 읽기에서부터 그 다음 FETCH까지 읽기에서 ROLLBACK 또는 COMMIT까지	READ READ READ READ
부속 조회(읽기 전용 커서 또는 SELECT INTO)	*NONE *CHG *CS *ALL	잠금이 없음 잠금이 없음 읽는 동안에 잠긴 각 레코드 읽기에서 ROLLBACK 또는 COMMIT까지	READ READ

표 6. 레코드 잠금 지속 (계속)

SQL 명령문	COMMIT 매개변수	레코드 잠금의 지속 시간	잠금 유형
<p>주:</p> <ol style="list-style-type: none"> 결과표가 읽기 전용이 아닌 경우(iSeries Information Center의 SQL 참조서 주제의 DECLARE CURSOR 설명 참조)와 다음 중 하나에 해당하는 경우에는 커서가 UPDATE 또는 DELETE 기능을 사용하여 열립니다. <ul style="list-style-type: none"> 커서가 FOR UPDATE절을 포함하여 정의됩니다. 커서가 FOR UPDATE, FOR FETCH ONLY 또는 ORDER BY절을 포함하지 않고 정의되며 프로그램에 다음 중에서 적어도 하나가 포함되어 있습니다. <ul style="list-style-type: none"> 같은 커서명을 참조하는 UPDATE 커서 같은 커서명을 참조하는 DELETE 커서 CRTSQLxxx 명령에 지정된 ALWBLK(*READ) 또는 ALWBLK(*NONE)를 포함하는 EXECUTE 또는 EXECUTE IMMEDIATE 명령문 표나 뷰는 COMMIT(*ALL)을 만족시키기 위해서만 잠길 수 있습니다. 부속 선택이 처리되어 이것에 그룹이나 조합이 포함되는 경우 또는 조회의 처리로 임시의 결과에 대한 사용이 요구되는 경우, 예약되지 않은 변경을 참조하지 못하게 사용자를 보호하기 위해 잠금에 대한 독점적 사용이 지정됩니다. 행이 갱신되거나 삭제되지 않을 경우, 잠금은 *READ로 떨어집니다. 목표 표의 행에서 UPDATE 잠금과 부속 선택 표의 행에서 READ 잠금. 표나 뷰는 COMMIT(*ALL)을 만족시키기 위해서만 오로지 잠길 수 있습니다. 행 잠금은 아직까지 반복 읽기하에서 실행됩니다. 사용지정된 잠금과 잠금 지속 시간은 *ALL과 같습니다. 			

분산 관계형 데이터베이스에 대한 저장 및 복원 처리

자료와 프로그램을 저장하고 복원하면 프로그램이나 서버 장애로부터 회복, 서버 간에 정보 교환 또는 오브젝트나 자료 오프라인의 저장이 가능합니다. 분산 관계형 데이터베이스(RDB) 네트워크 내의 각 서버에서 백업에 대한 확실한 정책은 문제가 생겼을 때 서버가 신속하게 복원되고 네트워크 사용자에게 제공될 수 있도록 보장합니다.

테이프와 같이 외부의 매체에 서버를 저장하면 화재나 홍수와 같이 비상 재해로부터 서버 프로그램과 자료를 보호할 수 있습니다. 그러나, 정보는 저장 파일이라고도 부르는 디스크 파일에 저장될 수 있습니다. 저장 파일은 입력 및 출력 조작이나 통신 회선을 통한 또다른 iSeries 서버로 전송하기 위해 사용될 때까지 자료를 저장하는 데 사용되는 디스크 상주 파일(disk-resident file)입니다. 저장 파일을 사용하면 오퍼레이터가 디스켓이나 테이프를 로드할 필요가 없기 때문에 방치된 상태에서 저장 조작이 허용됩니다. 분산 관계형 데이터베이스(RDB)에서, 저장 파일은 보호 메소드로서 또다른 서버로 송신될 수 있습니다.

정보가 복원될 때 정보는 디스켓, 테이프 또는 저장 파일로부터 보조 기억장치으로 기록되고 여기에서 서버 사용자에게 의해 액세스될 수 있습니다.

iSeries 서버는 사용자의 데이터베이스 표와 SQL 오브젝트를 저장하고 복원하기 위해 충분한 만큼의 명령 집합을 갖고 있습니다.

- 라이브러리 저장(SAVLIB) 명령은 하나 또는 그 이상의 콜렉션을 저장합니다.

- 오브젝트 저장(SAVOBJ) 명령은 SQL 표, 뷰 및 색인과 같이 하나 또는 그 이상의 오브젝트를 저장합니다.
- 변경된 오브젝트 저장(SAVCHGOBJ) 명령은 콜렉션을 저장했던 마지막 시간 또는 지정된 날짜로부터 그 이후에 변경된 모든 오브젝트를 저장합니다.
- 저장 파일 자료 저장(SAVSAVFDTA) 명령은 저장 파일의 내용을 저장합니다.
- 시스템 저장(SAVSYS) 명령은 오퍼레이팅 시스템, 보안 정보, 장치 구성 및 서버 값을 저장합니다.
- 라이브러리 복원(RSTLIB) 명령은 콜렉션을 복원합니다.
- 오브젝트 복원(RSTOBJ) 명령은 SQL 표, 뷰 및 색인과 같이 하나 또는 그 이상의 오브젝트를 복원합니다.
- 사용자 프로파일 복원(RSTUSRPRF), 권한 복원(RSTAUT) 및 구성 복원(RSTCFG) 명령은 시스템 저장(SAVSYS) 명령에 의해 저장된 사용자 프로파일, 권한 및 구성을 복원합니다.

이 기능과 명령에 대한 자세한 정보는 iSeries Information Center의 백업 및 복구 주제를 참조하십시오.

분산 관계형 데이터베이스 환경에서 색인 저장 및 복원

SQL 색인을 복원하는 것은 재설정하는 것보다 빠를 수 있습니다. 수많은 요소에 따라 걸리는 시간이 모두 다르지만, 데이터베이스 색인을 재설정하는 데 걸리는 시간은 매 10,000행 당 1분입니다.

색인을 복원한 후, (저널링이 활동 중인지 아닌지에 따라) 최근의 저널 변경을 적용함으로써 최근의 표까지 회생시켜야 할 수도 있습니다. 이 추가 회복 시간을 사용한다 하더라도 색인을 재설정하는 것보다 색인을 복원하는 것이 더 빠릅니다.

서버는 사용자가 색인을 사용할 수 있기 전에 이것의 무결성을 확인합니다. 서버에서 해당 색인을 사용할 수 없다고 판별하면 서버는 색인을 회복하려고 시도합니다. 사용자는 색인이 회복될 때 제어할 수 있습니다. 서버가 이상 종료하면, 그 다음 IPL 시간 동안 서버는 색인이나 뷰 회복을 요구하는 해당 표를 자동으로 나열합니다. 사용자는 다음과 같은 시간대 중 하나에서 색인을 재설정할지 또는 회복을 시도할지에 대해 결정할 수 있습니다.

- IPL 시간 동안
- IPL 시간 후
- 표가 처음 사용될 때

액세스 경로 저장 및 복원에 대한 자세한 정보는 iSeries Information Center의 백업 및 복구 주제를 참조하십시오.

분산 관계형 데이터베이스 환경에서 보안 정보 저장 및 복원

사용자 프로파일을 갱신하고 분산 관계형 데이터베이스(RDB) 네트워크 내의 사용자에 대한 권한을 갱신함으로써 서버 보안 환경을 자주 변경하는 경우, 사용자는 완전한 시스템 저장(SAVSYS) 명령이나 전용 서버를 사용하는 장기 실행 프로세스를 사용하지 않고 보안 정보를 매체나 저장 파일에 저장할 수 있습니다. 보안 자료 저장(SAVSECDTA) 명령을 사용하면 전용 서버를 사용하지 않고도 단 시간에 보안 자료를 저장할 수 있습니다. (SAVSECDTA) 명령을 사용하여 저장된 자료는 사용자 프로파일 복원(RSTUSRPRF) 또는 권한 복원(RSTAUT) 명령을 사용하여 복원할 수 있습니다.

보안 자료 저장(SAVSECDTA) 및 사용자 프로파일 복원(RSTUSRPRF) 명령이 저장하고 복원할 수 있는 보안 정보에 포함되어 있는 항목에는 DRDA TCP/IP 지원이 리모트 서버 사용자 ID와 암호 정보를 저장하고 검색하는 데 사용하는 서버 권한 입력이 있습니다.

분산 관계형 데이터베이스 환경에서 SQL 패키지 저장 및 복원

리모트 서버에서 관계형 데이터베이스에 관련하는 어플리케이션 프로그램이 사전컴파일되어 있고 바인드되어 있을 때, SQL 패키지는 해당 어플리케이션에서 SQL 명령문을 처리하는 데 필요한 제어구조를 포함하도록 어플리케이션 서버(AS)에서 작성됩니다.

SQL 패키지는 iSeries 오브젝트이므로 오브젝트 저장(SAVOBJ) 명령을 사용하여 매체나 저장 파일에 저장하고 오브젝트 복원(RSTOBJ) 명령을 사용하여 복원할 수 있습니다.

SQL 패키지는 저장된 콜렉션과 같은 이름을 갖고 있는 콜렉션으로 복원되어야 하고 이것은 이름 변경될 수 없습니다.

관계형 데이터베이스 디렉토리 저장 및 복원

관계형 데이터베이스 디렉토리는 iSeries 오브젝트가 아닙니다. 관계형 데이터베이스 디렉토리는 IPL 시간에 서버에 의해 열린 파일로 구성됩니다. 따라서, 오브젝트 저장(SAVOBJ) 명령은 이 파일을 직접 저장하는 데 사용될 수 없습니다. 사용자는 관계형 데이터베이스 디렉토리 자료에서 출력 파일을 작성함으로써 관계형 데이터베이스 디렉토리를 저장할 수 있습니다. 이때, 이 출력 파일은 손상되는 경우에 다시 해당 디렉토리에 입력을 추가하는 데 사용될 수 있습니다.

입력이 추가되었고 사용자가 관계형 데이터베이스 디렉토리를 저장하고 싶을 때에는 관계형 데이터베이스 디렉토리 항목 표시(DSPRDBDIRE) 명령에 OUTFILE 매개변수를 지정하십시오. 출력 파일은 테이프, 디스켓 또는 저장 파일에 저장되고 서버로 복원될 수 있습니다. 사용자의 관계형 데이터베이스 디렉토리가 손상되거나 사용자의 서버가 회복되어야 할 경우, CL(제어 언어) 프로그램을 사용하여 관계형 데이터베이스 항목 자료를 포함하는 출력 파일을 복원할 수 있습니다. CL 프로그램은 복원된 출력 파일로부터 자료를 읽고 새로운 관계형 데이터베이스 디렉토리에 입력을 추가하는 CL 명령을 작성합니다.

예를 들어, Spiffy사 MP000 서버용 관계형 데이터베이스 디렉토리는 다음과 같이 RDBDIRM이란 이름의 출력 파일로 송신됩니다.

```
DSPRDBDIRE OUTPUT(*OUTFILE) OUTFILE(RDBDIRM)
```

다음의 샘플 CL 프로그램은 출력 파일 RDBDIRM의 내용을 읽고 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령을 사용하여 관계형 데이터베이스 디렉토리를 재작성합니다. 오래된 디렉토리 항목은 새 항목이 만들어지기 전에 제거된다는 점에 주의하십시오.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

/*****/
/* - Restore RDB Entries from output file created with:           - */
/* -   DSPRDBDIRE OUTPUT(*OUTFILE) OUTFILE(RDBDIRM)             - */
/* -   FROM A V4R2 OR LATER LEVEL OF OS/400                     - */
/*****/
PGM  PARM(&ACTIVATE)

```

```

DCL    VAR(&ACTIVATE) TYPE(*CHAR) LEN(7)

/* Declare Entry Types Variables to Compare with &RWTYPE          */
DCL &LOCAL    *CHAR 1
DCL &SNA      *CHAR 1
DCL &IP       *CHAR 1
DCL &ARD      *CHAR 1
DCL &ARDSNA   *CHAR 1
DCL &ARDIP    *CHAR 1
DCL &RWTYPE   *CHAR 1
DCL &RWRDB   *CHAR 18
DCL &RWRLLOC *CHAR 8
DCL &RWTEXT   *CHAR 50
DCL &RWDEV    *CHAR 10
DCL &RWLLOC   *CHAR 8
DCL &RWNTID   *CHAR 8
DCL &RWMODE   *CHAR 8
DCL &RWTPN    *CHAR 8
DCL &RWSLOC   *CHAR 254
DCL &RWPORT   *CHAR 14
DCL &RWDPGM   *CHAR 10
DCL &RWDLIB   *CHAR 10

DCLF FILE(RDBSAV/RDBDIRM) /* SEE PROLOG CONCERNING THIS          */
IF COND(&ACTIVATE = SAVE) THEN(GOTO CMBLBL(SAVE))
IF COND(&ACTIVATE = RESTORE) THEN(GOTO CMDLBL(RESTORE))
SAVE:
CRTLIB RDBSAV
DSPRDBDIRE OUTPUT(*OUTFILE) OUTFILE(RDBSAV/RDBDIRM)
GOTO CMDLBL(END)

RESTORE:
/* Initialize Entry Type Variables to Assigned Values          */
CHGVAR &LOCAL    '0'          /* Local RDB (one per system) */
CHGVAR &SNA      '1'          /* APPC entry (no ARD pgm)   */
CHGVAR &IP       '2'          /* TCP/IP entry (no ARD pgm) */
CHGVAR &ARD      '3'          /* ARD pgm w/o comm parms    */
CHGVAR &ARDSNA   '4'          /* ARD pgm with APPC parms   */
CHGVAR &ARDIP    '5'          /* ARD pgm with TCP/IP parms */

RMVRDBDIRE    RDB(*ALL)      /* Clear out directory        */

NEXTENT:      /* Start of processing loop   */
RCVF          /* Get a directory entry      */
MONMSG      MSGID(CPF0864) EXEC(DO) /* End of file processing    */
  QSYS/RCVMSG PGMQ(*SAME (*)) MSGTYPE(*EXCP) RMV(*YES) MSGQ(*PGMQ)
  GOTO CMDLBL(LASTENT)
ENDDO

/*          Process entry based on type code                    */
IF (&RWTYPE = &LOCAL) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLLOC) TEXT(&RWTEXT) )

ELSE IF (&RWTYPE = &SNA) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLLOC) TEXT(&RWTEXT) +
  DEV(&RWDEV) LCLLOCNAME(&RWLLOC) +
  RMTNETID(&RWNTID) MODE(&RWMODE) TNSPGM(&RWTPN) )

```

```

ELSE IF (&RWTYPE = &IP) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWSLOC *IP) +
  TEXT(&RWTEXT) PORT(&RWPORT) )

ELSE IF (&RWTYPE = &ARD) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLOC) TEXT(&RWTEXT) +
  ARDPGM(&RWDLIB/&RWDPGM) )

ELSE IF (&RWTYPE = &ARDSNA) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLOC) TEXT(&RWTEXT) +
  DEV(&RWDEV) LCLLOCNAME(&RWLLOC) +
  RMTNETID(&RWNTID) MODE(&RWMODE) TNSPGM(&RWTPN) +
  ARDPGM(&RWDLIB/&RWDPGM) )

ELSE IF (&RWTYPE = &ARDIP) THEN( +
  QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWSLOC *IP) +
  TEXT(&RWTEXT) PORT(&RWPORT) +
  ARDPGM(&RWDLIB/&RWDPGM) )

GOTO CMDLBL(NEXTENT)

LASTENT:
RETURN
DLTLIB RBSAV
END

```

ENDPGM

다음은 위에서 기술한 유형의 출력파일을 모두 사용할 수 없는 경우에 디렉토리를 복원하는 대체 방법입니다. 이 방법은 저장된 서버에서 오브젝트를 추출하여 다른 라이브러리에 복원한 다음 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령으로 수동으로 항목을 입력하는 것입니다.

관계형 데이터베이스 디렉토리를 구성하는 파일은 시스템 저장(SAVSYS) 명령이 실행될 때 저장됩니다. 관계형 데이터베이스 디렉토리를 포함하는 실제 파일은 다음의 오브젝트 복원(RSTOBJ) 명령으로 해당 저장 매체로부터 사용자의 라이브러리로 복원될 수 있습니다.

```

RSTOBJ      OBJ(QADBXRDBD) SAVLIB(QSYS)
            DEV(TAP01) OBJTYPE(*FILE)
            LABEL(Qpppppppvrmtx0003)
            RSTLIB(your lib)

```

이 예에서, 관계형 데이터베이스 디렉토리는 테이프로부터 복원됩니다. LABEL 매개변수에서 *ppppppp* 문자는 Operating System/400의 제품 코드를 나타냅니다(예를 들어, 버전 5 릴리스 3에 대해 5722SS1). LABEL 매개변수에서 *vr*은 해당 버전, 릴리스 및 OS/400의 수정 레벨입니다. LABEL 매개변수에서 *xx*는 현재 서버 언어값의 마지막 두 자리수입니다. 예를 들어, 2924는 영어에 대한 표시이고, 따라서 *xx*의 값은 24입니다.

사용자는 이 파일을 사용자의 라이브러리로 복원한 다음에 이 파일의 정보를 사용하여 관계형 데이터베이스 디렉토리를 수동으로 재작성할 수 있습니다.

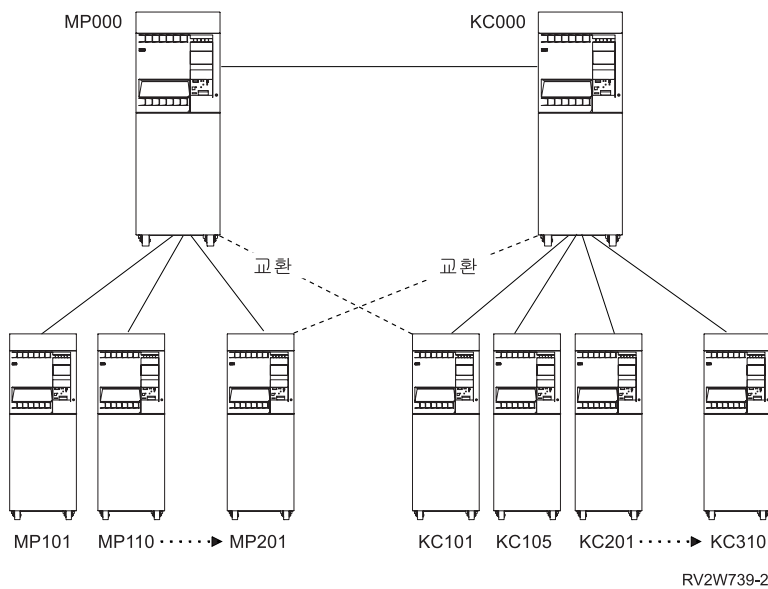
분산 관계형 데이터베이스에 대한 네트워크 중복 문제

네트워크 중복은 분산 관계형 데이터베이스(RDB) 네트워크의 사용자가 해당 네트워크의 관계형 데이터베이스에 액세스할 수 있도록 다양한 방법을 제공합니다. 어플리케이션 리퀘스터(AR)에서부터 어플리케이션 서버(AS)까지 단 하나의 통신 경로가 있으면 통신 회선이 다운될 때 AR의 사용자는 AS 관계형 데이터베이스에 액세스하지 못합니다. 이 때문에, 네트워크 중복 제공은 Spiffy사의 분산 관계형 데이터베이스(RDB) 관리자에게 중요합니다.

예를 들어, 서비스 계약이나 고객 부품 구매서를 판매 대리점에 발행한다고 가정해 보십시오. 고객이 서비스나 부품 구매를 기다리고 있을 때, 서비스 직원은 작업 스케줄을 짜거나 부품을 판매하기 위해 기업망 정보의 권한 범위 내에 있는 전체 표에 액세스해야 합니다.

로컬 서버가 다운되면 어떠한 작업을 실행할 수 없습니다. 로컬 서버는 실행 중이지만 리모트 서버에 대한 요구가 작업 처리에 필요하고 이 때에 해당 리모트 서버가 다운되는 경우, 해당 요구를 처리할 수 없습니다. Spiffy사의 예에서, 이것은 판매 대리점이 지역의 재고 관리 센터로부터 부품 정보를 요구할 수 없다는 의미일 수도 있습니다. 또한, 많은 AR 작업을 처리하는 AS가 다운되면 AR 가운데 그 어느 것도 해당 요구를 완료할 수 없습니다. Spiffy사 네트워크의 경우, 지역 센터에서 다운이 발생하면 이것이 지원하는 어플리케이션 서버는 부품을 주문할 수 없습니다.

해당 지역의 판매 대리점에 지역의 재고 자료에 대한 액세스를 제공하는 것은 Spiffy사 분산 관계형 데이터베이스(RDB) 관리자에게 중요한 일입니다. 자료에 대해 네트워크를 통한 경로 제공 방법에는 여러 가지가 있을 수 있습니다. Spiffy사의 원래의 네트워크 구성은 끝 노드의 판매 대리점을 이것의 해당 네트워크 노드의 지역 센터에 링크하였습니다.



RV2W739-2

그림 11. 대체 네트워크 경로

일부 판매 대리점의 대안은 다른 지역 센터로 교환 회선 연결을 하는 것일 수도 있습니다. 로컬 지역 센터가 네트워크를 사용할 수 없는 경우, 또 하나의 AS에 액세스하여 해당 요구의 판매 대리점에서 작업을 하는 데

필요한 정보를 얻을 수 있습니다. 141 페이지의 그림 11에서, MP000 서버에 의해 제공되는 일부 AR은 KC000 서버에 링크를 설치하여, MP000 서버를 사용할 VRYCFG(다양한 구성) 또는 구성 상태에 대한 작업 (WRKCFGSTS) 명령은 서버 오퍼레이터나 분산 관계형 데이터베이스(RDB) 관리자가 필요 시에 회선을 ON(켜짐)으로 변경하고 1차 AS를 사용할 수 있을 때에는 회선을 OFF(꺼짐)으로 변경하도록 사용될 수 있습니다.

또 하나의 대안은 보다 큰 지역의 판매 대리점 중 하나가 다른 판매 대리점의 AS로 활동하는 경우가 될 수 있습니다. 그림 12에서 141 페이지의 그림 11에서, 미니애폴리스로 링크가 다운되면 어떤 판매 대리점도 재고에 대해 또다른(끝 노드)에 조회할 수 없습니다. 위에서 설명한 구성은 판매 대리점 중 하나가 APPN 네트워크 노드로서 구성되고 다른 지역의 판매 대리점으로부터 해당 판매 대리점으로 회선이 설정되도록 변경될 수도 있습니다.

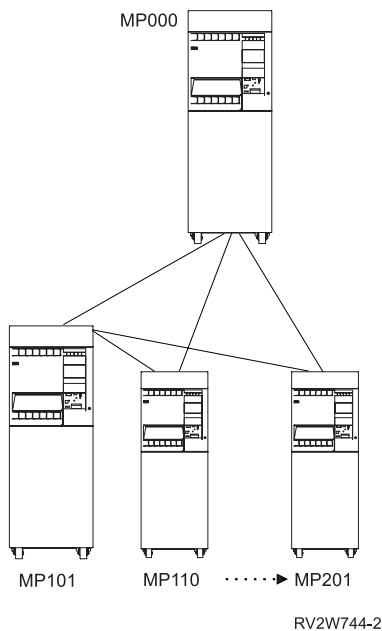


그림 12. 대체 어플리케이션 서버

분산 관계형 데이터베이스 네트워크에서 자료 중복

또한, 분산 관계형 데이터베이스(RDB)에서 자료 중복은 분산 관계형 데이터베이스(RDB) 네트워크의 사용자가 해당 네트워크에서 데이터베이스에 액세스할 수 있는 다양한 방법을 제공합니다. 자료 중복 전략을 작성하는 데 분산 관계형 데이터베이스(RDB) 관리자가 검사하는 문제는 통신 경로가 해당 자료에 제공되어 있는지 확인하는 것 이상으로 복잡합니다. 표는 네트워크에서 서버 전체를 통해 복사될 수 있고 자료의 스냅샷은 자료 가용성을 제공하는 데 사용될 수 있습니다. DataPropagator Relational Capture 및 Apply/400 제품은 이 기능을 제공할 수 있습니다.

아래의 그림은 MP000 서버 분산 관계형 데이터베이스(RDB)의 복사가 KC000 서버에 저장될 수 있고, KC000 서버 분산 관계형 데이터베이스(RDB)의 복사가 MP000 서버에 저장될 수 있다는 것을 나타내고 있습니다. 한

지역으로부터 어플리케이션 리퀘스터(AR)는 상대의 어플리케이션 서버(AS)로 링크되어 AR의 관계형 데이터베이스의 반복된 복사를 조회하고 갱신할 수 있습니다.

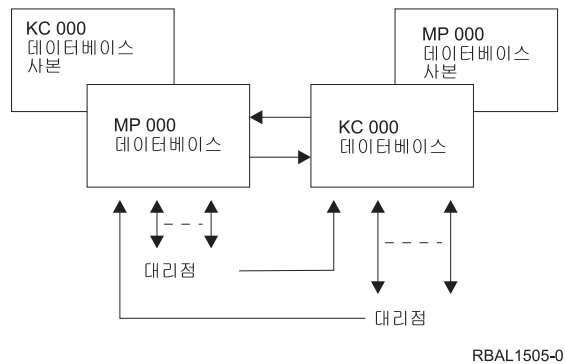


그림 13. 데이터베이스 중복 예

관리자는 분산 관계형 데이터베이스(RDB) 처리를 할 수 있는 가장 효율적이고 효과적인 전략에 대해서 결정해야 합니다. 대체 전략이 이 시나리오에 포함될 수도 있습니다.

하나의 대안은 MP000을 사용할 수 없을 때 이것의 AR을 KC000 서버에 연결하여 MP000 분산 관계형 데이터베이스(RDB)의 읽기 전용 스냅샷을 조회함으로써 서비스 작업을 스케줄할 수 있도록 하는 것입니다.

DataPropagator Relational/400은 표의 읽기 전용 복사(또는 스냅샷)을 정기적으로 리모트 서버에 제공할 수 있습니다. Spiffy사에서, 이것은 각 업무일의 처음 또는 마지막에 해당될 수 있습니다. 이 예에서, MP000 데이터베이스 스냅샷은 판매 대리점이 계획용으로만 사용할 수 있도록 24시간 동안 last-point-in-time 영상을 제공합니다. MP000 서버가 회선에서 연진행할 때, 해당 AR은 MP000 분산 관계형 데이터베이스(RDB)를 조회하여 재고 요구나 스냅샷에서 조회된 다른 작업을 완전히 처리하게 합니다.

또 하나의 대안은 Spiffy사가 판매 대리점 사용자가 지역 AS를 사용할 수 없을 때 또다른 AS에서 복제된 표를 갱신할 수 있도록 하는 것입니다.

예를 들어, MP000 데이터베이스에 정상적으로 연결된 AR은 KC000 서버에서 복제된 MP000 데이터베이스에 연결하여 작업을 처리할 수 있습니다. MP000 서버를 다시 사용할 수 있을 때, MP000 관계형 데이터베이스는 KC000 위치에서 이 데이터베이스의 복제된 표에 기원하는 활동으로부터의 저널 항목을 적용함으로써 갱신될 수 있습니다. 이 저널 항목이 원래의 MP000 표에 적용되었을 때, 분산 관계형 데이터베이스(RDB) 사용자는 AS로서 다시 MP000에 액세스할 수 있습니다.

각 지역의 서버에서 저널 관리 프로세스는 관계형 데이터베이스 전체를 갱신합니다. 이 상황에서 저널 관리 복사 활동의 양은 이 서버에서 잠재적인 역실행의 결과가 생길 수 있으므로 검사해야 합니다.

제 8 장 분산 관계형 데이터베이스(DRDB) 성능

서버에서 실행되는 어플리케이션 프로그램의 종류에 관계없이 성능은 항상 주요 관심사입니다. 분산 관계형 데이터베이스의 경우, 네트워크, 서버 및 어플리케이션 성능이 모두 중요합니다. 주 기억장치와 보조 기억장치의 크기와 구성이 서버 성능에 영향을 줄 수 있습니다. SQL 프로그램의 장단점을 알고 있으면 성능을 향상시킬 수 있습니다. 보다 자세한 정보는 제 9 장 『분산 관계형 데이터베이스(DRDB) 처리』을 참조하십시오.

사용자의 네트워크, 서버 및 데이터베이스 설계를 개선하는 방법에 관한 세부사항은 다음 주제를 참조하십시오.

- 네트워크를 통한 분산 관계형 데이터베이스 성능 개선
- 서버를 통한 분산 관계형 데이터베이스 성능 개선
- 데이터베이스를 통한 분산 관계형 데이터베이스 성능 개선


네트워크를 통한 분산 관계형 데이터베이스 성능 개선

여러 가지 방법으로 네트워크 성능을 개선할 수 있습니다. 다음과 같은 방법이 있습니다.

- 회선 속도
- 페이징
- 프레임 크기
- RU 크기
- 연결 유형(비전환 대 전환)

주: 비보호 대화식은 연결이 RUW 연결 관리를 사용하는 프로그램으로부터 실행되었거나 연결한 프로그램이 확약 제어 하에서 실행되지 않을 경우, 또는 연결된 데이터베이스가 사용된 프로토콜에 대해 2단계 확약을 지원하지 않을 경우, DRDA연결에 사용됩니다. 만일 자료 특성이 하나의 데이터베이스 관리 시스템(DBMS)에만 영향을 미치는 트랜잭션(transaction)과 같은 것이라면, RUW 연결 관리를 사용하는 프로그램 또는 확약 제어없이 실행되는 프로그램으로 연결을 형성하면 오버헤드가 2단계 확약 흐름과 관련되지 않도록 피할 수 있습니다.

또한 대화식이 DDMCNV(*KEEP)를 지속적으로 사용하며, 그것이 보호 대화식이라면, 작업 단위 동안 DRDA 또는 DDM 처리에 대화식이 사용되었더라도 관계없이 2단계 확약 흐름이 송신됩니다. 따라서 DDMCNV(*KEEP)를 실행할 때, 가능하다면 비보호 대화식으로 실행하는 것이 좋습니다. 만일 보호 대화식으로 실행한다면, DDMCNV(*DROP)를 사용해야 하고 앞으로의 작업 단위에서 대화식을 사용되지 않는다는 다음 확약 때의 연결과 대화식을 종료하기 위해서 RELEASE 명령문을 사용해야 합니다.

자세한 정보는 *Communications Management*  매뉴얼을 V5R1 보충 매뉴얼 웹 사이트에서 참조하십시오. RU 크기 및 페이싱에 대한 정보는 APPC, APPN 및 HPR 주제를 참조하십시오.

기타 통신 관련 성능에 대한 고려사항은 iSeries Information Center의 TCP/IP 설정 주제를 참조하십시오.

서버를 통한 분산 관계형 데이터베이스 성능 개선

충분한 서버 성능을 발휘하려면 서버 자원 간의 적절한 균형이 필요합니다. 임의의 자원을 남용하는 것은 성능에 좋지 않은 영향을 미칩니다.

이 섹션에서는 서버 성능을 감시하도록 돕기 위해 제공된 서버 명령어를 설명합니다.

실행 분석을 돕기 위해 iSeries 성능 분석 툴 사용권 프로그램을 사용할 수 있습니다. 또한 제공된 시스템 명령어를 사용하여 서버 성능을 감시할 수 있습니다.

- 시스템 상태에 대한 작업(WRKSYSSTS) 명령
- 디스크 상태에 대한 작업(WRKDSKSTS) 명령
- 활동 작업에 대한 작업(WRKACTJOB) 명령

이 명령어를 사용할 때는 반드시 일반 활동 레벨 중에 서버 성능을 감시해야만 합니다. 예를 들어, 서버에 작업이 실행되지 않는 중 언은 통계는 서버 성능 평가에 아무 의미가 없습니다. 서버 성능을 감시하기 위해서 다음의 단계를 완료하십시오.

1. (WRKSYSSTS), (WRKDSKSTS) 또는 (WRKACTJOB) 명령을 입력하십시오.
2. 서버가 최소한 5분간 자료를 수집하도록 하십시오.
3. 화면을 정리하기 위해 F5(화면정리) 키를 누르십시오. 실행 자료를 표시하십시오.
4. 신규 자료를 바탕으로 서버를 조절하십시오.

가동이 중지된 시간 카운터를 재시작하려면 F10(재시작) 키를 누르십시오.

서버 상태 및 디스크 상태에 대한 자세한 작업 방법은 작업 관리 주제의 성능 조정 관련 장을 참조하십시오.

이전에 설명했던 활동 작업에 대한 작업(WRKACTJOB) 명령의 기능 중 하나가 서버의 성능을 측정하는 것입니다. 활동 작업에 대한 작업에 대해서는 100 페이지의 『분산 관계형 데이터베이스에서 활동 작업(Job)에 대한 작업』에서 참조하실 수 있습니다.

시스템의 성능을 관찰할 때 시스템 상태에 대한 작업(WRKSYSSTS)과 활동 작업에 대한 작업(WRKACTJOB) 명령을 모두 사용하십시오. 설정해 놓은 목표에 대하여 각 감시 기간 동안 서버 성능의 측정을 검사하고 평가해야 합니다.

몇 가지 일반 측정에는 다음과 같은 사항이 포함되어 있습니다.

- (WRKACTJOB) 화면에서 제공되는 대화식 처리량 및 응답 시간.
- 일괄처리 처리량. 활동 일괄처리 작업에 대한 AuxIO 및 CPU% 값을 감시하십시오.

- 스푼 처리량. 활동 출력기에 대한 AuxIO 및 CPU% 값을 감시하십시오.

변경 조절을 할 때마다 반드시 모든 기본 성능 측정을 측정하고 비교해야 합니다. 한 번에 하나씩 조정하고 평가하십시오.

데이터베이스를 통한 분산 관계형 데이터베이스 성능 개선

분산 관계형 데이터베이스(DRDB) 성능은 15 페이지의 제 2 장 『분산 관계형 데이터베이스의 계획 및 설계』에서 설명되었듯이 데이터베이스의 개괄 설계에 따라 달라집니다. 분배 자료 위치, 사용하는 확약 제어 레벨 및 SQL 색인 설계 등 모든 것이 성능에 영향을 미칩니다.

데이터베이스 성능을 최적화하는데 관한 자세한 정보는 다음 주제를 참조하십시오.

- DRDA 자료 위치 결정
- DRDA블로킹에 영향을 주는 요소
- DRDA 조회 블록의 크기에 영향을 미치는 요소

DRDA 자료 위치 결정

어플리케이션과 어플리케이션이 필요로 하는 자료 사이에 네트워크가 연결되면 실행 속도가 저하되므로, 자료를 송신할 장소를 결정할 때는 다음 사항을 고려하십시오.

- 자료를 사용하는 트랜잭션
- 트랜잭션 실행 빈도
- 트랜잭션이 송신하거나 수신하는 자료량

만일 어플리케이션이 자주 실행하거나 다량의 자료를 송신 또는 수신하는 트랜잭션을 동반할 경우, 반드시 트랜잭션을 자료와 같은 위치에 보유하도록 노력하십시오. 예를 들어, 1초에 수 회 실행되거나 한 번에 수백 행의 자료를 수신하는 어플리케이션은 어플리케이션과 자료가 같은 서버 안에 있을 경우, 훨씬 뛰어난 성능을 발휘할 것입니다. 반대로, 어플리케이션이 사용도가 낮거나 한 번에 단지 미미한 양의 자료만 송신 또는 수신하는 트랜잭션을 동반할 경우라면, 어플리케이션과 자료를 각 다른 위치에 배치시키도록 하십시오.

DRDA블로킹에 영향을 주는 요소

어플리케이션 리퀘스터(AR) 및 어플리케이션 서버(AS)간의 자료가 전송될 때의 블로킹 발생 여부는 매우 중요한 실행 요소입니다. 자료 블록으로서 전송된 행의 그룹은 한 번에 하나의 행을 송신할 때보다 훨씬 적은 통신 오버헤드를 필요로 합니다. 다른 iSeries 서버에 연결될 때 블로킹을 제어하는 한 가지 방법은 버전 2 릴리스 2 및 OS/400 오퍼레이팅 시스템의 다음 버전의 SQL 복수 행 INSERT 및 복수 행 FETCH 명령어를 사용하는 것입니다. 복수 행 FETCH 명령어는 중대한 오류나 자료 끝이 충돌하지 않는 한, FOR n ROWS절에 지정된 행 수를 블로킹합니다. 다음에서는 블로킹이 단일 행 FETCH에 의해 발생할 경우를 판별하기 위한 규칙을 설명합니다.

어플리케이션 리퀘스터와 어플리케이션 서버간 조회 자료의 블로킹을 방지하는 조건도 설명합니다. 이 조건은 복수 행 FETCH 명령문 사용을 위해서는 사용되지 않습니다. 다음 각각의 경우에 대해 나열된 조건을 참조하면 블로킹 발생을 예방할 수 있습니다.

경우 1: iSeries용 DB2 UDB간 블로킹

다음의 경우 블로킹이 발생하지 않습니다.

- 커서의 갱신이 가능합니다(주 1을 참조).
- 커서의 갱신이 임시적으로 가능합니다(주 2를 참조).
- ALWBLK(*NONE) 사전컴파일 옵션이 사용되었습니다.
- 확약 제어 레벨이 *CS이고 OS/400의 레벨이 버전 3 릴리스 1 이전의 레벨입니다.
- 확약 제어 레벨이 *ALL이고 외부 부속 선택에 다음 요소가 포함되지 않습니다.
 - DISTINCT 키워드
 - UNION 오퍼레이터
 - ORDER BY절 및 절 안의 필드 길이 합의 정렬이 필요합니다.
 - 서버 데이터베이스 파일(서버 데이터베이스 파일은 라이브러리 QSYS 내에 있는 QADBxxxx라고 명명된 파일이며 임의의 뷰가 이 파일에 대해 빌드되어 있습니다.)에 관한 참조.
- 행 크기가 약 2K 이상입니다. 만일 리모트 명령 제출(SBMRMTCMD) 명령 또는 저장된 프로시저어가 디폴트 어플리케이션 서버 데이터베이스 버퍼 크기를 확장하도록 사용되었다면, 행 크기는 데이터베이스 파일과 대체(OVRDBF) SEQONLY 레코드 수 매개변수의 스펙에서 비롯된 어플리케이션 서버 버퍼의 크기의 약 절반 이상이 될 것입니다. (리모트 작동을 위한 (OVRDBF) 명령의 경우, 반드시 OVRSCOPE(*JOB)를 지정해야 한다는 사실에 주의하십시오.)
- 커서의 화면이동이 가능하도록 선언되었습니다(DECLARE...SCROLL CURSOR...). 또한 FETCH 명령문 내에 지정된 화면이동 옵션이 RELATIVE, PRIOR 또는 CURRENT 중 하나입니다. (복수 행 FETCH 명령문이 완료되지 않은 경우, 위에 언급된 대로입니다.)

경우 2: iSeries용 DB2 UDB 제품에서 비iSeries용 DB2 UDB 제품으로의 블로킹

다음의 경우 블로킹이 발생하지 않습니다.

- 커서의 갱신이 가능합니다(주 1을 참조).
- 커서의 갱신이 임시적으로 가능합니다(주 2를 참조).
- ALWBLK(*NONE) 사전컴파일 옵션이 사용되었습니다.
- 행 크기가 약 16k 이상입니다.

경우 3: 비iSeries용 DB2 UDB 제품에서 iSeries용 DB2 UDB 제품으로의 블로킹

다음의 경우 블로킹이 발생하지 않습니다.

- 커서의 갱신이 가능합니다(주 1을 참조).
- 커서의 갱신이 임시적으로 가능합니다(주 2를 참조).
- 사전컴파일 또는 바인드 옵션이 사용되며 단일 행 프로토콜이 되기 위한 패키지 디폴트 값을 지정하는 원인이 되었습니다.
 - DB2의 경우, 이와 같은 실행에 대한 옵션이 없습니다.
 - DB2 UDB for VM의 경우, SQLPREP 상에 NOBLOCK 키워드가 있습니다(디폴트).

- DB2/2의 경우, SQLPREP 상의 또는 SQLBIND 상의 /K=NO입니다.
- 행 크기가 약 0.5*QRYBLKSIZ 이상입니다. 디폴트 및 최대 QRYBLKSIZ 값은 다음과 같습니다.

표 7. QRYBLKSIZ

DB2 제품	디폴트 QRYBLKSIZ	버전 8의 최대 QRYBLKSIZ
DB2 Universal Driver for SQLJ 및 JDBC	32K	32K
DB2 UDB for z/OS	32K	64K
DB2 UDB for VM	8K	32K
DB2 Connect	32K	64K

최신 DRDA레벨에서 서버는 결과 세트를 리턴할 때 고정된 조회 블록 크기로 제한되지 않는 모드에서 작동하도록 선택할 수 있습니다.

- 커서는 화면이동 커서로 정의되며 블록 커서는 어플리케이션에서 사용되지 않습니다.

DRDA 블로킹 규칙 요약

요약에서 이 규칙은(주를 포함하여) 모두 어떤 특별한, 또는 비정상적인 조건에서는 블로킹이 발생한다는 내용에 대한 것입니다. 다음 두 가지 경우 모두에서 발생합니다.

- 커서가 읽기 전용일 때 발생합니다(주 3을 참조).
 - 어플리케이션 리퀘스터 또는 어플리케이션 서버 중 하나가 iSeries용 DB2 Universal Database 제품이 아닌 경우에 발생합니다.
 - 어플리케이션 리퀘스터 및 어플리케이션 서버 모두가 iSeries용 DB2 Universal Database 제품이고 ALWBLK(*ALLREAD)가 지정되었으며 COMMIT(*ALL)은 지정되지 않았습니다.
- 만일 COMMIT(*ALL)이 지정되지 않고 다음 조건에 모두 해당할 경우, 블로킹이 발생합니다.
 - SELECT문 내에 FOR UPDATE OF절이 없습니다.
 - 프로그램 내의 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 명령문이 없습니다.
 - 프로그램에 동적 SQL 명령문이 없거나 사전컴파일/바인드 옵션이 제한된 프로토콜을 요청하도록 사용되었습니다(DB2 Connect에 대한 /K=ALL, DB2 UDB for iSeries에 대한 ALWBLK(*ALLREAD), DB2 UDB for z/OS에 대한 CURRENTDATA(NO), DB2 UDB for VM에 대한 SBLOCK).

주:

1. 만일 읽기 전용이 아니고 다음 조건 중 하나에 해당할 경우 커서는 갱신이 가능합니다(주 3을 참조).
 - SELECT문에 FOR UPDATE OF절이 들어 있습니다.
 - 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 구가 프로그램 내에 있습니다.
2. 만약 읽기 전용이 아니고(주 3을 참조), 프로그램에 EXECUTE 또는 EXECUTE IMMEDIATE 명령문이 있고(또는 iSeries 서버가 아닌 다른 제품에 연결되었을 때, 동적 명령문이 존재할 경우) 사전컴파일 또는 바인드 옵션이 사용되어 단일 행 프로토콜이 되기 위한 패키지 디폴트 값을 지정하는 원인이 된 경우, 커서는 갱신이 임시적으로 가능합니다.
 - iSeries용 DB2 Universal Database의 경우, ALWBLK(*READ) 사전컴파일 옵션입니다(디폴트).
 - DB2의 경우, BIND PACKAGE 상의 CURRENTDATA(YES)가 있습니다(디폴트).

- DB2 UDB for VM의 경우, SQLPREP 상에 SBLOCK 키워드가 있습니다.
 - DB2/2의 경우, SQLPREP 상의 또는 SQLBIND 상의 /K=UNAMBIG입니다(디폴트).
3. 만일 다음 조건 중 하나 이상에 해당할 경우, 커서는 읽기 전용입니다.
- DECLARE CURSOR 명령문이 ORDER BY절을 지정했으나 FOR UPDATE OF절은 지정하지 않았 습니다.
 - DECLARE CURSOR 명령문이 FOR FETCH ONLY절을 지정했습니다.
 - DECLARE CURSOR 명령문이 DYNAMIC(OS/400 전용)없이 SCROLL 키워드를 지정했습니다.
 - 커서, 뷰 또는 커서 관련 외부 부속선택 내에 적용된 논리 파일 등에 대해서는 다음 조건 중 하나 이 해당됩니다.
 - 외부 부속 선택에는 DISTINCT 키워드, GROUP BY절, HAVING 절 또는 외부 부속 선택 내의 열 함수가 들어있습니다.
 - SELECT에 결합 함수가 있습니다.
 - SELECT에 UNION 오퍼레이터가 있습니다.
 - SELECT에 외부 전용 부속 선택 표와 똑같은 표에 적용되는 부속 조회(subquery)가 있습니다.
 - SELECT에 임시 파일로서 복사된 복잡한 논리 파일이 들어 있습니다.
 - 모든 선택된 열은 표현식, 스칼라 함수 또는 상수 등입니다.
 - 참조적인 논리 파일의 모든 열은 입력 전용입니다(OS/400 전용).

DRDA 조회 블록의 크기에 영향을 미치는 요소

- | 만일 대량의 자료가 조회를 통해 리턴된다면, 조회 자료 블록 크기를 늘려서 실행 성능을 개선할 수 있습니다.
- | 조회 내에 직접 관여하는 서버 유형에 따라 실행되는 방법이 달라집니다. 비슷하지 않은 환경에서라면, 조회
- | 열기 명령으로 송신된 매개변수에 의해 어플리케이션 리퀘스터에서 조회 블록의 크기가 판별됩니다. iSeries 서
- | 버가 어플리케이션 리퀘스터(AR)일 때, 시스템은 초기에 항상 32K의 조회 블록 크기가 필요합니다. 복수 블
- | 록을 전송해야 하는 큰 조회의 경우 요구한 각 연속 조회 블록 크기가 증가됩니다. 다른 유형의 어플리케이션
- | 리퀘스터 유형은 사용자에게 어떤 블록 크기를 사용할지 선택할 수 있도록 허용합니다. DB2 Universal Driver
- | for SQLJ와 JDBC, DB2 UDB for z/OS, DB2 UDB for VM 및 DB2 Connect에 대한 디폴트 조회 블록
- | 크기는 32K, 32K, 8K 및 32K입니다. iSeries용 DB2 UDB 서버가 같은 종류가 아닌 어플리케이션 리퀘스
- | 터에 연결되었을 때 어플리케이션 리퀘스터로서 사용된 플랫폼에 대한 제품 문서를 참조하십시오.

iSeries용 DB2 UDB으로부터 iSeries용 DB2 UDB으로 이어지는 환경에서, 조회 블록 크기는 데이터베이스 관리자에 의해 사용된 버퍼 크기에 의해 판별됩니다. 디폴트 크기는 4K입니다. 버전 2 릴리스 3 또는 그보다 상위 레벨의 어플리케이션 서버에서는 이 크기가 바뀔 수도 있습니다. 크기를 바꾸기 위해서는 리모트 명령 제출(SBMRMTCMD) 명령을 사용하여 송신한 후 어플리케이션 서버(AS)에 데이터베이스 파일과 대체(OVRDBF) 명령을 실행하십시오. 대체된 파일 명외에도, OVRSCOPE(*JOB) 및 SEQONLY(*YES nnn)가 (OVRDBF) 명령문에 포함되어야 합니다. 블록 당 필요한 레코드 수는 SEQONLY 매개변수에서 NNN으로 대체됩니다. 데이터베이스 버퍼 크기를 늘리면, 통신 오버헤드를 절감하고 행을 불러오기 위한 데이터베이스 관리자 호출 수를 절감할 수 있습니다.

또는 iSeries 서버가 아닌 제품 또는 iSeries 서버 내에서 SQL CALL 명령문(저장 프로시저)을 사용하여 조회 블록 크기를 변경할 수 있습니다.

제 9 장 분산 관계형 데이터베이스(DRDB) 처리

분산 관계형 데이터베이스(DRDB)에 액세스하는 중 문제점이 발생할 때, 관리자가 맡아야 할 작업은 다음과 같습니다.

- 문제점의 본질을 판별합니다.
- 문제점이 어플리케이션에 대한 것인지 로컬 또는 리모트 시스템에 대한 것인지 판별합니다.

그 후에 문제점을 해결하거나 문제점 해결을 위한 고객 지원 도움을 받아야 합니다. 도움을 받기 위해서는 다음 사항이 필요합니다.

- OS/400 프로그램 지원에 대한 이해.
- 어플리케이션 리퀘스터(AR) 또는 어플리케이션 서버(AS)에 대한 문제인지 결정하기 위한 방법.
- OS/400 문제점 관리 기능에 대한 지식.

분산 관계형 데이터베이스 문제점에 대한 자세한 정보는 다음 주제를 참조하십시오.

- iSeries 문제점 처리 개요
- 분리되는 분산 관계형 데이터베이스(DRDB) 문제
- 분산 관계형 데이터베이스 사용자에게 대한 작업
- 어플리케이션 문제점
- 실패 보고를 위한 자료 수집
- 첫 번째 실패 자료 캡처(FFDC) 자료 찾기
- application server 문제점을 진단하기 위한 서비스 작업 시작
- 시스템 및 통신 문제점

분산 관계형 데이터베이스(DRDB) 내의 문제점 진단에 대한 자세한 정보는 *Distributed Relational Database Problem*을 참조하십시오.

iSeries 문제점 처리 개요

사용자는 OS/400 프로그램으로 로컬 및 리모트 iSeries 서버 상에 발생하는 사용자 및 시스템 감지 문제점을 관리할 수 있습니다. 문제점 처리 지원에는 다음과 같은 사항이 포함되어 있습니다.

- 초기 문제점 처리 정보에 대한 메시지
- 시스템 감지 문제점에 대한 자동 경고
- 통합 문제점 기록 및 트랙
- FFDC(First-Failure Data Capture) 지원
- 전자 고객 지원 서비스 요청
- 전자 고객 지원, 프로그램 임시 수정(PTF) 요청

iSeries 서버 및 접속된 장치는 몇 가지 유형의 문제를 감지할 수 있습니다. 이 문제를 시스템 감지 문제점이라고 부릅니다. 문제점이 감지되면, 몇 가지 작업이 발생합니다.

- 제품 활동 기록부 내에 항목이 작성됩니다.
- 문제점 레코드가 작성됩니다.
- QSYSOPR 메시지 대기행렬에 메시지가 송신됩니다.

오류 기록부 및 문제점 레코드 내에 정보가 저장됩니다. 심각한 문제점이 감지되면, FFDC 정보에 대한 스펙 파일도 작성됩니다. 오류 기록부 및 문제점 레코드에는 다음의 정보가 수록되어 있습니다.

- 가상 제품 자료(VPD)
- 구성 정보
- 참조 코드
- 관련 장치명
- 추가 실패 정보

사용자 감지 문제점은 대부분 다음의 문제점을 발생시키는 원인인 프로그램 오류와 관련이 있습니다.

- 작업 문제점
- 부정확한 출력
- 프로그램 실패 표시 메시지
- 시스템에 의해 감지되지 않는 장치 결함
- 불완전한 실행

사용자가 문제점을 감지하면, 프로그램 분석이 실행되거나 Operational Assistant* USERHELP 메뉴에서 문제점 해결을 위해 정보를 저장하는 옵션을 선택할 때까지 서버에서 정보를 수집하지 않습니다.

iSeries 서버에서는 문제점 기록부 및 문제점 관리자를 사용하여 사용자 및 시스템 감지 문제점을 모두 트래킹합니다. 문제점이 감지되었을 때부터(OPENED) 문제점이 해결될 때까지(CLOSED) 트래킹을 통해 사용자를 보조하도록 문제점 상태가 유지됩니다. 자세한 정보는 176 페이지의 『iSeries 문제점 기록부』를 참조하십시오.

분리되는 분산 관계형 데이터베이스(DRDB) 문제

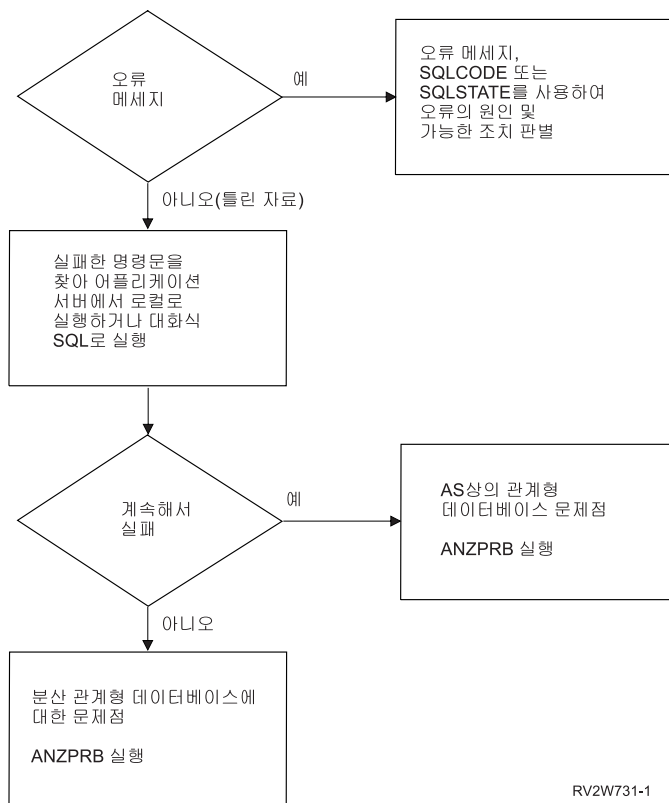
분산 관계형 데이터베이스(DRDB) 어플리케이션을 실행할 때 발생한 문제점은 두 가지 일반적인 증후를 보입니다.

- 사용자가 부정확한 출력을 수신합니다.
- 예상 시간 내에 어플리케이션이 완료되지 않았습니다.

아래의 다이어그램 및 프로시더는 어플리케이션 프로그램 문제점, 성능 관련 문제 및 서버 관련 문제점 등으로 문제점을 분류하는 방법을 일반적으로 나타내고 있습니다. 따라서 표준 iSeries 서버 문제점 분석 방법을 사용하여 문제점을 해결할 수 있습니다.

DRDA의 부정확한 출력 문제

만일 오류 메시지를 수신했다면, 오류 메시지 SQLCODE 또는 SQLSTATE를 사용하여 문제점의 원인을 판별합니다. 그림 14를 참조하십시오. 메시지 설명에서는 문제점이 무엇인지 표시하고 정확한 조치를 제공합니다. 만일 오류 메시지를 수신하지 못했다면, 분산 관계형 데이터베이스(RDB)가 실패의 원인인지 여부를 판별해야 합니다. 판별을 위해서는 실패한 명령문을 **application server(AS)**에서 로컬 실행하거나 대화식 SQL(Structured Query Language)을 사용하여 어플리케이션 서버에 명령문을 실행합니다. 만일 문제점을 로컬로 작성할 수 있다면, 문제점은 분산 관계형 데이터베이스(RDB) 지원에 대한 것이 아닙니다. 이 조작의 결과에 따라 지원 스텝에 대한 특정 정보를 제공하기 위해서는 iSeries 서버 문제점 분석을 사용하십시오.



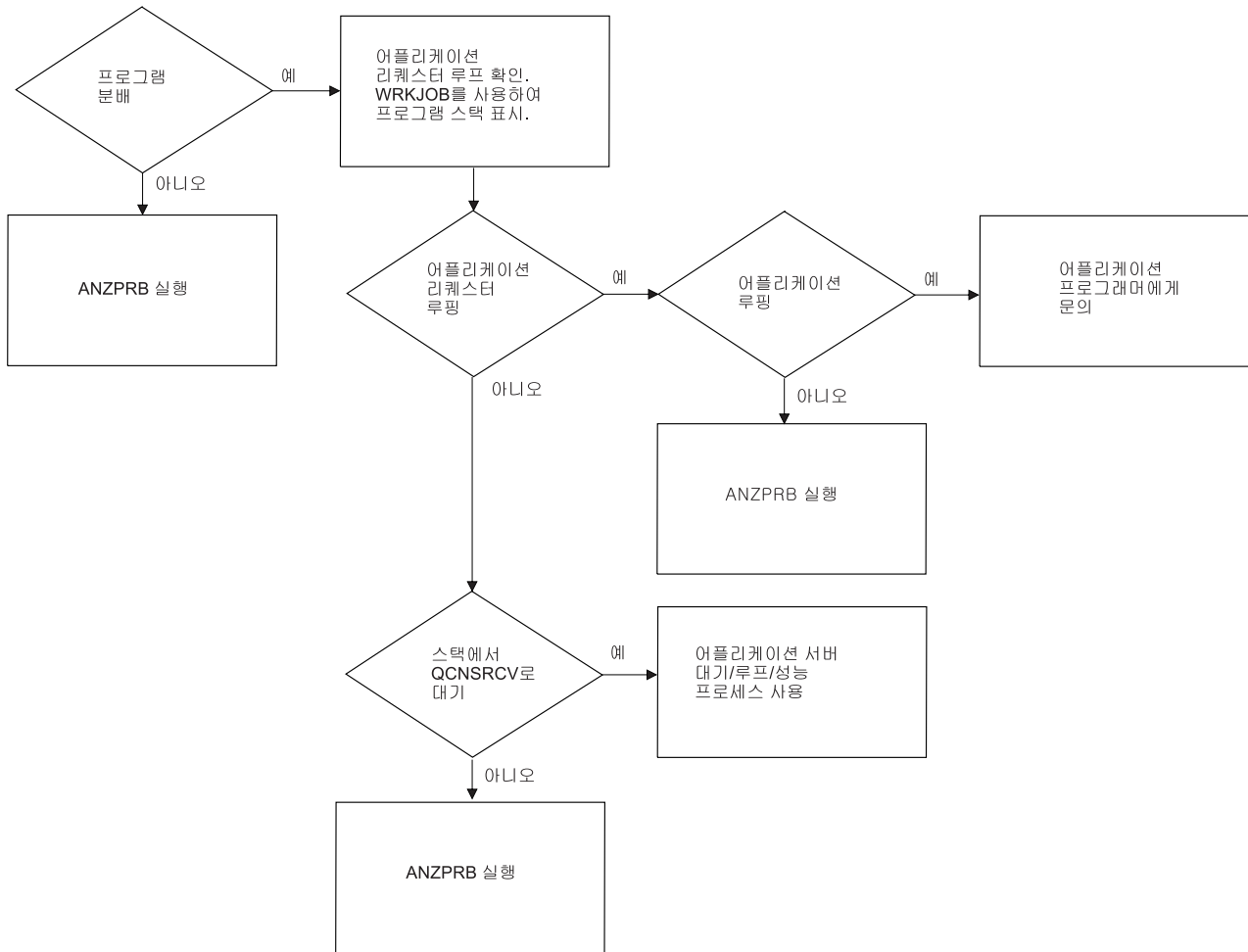
RV2W731-1

그림 14. 부정확한 출력 문제점 해결

어플리케이션이 예상 시간 내에 완료되지 않는 문제

만일 완료 예상 시간보다 요청이 오래 걸릴 경우, 첫 번째 검사는 어플리케이션 리퀘스터(AR)에서 실행됩니다. 관계형 데이터베이스(RDB)에 대한 연결이 완료되었음을 표시하는 메시지 SQL7969에 대한 작업 기록부를 검사하십시오. 이것은 어플리케이션이 분산 관계형 데이터베이스(RDB) 어플리케이션임을 설명합니다. 프로그램 스택을 표시하기 위한 작업에 대한 작업(WRKJOB) 명령을 사용하여 루프에 대한 어플리케이션 리퀘스터를 검사하십시오. 그리고 프로그램 스택을 검사하여 시스템이 루핑하는지 여부를 판별하십시오. 156 페이지의 그림 15를 참조하십시오. 만일 어플리케이션 자체에 루프가 있다면 어플리케이션 프로그래머에게 솔루션을 의뢰하십시오. 만일 스택에서 QAPDEQUE 및 QCNSRCV를 발견한다면 어플리케이션 리퀘스터가 어플리케이션

이전 서버(AS)에 대해 대기 중이라는 뜻입니다. 158 페이지의 그림 16을 참조하십시오. 만일 시스템이 통신 대기 상태에 있지 않다면, 문제점 분석 프로시듀어를 사용하여 성능에 문제점이 있는지 또는 다른 곳이 대기 상태에 있는지 확인하십시오.



RV2W732-2

그림 15. 어플리케이션 리퀘스터 상의 대기, 루프 또는 성능 문제점 해결.

어플리케이션 서버 상의 작업 기록부를 검토하여 어플리케이션 리퀘스터 작업명을 찾을 수 있습니다. 어플리케이션 서버 상의 작업을 찾기 위한 자세한 정보에 대해서는 103 페이지의 『분산 관계형 데이터베이스 작업 찾기』를 참조하십시오. AS 작업을 검사해야 할 경우, 작업에 대한 작업(WRKJOB), 활동 작업에 대한 작업(WRKACTJOB) 또는 사용자 작업에 대한 작업(WRKUSRJOB) 명령을 사용하여 AS에 작업을 배치하십시오. 이 명령 사용에 대한 정보는 다음 주제를 참조하십시오.

- 98 페이지의 『분산 관계형 데이터베이스에서 작업(Job)에 대한 작업』
- 99 페이지의 『분산 관계형 데이터베이스에서 사용자 작업(Job)에 대한 작업』
- 100 페이지의 『분산 관계형 데이터베이스에서 활동 작업(Job)에 대한 작업』

이 작업 표시장치 중 하나를 통하여, 어플리케이션 서버가 루핑하고 있는지 여부를 확인하기 위해 프로그램 스택을 점검하십시오. 루핑이 있다면, 문제점 처리를 위해 문제점 분석을 사용하십시오. 루핑이 없다면,

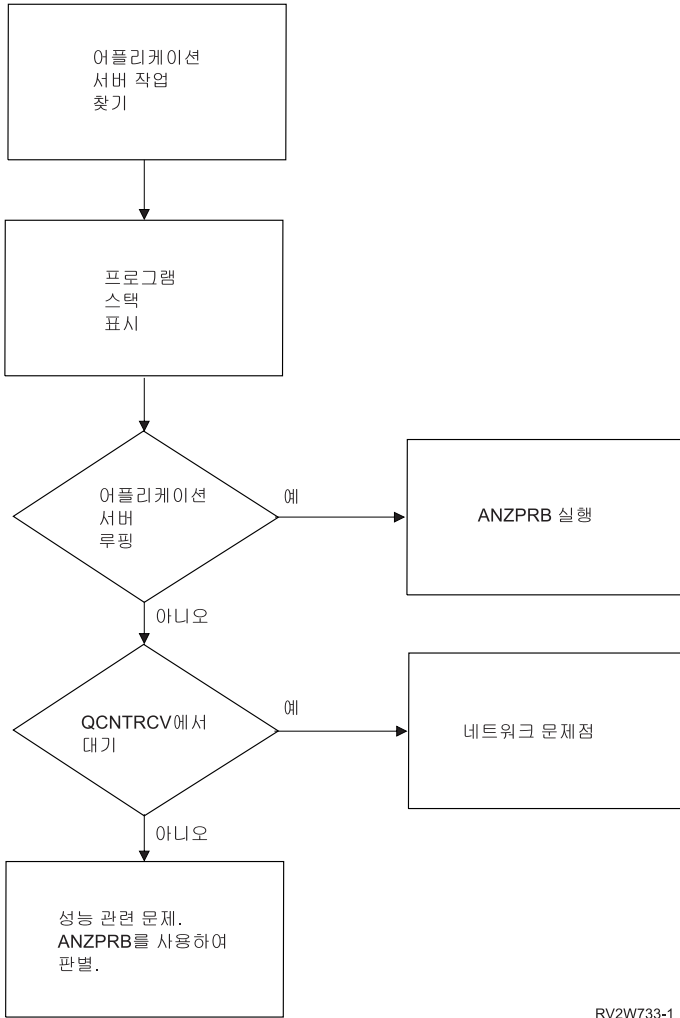
QCNTRCV로 WAIT에 대한 프로그램 스택을 검사하십시오. QCNTRCV는 어플리케이션 서버가 어플리케이션 리퀘스터에 대해 대기 중이라는 뜻입니다. 서버 모두가 이 통신 대기 상태에 있다면, 네트워크 상에 문제점이 있다는 뜻입니다. 만일 어플리케이션 서버가 대기 상태에 있지 않다면, 알려져야 할 성능 제공이 있다는 뜻입니다.

느린 조회 실행의 두 가지 일반 소스는 다음과 같습니다.

- 액세스된 표에는 색인이 없습니다. 만일 이 경우라면, 키로서 적절한 필드 또는 필드를 사용하여 색인을 작성하십시오.
- 조회 요구에 대해 리턴된 행이 블록화 되어 있지 않습니다. 행의 블록화 여부는 조회 성능 상 중대한 차이를 야기합니다. 블록킹에 영향을 주는 요소를 이해하고 어플리케이션의 장점을 최대한 살리도록 조절하는 것이 매우 중요합니다. 자세한 정보는 147 페이지의 『DRDA블록킹에 영향을 주는 요소』를 참조하십시오.

| 만일 iSeries용 DB2 UDB내의 제품에 대한 SQL 패키지를 아직 작성하지 못했다면, DB2 JDBC Universal
| Driver 또는 DB2 연결과 같은 제품을 사용하여 워크스테이션으로부터 iSeries용 DB2 UDB로 처음으로 연결
| 했을 때 패키지는 자동으로 작성됩니다. NULLID 콜렉션 또한 자동적으로 작성되어야 할 경우도 있습니다. 이
| 로 인해 초기 연결 후에 발행된 첫 번째 SQL문 중 하나에 대해 서버에서 응답을 수신할 때 약간의 지연이 발
| 생합니다.

만일 TCP/IP를 통해 연결하려는 서버가 사용 불가할 경우, 상당한 지연이 발생할 것입니다. 몇 분에 걸친 시간 종료(time-out) 지연에 앞서 리모트 호스트가 시간 종료 기간 내에 응답하지 않습니다라는 메시지를 표시합니다. RDB 디렉토리 내의 IP 주소가 부정확할 경우도 이 작동의 원인이 됩니다.



RV2W733-1

그림 16. 어플리케이션 서버 상의 대기, 루프 또는 성능 문제점 해결.

분산 관계형 데이터베이스 사용자에게 대한 작업

문제점에 대한 연구를 처음 시작하는 것은 대부분 사용자입니다. 사용자가 프로그램 실행 시 예상한 결과를 얻지 못하거나 문제점을 지적하는 메시지를 통보받습니다. 때로는 문제점을 진단하고 해결하는 가장 좋은 방법이 사용자와 각 프로시듀어를 단계적으로 실행하는 것입니다. 복사 화면 기능은 사용자와 리얼 타임 작업을 가능하게 하거나 이전에 사용자가 봤던 표시장치의 파일의 검사를 가능케 합니다.

또한 메시지에서부터 화면의이 섹션에서는 다른 사용자가 봤던 표시장치의 복사 방법을 설명하고 분산 관계형 데이터베이스(RDB) 작업을 실행할 때 사용자가 수신하는 메시지에 대한 보다 많은 정보를 얻는 방법을 설명합니다.

프로그래밍 관련 문제 외에도 프로그램 시작이나 서버 연결에 문제가 발생할 수도 있습니다. 이 문제를 처리하는 방법에 관한 세부사항은 다음 주제를 참조하십시오.

- APPC에 대한 프로그램 시작 요구 실패 처리
- TCP/IP에 대한 연결 요구 실패 처리

화면 복사

STRCPYSCN(화면 복사 시작) 명령으로 워크스테이션을 시작하고 다른 워크스테이션에서 타인이 봤던 똑같은 표시 장치를 볼 수 있습니다. 사용자와 똑같은 iSeries 서버를 시작해야만 합니다. 만일 리모트 서버에서 실행된 것이라면, 그 서버를 사인 온하기 위해 표시장치 pass-through를 사용할 수 있습니다. 그 후에 STRCPYSCN 명령을 입력하여 기타 표시장치를 볼 수 있습니다. 화면 이미지는 그 이미지가 다른 워크스테이션에 복사된 때 또는 다른 워크스테이션이 사용 불가할 때와 동시에 데이터베이스 파일에 복사될 수 있습니다. 이 방식으로 이후에 해당 자료를 처리할 수 있고 문제점 상황 중 발생한 작업에 대한 감사 추적을 준비할 수 있습니다.

이미지 표시를 다른 표시장치에 복사하려면 다음 요구사항이 충족되어야 합니다.

- 표시장치가 모두 서버에 정의되어 있습니다.
- 표시장치가 모두 칼라이거나 모두 단색입니다. 하지만 하나만 칼라이고 하나만 단색인 경우는 아닙니다.
- 표시장치가 모두 수평, 수직으로 문자 위치에 대해 똑같은 번호를 가집니다.

만일 송신 장치로서 자체의 표시장치 ID로서 입력한다면, 수신 표시장치에는 화면 이미지의 복사를 시작할 때 보여진 사인 온 화면이 있어야 합니다. 그래픽은 공백처리되어 복사됩니다.

만일 똑같은 서버에서 사인 온하지 못하는 경우, 다음 프로세스를 사용하여 다른 사용자가 리모트 서버 상에서 보는 표시장치를 볼 수 있습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

1. Pass-Through 시작(STRPASTHR) 명령을 입력하십시오.

```
STRPASTHR RMTLOCNAME(KC105)
```

2. application source(AS)에 로그인하십시오.

3. (STRCPYSCN) 명령을 입력하십시오.

```
STRCPYSCN SRCDEV(KC105)
          OUTDEV(*REQUESTER)
          OUTFILE(KCHELP/TEST)
```

- SRCDEV는 소스 장치, 즉 이미지 표시를 송신하는 표시장치의 이름을 지정합니다. 다른 장치에 대한 명령으로 본인의 화면에 송신하려면 이 매개변수에 대해 *REQUESTER 값을 입력하십시오.
- OUTDEV는 이미지 표시가 송신될 출력 장치명을 지정합니다. 이 예에서, 이미지 표시는 명령(*REQUESTER)을 입력한 사용자의 표시장치로 송신됩니다. 다른 표시장치, 다른 장치(제 3자가 보고 있는)를 명명하거나 장치를 아예 명명하지 않을 수도(*NONE) 있습니다. *NONE 값을 사용할 때 이미지 표시에 대한 출력 파일을 지정하십시오.
- OUTFILE은 명령이 활동 중일 때 보여진 모든 표시장치의 이미지가 들어있는 출력 파일명을 지정합니다.

4. 표시장치가 다른 장치 또는 파일에 복사될 것임을 장치 사용자에게 통지하기 위해 소스 장치에 조회 메시지가 송신됩니다. g(Go)를 입력하여 요구 장치에 대한 이미지 송신을 시작하십시오.

송신 표시장치의 화면이 다른 표시장치에 복사됩니다. 수신 표시장치에 보여진 이미지는 화면별로 송신 표시장치를 추적합니다. 만일 송신 표시장치를 사용하는 사용자가 활동 중이 아닌 키를(HOME 키와 같은) 누른다면, 표시장치 모두가 똑같은 화면을 보일 것입니다.

화면을 복사하는 동안, 수신 표시장치의 오퍼레이터는 화면 복사가 끝날 때까지 그 표시장치에 대해 어떠한 작업도 할 수 없습니다.

송신 표시장치로부터 화면 복사 기능을 종료하려면, 임의의 명령 행에서 ENDCPYSCN(화면 복사 종료) 명령을 입력하고 Enter를 누르십시오.

ENDCPYSCN

화면 복사 기능을 시작할 때 봤던 화면이 보일 것입니다.

메세지

iSeries 서버에서는 간단한 입력 오류에서 서버 장치 또는 프로그램에 대한 문제점을 망라한 조건을 표시하는 시스템 메시지를 다양하게 송신할 수 있습니다. 메시지는 다음 중 하나입니다.

- 현재 화면 상의 오류 메시지

이 메시지는 작업을 인터럽트시키거나 경보를 발생시킬 수 있습니다. 임의의 명령 행에 DSPMSG을 입력하면 이 메시지를 표시할 수 있습니다.

- 서버 오퍼레이터 메시지 대기행렬에 송신되었거나 분리된 메시지에 대한 작업 화면 상 표시된 서버 문제점에 관련된 메시지.

이 메시지를 보기 위해서는 임의의 서버 명령 행에 DSPMSG QSYSOPR을 입력하십시오.

- 장치 설명에 명시된 메시지 대기행렬에 송신된 서버 문제점에 관련된 메시지.

이 메시지를 보기 위해서는 임의의 서버 명령 행에 DSPMSG message-queue-name을 입력하십시오.

서버가 특정 서버 이벤트에 대한 정보 또는 조회 메시지를 송신합니다. 정보용 메시지는 서버의 사용 상태를 알려 줍니다. 조회 메시지는 서버에 대한 정보를 알려 주지만 또한 그에 대한 응답을 요청합니다.

몇 가지 메시지 표시장치 내에서 메시지는 다음과 같은 문자 및 번호 코드를 동반합니다.

CPF0083

첫 두 개 또는 세 개의 문자는 메시지 분류를 표시합니다. 분산 관계형 데이터베이스(DRDB)에 대한 몇 가지 메시지 분류는 다음과 같습니다.

표 8. 메시지 분류

분류	설명	라이브러리
CPA에서 CPZ까지	오퍼레이팅 시스템발 메시지	QSYS/QCPFMSG
MCH	사용권 내부 코드 메시지	QSYS/QCPFMSG

표 8. 메세지 분류 (계속)

분류	설명	라이브러리
SQ 및 SQL	SQL(Structured Query Language) 메세지	QSYS/QSQLMSG
TCP	TCP/IP 메세지	QTCP/QTCPMSGF

남아 있는 네 자리수(접두부가 SQ일 경우 다섯 자리수)는 메세지의 순번을 표시합니다. 보여진 메세지 ID 예에서는 오퍼레이팅 시스템에서 메세지 0083번임을 표시합니다.

화면 또는 메세지 대기행렬의 메세지 행에 대한 메세지 관련 정보를 얻기 위해서는 다음 단계를 따르십시오.

1. 커서를 메세지가 있는 같은 행으로 옮기십시오.
2. 도움말 키를 누르십시오. 추가 메세지 정보화면이 나타납니다.

추가 메세지 정보

메세지 ID	CPD6A64	심각도	30
메세지 유형	진단	송신 시간	13:49:06
송신 날짜	03/29/92	지시	0800
송신 프로그램	QUIACT	지시	03C5
발신 프로그램	QUIMGFLW		

메세지 : 지정된 메뉴 선택이 올바르지 않습니다.
 원인 : 지정한 선택이 올바르지 않은 이유는
 다음 중 하나입니다.
 -- 선택된 번호가 유효하지 않습니다.
 -- 메뉴 옵션이 아닌 다른 것이 옵션 행에 입력되었습니다.
 회복 : 유효 옵션을 선택하고 Enter를 누르거나 도움말 키를 다시
 누르십시오.

맨 아래

계속하려면 Enter를 누르십시오.

F3=종료 F6=인쇄 F9=화면 메세지 세부사항
 F10=작업 기록부 내의 메세지 화면 F12=취소 F21=지원 레벨 선택

만일 메세지 ID 및 위치한 라이브러리를 알고 있다면 화면에서 볼 수 없는 메세지에 대해 더 많은 정보를 얻을 수 있습니다. 이 정보를 얻기 위해서는 메세지 설명 표시(DSPMSG) 명령을 입력하십시오.

DSPMSGD RANGE(SQL0204) MSGF(QSYS/QSQLMSG)

이 명령은 메세지에 대한 다음의 정보를 선택할 수 있도록 허용하는 화면을 생성합니다.

- 메세지 텍스트
- 필드 자료
- 메세지 속성
- 모두

텍스트는 추가 메세지 정보 화면에서 보는 메세지 및 메세지 도움말 텍스트와 똑같습니다. 필드 자료는 메세지 및 그 속성에 대해 정의된 모든 대체 변수 리스트입니다. 메세지 속성이 심각도, 로깅, 메세지 레벨, 디폴트 프로그래밍 및 덤프 매개변수에 대한 값(정의되었을 때)입니다. 이 정보를 사용하여 메세지가 나타날 때 사용자의 작업을 판별할 수 있습니다.

메세지 유형

추가 메세지 정보 화면에서 메세지에 대한 메세지 유형 및 심각도 코드를 볼 수 있습니다. 표 9에서는 iSeries 메세지 및 관련 심각도 코드에 대한 각 다른 메세지 유형을 나타내고 있습니다.

표 9. 메세지 심각도 코드

메세지 유형	심각도 코드
정보용 메세지. 정보용 목적 전용으로서라면 응답이 요구되지 않습니다. 메세지는 기능이 진행 중인지 또는 기능이 성공적으로 완료되었는 지를 표시합니다.	00
경고. 잠재 오류 조건이 존재합니다. 프로그램에서는 잃어버린 자료를 공급하는 등의 디폴트 작업을 취할 수 있습니다. 작업의 결과는 성공적인 것으로 전제합니다.	10
오류. 오류가 발견되었습니다. 하지만 자동 회복 프로시더어가 적용되었습니다. 처리가 계속되었습니다. 디폴트 작업이 틀린 자료를 대체하기 위해 발생하였습니다. 작업의 결과가 옳지 못할 수도 있습니다. 기능이 완료되지 못했을 수도 있습니다. 예를 들어, 리스트 내의 몇 가지 항목은 올바르게 실행되지만 몇 가지는 그렇지 못합니다.	20
심각한 오류. 발견된 오류가 자동 회복 프로시더어로 감당하기에 너무 심각합니다. 디폴트 작업이 불가능합니다. 만일 오류가 소스 자료 내에 발생한 것이라면 전체 자료 레코드를 생략합니다. 만일 프로그램 내에 발생한 것이라면 프로그램의 이상 종료를 초래할 것입니다(심각도40). 작업의 결과가 옳지 못할 수도 있습니다.	30
심각한 오류: 프로그램 또는 기능의 이상 종료. 프로그램이 옳지 못한 자료를 처리할 수 없기 때문에 또는 사용자가 처리를 취소했기 때문에 작업이 종료되었습니다.	40
작업 또는 프로그램의 이상 종료. 작업이 시작되지 않았거나 시작이 실패했습니다. 작업 레벨 기능이 필요한 만큼 완수되지 못했거나 작업이 취소되었습니다.	50
시스템 상태. 시스템 오퍼레이터 메세지 대기행렬에 대해서만 발행되었습니다. 장치, 서브시스템 또는 시스템에 대한 상태 또는 경고를 제공합니다.	60
장치 무결성. 장치가 올바르게 작동하지 않거나 더이상 장치가 작동가능하지 않다는 사실을 표시하여 시스템 오퍼레이터 메세지 대기행렬에만 발행되었습니다.	70
시스템 경고 및 사용자 메세지. 비록 현재 시스템을 중단할 만큼 심각하지는 않지만 예방 측정이 취해지지 않는다면 조건이 더욱 심각해질 수 있습니다.	80
시스템 무결성. 시스템 오퍼레이터 메세지 대기행렬에 대해서만 발행되었습니다. 서브시스템 또는 시스템이 조작할 수 없는 조건을 설명합니다.	90
조치. 응답을 입력하거나 프린터 양식을 변경하는 등의 몇 가지 수동 조치가 요구됩니다.	99

분산 관계형 데이터베이스(DRDB) 메시지

만일 어플리케이션 서버(AS) 또는 어플리케이션 리퀘스터(AR)에서 오류 메시지가 발생한 경우, 서버 메시지는 실패에 대한 원인을 표시하기 위해 작업 기록부에 로그 온됩니다. 작업 기록부 사용 방법 및 어플리케이션 서버 상에 작업 기록부 위치지정 방법에 대한 내용은 102 페이지의 『분산 관계형 데이터베이스의 작업 기록부를 사용한 요구 정보 트래킹』을 참조하십시오.

서버 메시지가 iSeries용 DB2 Universal Database 프로그램에 의해 지원받는 SQL문으로부터 리턴된 각 SQLCODE에 대해 존재합니다. 메시지는 디버그 모드에서 실행될 때, 사전컴파일러 리스팅, 대화식 SQL 또는 작업 기록부 내에서 작성이 가능합니다. 하지만 iSeries 서버가 아닌 어플리케이션 서버에서 작업할 때라면, 다음 경우와 같은 모든 오류 조건에 대한 특정 메시지가 없을 수도 있습니다.

- iSeries 서버에 의해 사용되지 않는 함수와 관련된 오류입니다.

예를 들어, 특수 레지스터 CURRENT SQLID는 iSeries용 DB2 UDB에서 지원되지 않습니다. 따라서 SQLCODE -411(SQLSTATE 56040) 『CURRENT SQLID 는 리모트 오브젝트를 참조하는 명령문 내에서 사용될 수 없습니다』란 존재하지 않습니다.

- 제품에서 기인한 오류이며 iSeries용 DB2 UDB을 사용하는 중에 결코 발생하지 않습니다.

iSeries용 DB2 UDB에는 결코 SQLCODE -925(SQLSTATE 56021), 『SQL 확약 또는 롤백이 IMS™ 또는 CICS® 환경 내에서 유효하지 않습니다』가 없습니다.

상응하는 메시지가 없는 SQLCODE의 경우, 메시지를 생성한 어플리케이션 서버의 관계형 데이터베이스(RDB) 명과 더불어 인지되지 못한 SQLCODE, SQLSTATE 및 토큰을 식별하는 총칭 메시지를 리턴합니다. 특정 조건과 토큰 이해 방법을 판별하기 위해서, 연결된 어플리케이션 서버의 특정 릴리스에 상응하는 제품 문서를 참조하십시오. SQLCODE에 대한 자세한 정보는 171 페이지의 『SQLCODE 및 SQLSTATE』를 참조하십시오.

CPx3E00에서 CPx3EFF까지 및 CPI9100에서 CPI91FF까지에 이르는 범위의 메시지는 분산 관계형 데이터베이스(RDB) 서버 메시지로 사용됩니다. 다음 리스트는 포괄적이지는 않으나 iSeries 서버 상의 분산 데이터베이스 작업 기록부 내에서 볼 수 있는 일반적인 서버 메시지를 더 많이 나타내고 있습니다. 분산 관계형 데이터베이스(RDB)에 대한 SQL 메시지 리스트는 SQL 프로그래밍 개념 책을 참조하십시오.

표 10. 분산 관계형 데이터베이스(DRDB) 메시지

MSG ID	설명
CPA3E01	*LOCAL RDB 디렉토리 항목의 삭제 시도
CPC3EC5	무시된 RDB 디렉토리 항목에 대한 몇 가지 매개변수
CPD3E30	상충되는 지정 리모트 네트워크 ID
CPD3E35	유효하지 않은 리모트 위치명 구조
CPD3E36	포트 식별이 유효하지 않습니다.
CPD3E38	리모트 위치에 대한 유형 충돌
CPD3E39	허용되지 않은 매개변수&2에 대한 값 &2
CPD3E3B	서버 권한 정보를 조회하던 중 오류가 발생했습니다.
CPD3ECA	RDB 디렉토리 작업이 완료되지 않았습니다.
CPD3E01	지원되지 않는 DBCS 또는 MBCS 코드화 문자 세트 ID(CCSID)

표 10. 분산 관계형 데이터베이스(DRDB) 메시지 (계속)

MSG ID	설명
CPD3E03	RDB 디렉토리에 없는 로컬 RDB명
CPD3E05	DDM 대화식 경로를 찾을 수 없습니다.
CPD3E31	DDM TCP/IP 서버가 사용 중이 아닙니다.
CPD3E32	DDM TCP/IP 서버를 종료하면서 오류가 발생했습니다.
CPD3E33	이유 코드...의 DDM TCP/IP 서버 오류가 발생했습니다.
CPD3E34	DDM TCP/IP 서버 통신 오류가 발생했습니다.
CPD3E37	DDM TCP/IP가 잘못된 이름의 호스트와 연결되었습니다.
CPF3E30	DDM TCP/IP 서버 시작 중 오류가 발생했습니다.
CPF3E31	DDM TCP/IP 서버를 시작할 수 없습니다.
CPF3EC6	실패한 DDM TCP/IP 속성을 변경하십시오.
CPF3EC9	RDB 인터럽트에 대한 범위 메시지
CPF3E0A	자원 한계 오류
CPF3E0B	조회가 열리지 않았습니다.
CPF3E0C	FDOCA LID 한계입니다.
CPF3E0D	인터럽트가 불가합니다.
CPF3E01	지원되지 않는 DDM 매개변수 값입니다.
CPF3E02	어플리케이션 리퀘스터가 작업을 지원하지 않습니다.
CPF3E04	SBCS 코드화 문자 세트 ID(CCSID)가 지원되지 않습니다.
CPF3E05	패키지 바인딩이 활동하지 않습니다.
CPF3E06	RDB를 찾을 수 없습니다.
CPF3E07	패키지 바인딩 프로세스가 활동 중입니다.
CPF3E08	조회 열기 실패
CPF3E09	바인드 시작 오류
CPF3E10	어플리케이션 서버가 DBCS 또는 MC를 지원하지 않습니다.
CPF3E12	확약/롤백 HOLD가 지원되지 않습니다.
CPF3E13	확약 제어 작업이 실패했습니다.
CPF3E14	RDB 요구 종료 실패했습니다.
CPF3E16	RDB에 대한 권한이 없습니다.
CPF3E17	RDB 종료 요구가 진행 중입니다.
CPF3E18	SQLCA를 사용한 COMMIT/ROLLBACK
CPF3E19	확약 제어 작업이 실패했습니다.
CPF3E20	DDM 대화식 경로를 찾을 수 없습니다.
CPF3E21	RDB 인터럽트가 실패했습니다.
CPF3E22	확약이 어플리케이션 서버에서 롤백을 초래했습니다.
CPF3E23	DDM 자료 스트림이 대화식 기능을 손상시켰습니다.
CPF3E30	DDM TCP/IP 서버 시작 중 오류가 발생했습니다.
CPF3E32	클라이언트 요구 처리 중 서버 오류가 발생했습니다.
CPF3E80	자료 스트림 구문 오류
CPF3E81	유효하지 않은 FDOCA 설명자
CPF3E82	ACCRDB가 두 번 전송되었습니다.

표 10. 분산 관계형 데이터베이스(DRDB) 메시지 (계속)


MSG ID	설명
CPF3E83	자료 불일치 오류
CPF3E84	DDM 대화식 프로토콜 오류
CPF3E85	RDB에 액세스하지 않았습니다.
CPF3E86	예상치 못한 상태입니다.
CPF3E87	영구 에이전트(agent) 오류
CPF3E88	조회가 이미 열렸습니다.
CPF3E89	조회가 열리지 않았습니다.
CPF3E99	RDB 요구 종료가 발생했습니다.
CPI9150	DDM 작업이 시작되었습니다.
CPI9152	목표 DDM 작업이 어플리케이션 리퀘스터(AR)에 의해 시작되었습니다.
CPI9160	DDM 연결이 TCP/IP를 통해 시작되었습니다.
CPI9161	DDM TCP/IP 연결이 종료되었습니다.
CPI9162	DDM 연결을 처리하기 위해 할당된 목표 작업이 시작되었습니다.
CPI9190	분배 데이터베이스 상의 권한 실패
CPI3E01	로컬 RDB로 성공적으로 액세스했습니다.
CPI3E02	로컬 RDB에서 성공적으로 단절되었습니다.
CPI3E04	관계형 데이터베이스(RDB) &1 연결이 종료되었습니다.
CPI3E30	DDM TCP/IP 서버가 이미 활동 중입니다.
CPI3E31	DDM TCP/IP 서버가 보안 체계를 지원하지 않습니다.
CPI3E32	DDM 서버가 성공적으로 시작되었습니다.
CPI3E33	DDM 서버가 성공적으로 종료되었습니다.
CPI3E34	mm/dd/yy, hh:mm:ss에 yyy 사용자에게 서비스하는 DDM 작업 xxxxQRWOPTIONS로 축약될 수 있습니다.)
CPI3E35	DDM 서버 사전시작 작업 항목이 없습니다.
CPI3E36	관계형 데이터베이스(RDB)xxxx 연결이 종료되었습니다.
SQ30082	연결 시도가 실패했습니다. 이유 코드는 ...입니다.
SQL7992	TCP/IP 상 연결이 완료되었습니다.
SQL7993	이미 연결되었습니다.

APPC에 대한 프로그램 시작 요구 실패 처리

프로그램 시작 요구를 **application server(AS)** 상의 OS/400 서브시스템에서 수신할 때, 서버에서는 프로그램 시작 요구와 송신된 정보에 기반한 작업의 시작을 시도합니다. 어플리케이션 서버(AS)에 대한 어플리케이션 리퀘스터(AR) 사용자의 권한, 요구된 데이터베이스 및 기타 많은 항목이 검사됩니다.

만일 어플리케이션 서버 서브시스템에서 작업을 시작할 수 없다고 판별할 경우(예를 들어, 사용자 프로파일이 어플리케이션 서버에 없거나, 사용자 프로파일이 있으나 사용이 불가하고 또는 사용자에게 어플리케이션 서버 상에 요구된 오브젝트에 대한 정당한 권한이 없을 경우), 서브시스템에서는 메시지 CPF1269를 QSYSMSG 메시지 대기행렬에 송신합니다(QSYSMSG이 존재하지 않을 경우, QSYSOPR을 송신합니다). CPF1269 메시지에는 두 가지의 이유 코드가 있습니다(이유 코드 중 하나는 제로이며, 이 코드는 무시할 수 있습니다).

제료가 아닌 이유 코드는 프로그램 시작 요구가 거부된 이유를 설명합니다. 리모트 작업이 어플리케이션 서버 상에서 시작되어야 하기 때문에 메시지 및 이유 코드는 어플리케이션 서버(AS) 상에 제공되며 어플리케이션 리퀘스터(AR)에 제공되지 않습니다. 어플리케이션 리퀘스터 상의 사용자는 프로그램 시작 요구가 실패했다는 사실만을 알 뿐 그 이유를 알 수 없습니다. 어플리케이션 리퀘스터 상의 사용자는 어플리케이션 서버의 시스템 오퍼레이터에게 문의하거나 어플리케이션 서버에 대한 표시장치 pass-through를 사용하여 요구가 실패한 원인을 판별해야 합니다.

이유 코드 및 그 의미에 대한 완전한 설명에 대해서는 *ICF Programming*  서적을 참조하십시오.

TCP/IP에 대한 연결 요구 실패 처리

TCP/IP용으로 구성된 분산 관계형 데이터베이스(DRDA) 서버에 대한 연결 요구 실패의 기본 원인은 DDM TCP/IP 서버가 시작되지 않았거나, 권한 오류가 발생했거나 기계가 실행되지 않는 경우입니다.

서버가 시작되지 않았습니까 또는 포트 ID가 유효하지 않습니다

DDM TCP/IP 서버가 시작되지 않은 경우, 주어지는 오류 메시지는 CPE3425입니다.

리모트 호스트가 연결 조작을 거부했습니다.

관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 또는 관계형 데이터베이스 디렉토리 항목 변경(CHGRDBDIRE) 명령에서 잘못된 포트를 지정한 경우도 iSeries용 DB2 UDB 서버의 경우, 포트는 대개 *DRDA(DRDA 잘알려진 446포트)입니다. 하지만 만일 IPSec를 사용하기 위해 포트 447로 구성했다면, 깨지기 쉬운 자료를 전송을 위해 그 포트를 사용할 수도 있습니다. 만일 SSL(Secure Sockets Layer)을 지원하는 DRDA 클라이언트를 사용하려면, 반드시 포트 448로 서버 상에 연결해야만 합니다.

리모트 서버에서 DDM 서버를 시작하려면, TCP/IP 서버 시작(STRTCPSPVR) *DDM 명령을 실행하십시오. DDM TCP/IP 속성 변경(CHGDDMTCPA) AUTOSTART(*YES) 명령을 실행하여 TCP/IP가 시작될 때마다 DDM 서버가 시작되도록 요구할 수 있습니다.

분산 관계형 데이터베이스(DRDA) 연결 권한 실패

권한 실패에 대해 주어진 오류 메시지는 SQ30082입니다.

분배 데이터베이스 연결 시도 상 권한 실패

메시지의 원인 섹션에서는 이유 코드 및 가능한 이유 코드에 대한 의미 리스트를 나타내고 있습니다. 이유 코드 17은 지원되지 않는 보안 체계가 있다는 의미입니다(SECMC).

iSeries용 DB2 UDB는 iSeries 어플리케이션 리퀘스터(AR)가 사용할 수 있는 몇몇 DRDA SEMEC를 구현합니다.

- 사용자 ID 전용
- 암호가 있는 사용자 ID
- 암호화된 암호 보안 메카니즘(V4R5 이후)
- Kerberos(V5R2)

암호화된 암호는 연결이 개시될 때 암호가 사용될 수 있는 경우에만 송신됩니다.

iSeries 서버용 디폴트 SECMEC는 사용자 ID와 암호를 요구합니다. 어플리케이션 리퀘스터(AR)가 암호없이 사용자 ID를 디폴트 보안 구성이 있는 서버에 송신하는 경우, 이유 코드 17인 오류 메시지 SQ30082를 받게 됩니다.

지원되지 않은 SECMEC 실패에 대한 솔루션은 다음과 같습니다.

- DDM TCP/IP 속성 변경(CHGDDMTCPA) 명령 PWDRQD(*NO) 명령을 실행함으로써 서버에서 사용자 ID 전용 SEMEC 허용
- PWDRQD(*YES)가 서버에서 실행 중인 경우, 연결 요구에 대해 최소한 지우기 텍스트 암호 전송
- PWDRQD(*ENCRYPTED)가 서버에서 실행 중인 경우, 암호화된 암호 전송
- RWDRQD(*KERBEROS)가 서버에서 실행 중인 경우, 클라이언트에서 Kerberos 사용

연결 시도가 실행된 경우, 해당하는 사용자 프로파일에 대한 서버 권한 부여 항목에 리모트 사용자 ID 및 암호를 추가하는 데 SQL CONNECT 명령문의 USER/USING 양식을 사용하거나 서버 권한 부여 항목 추가 (ADDSVRAUTE) 명령을 사용함으로써 암호를 송신할 수 있습니다. V4R5 이후 시스템에서는 암호화된 암호를 송신하려는 시도가 자동으로 실행됩니다. V4R5 이전의 iSeries 서버는 암호화된 암호를 송신할 수 없고 V4R5 iSeries AR에 의해 송신된 암호화된 암호 유형을 암호해독할 수도 없음을 주의하십시오.

반드시 시스템 값 QRETSVRSEC(서버 보안 자료를 유지)를 '1'로 설정하여 서버 권한 항목 내에 리모트 암호를 저장할 수 있도록 해야합니다.

경고: 반드시 DRDA 사용에 대해서는 상위 경우 내에 서버 인증 항목 추가 (ADDSVRAUTE) 명령에 RDB 명을 입력해야만 합니다. 그렇지 않으면 연결 처리 중 이름이 인식되지 않고 권한 항목 내의 정보가 사용되지 않습니다.

사용할 수 없는 서버

만일 리모트 서버가 실행되지 않은 경우, 또는 application source(AS)에 대한 RDB 디렉토리 항목에 잘못된 IP 주소를 지정한 경우, 메시지 CPE3447을 받게 됩니다.

리모트 호스트가 시간 종료 내에 응답하지 않았습니다.

이 메시지가 발생하기 전에 일반적으로 몇 분간의 지연이 있습니다. 지연 사유는 발생 시간 당시 무엇인가 걸려있거나 루핑 중이기 때문입니다.

대화식 SQL 관련 연결 실패

대화식 SQL로부터 CONNECT 명령문을 실행할 때 이따금 일반적인 SQ30080 메시지, 분배 데이터베이스 처리 중 통신 오류가 발생했습니다가 보여집니다. 이 오류의 세부사항을 알기 위해서, 반드시 대화식 SQL을 종료하고 작업 기록부를 검사해야 합니다.

만일 처음 연결에서(확약 제어에 대한 임의의 주어진 레벨에 대해) 메시지 SQL7020, SQL 패키지 생성이 실패했습니다가 수신되었다면, 추정 원인은 읽기 전용 서버인 리모트 서버에 액세스했기 때문이거나 SQL 패키지를 생성하기 위해 서버를 갱신할 필요가 생겼기 때문입니다.

작업 기록부 내의 메시지를 검토하여 검증할 수 있습니다. 솔루션은 연결 전에 모든 연결을 제거하기 위해 RELEASE ALL 또는 COMMIT을 실행하는 것입니다. 그러면 연결 갱신이 가능해집니다. 85 페이지의 『대화식 SQL(ISQL)용 SQL 패키지 설정』을 참조하십시오.

불충분한 서버 사전시작 작업

만일 TCP/IP와 관련된 사전시작 작업 수가 QSYSWRK 서브시스템의 QRWTSSRVR 사전시작 작업 항목에 의해 제한될 경우, 그리고 모든 사전시작 작업이 연결에 대해 사용된 경우, 신규 연결을 위한 시도는 실패하고 다음 메시지를 보냅니다.

CPE3426

소켓에 의해 재설정된 리모트 소켓과 연결

CPD3E34

DDM TCP/IP 통신 오류가 recv() -- MSG_PEEK에 발생했음.

QTWTSSRVR 항목에 대한 사전시작 작업 항목 변경(CHGPJE) 명령의 MAXJOBS 매개변수를 더 상위 번호 또는 *NOMAX에 설정하고 ADLJOBS 매개변수를 0이 아닌 다른 값으로 설정하면 서버에서 이 문제점의 발생을 방지할 수 있습니다.

어플리케이션 문제점

어플리케이션 문제점을 처리하는 최적기는 문제점이 발생하기 전에 처리하는 것입니다. 하지만 일반 사용 시 어플리케이션에 대해 존재할 모든 조건에 대해 일일이 예상한다는 것은 불가능합니다. 어플리케이션 리퀘스터 (AR) 또는 **application server (AS)**의 작업 기록부에서 패키지의 실패를 알 수 있습니다. 프로그램 또는 패키지의 리스트에서 실패 원인을 SQL 컴파일러에서는 진단 리스트 상의 사전컴파일 프로세스에 의해 생성된 SQLCODEs 및 SQLSTATEs를 나타내는 진단 테스트를 제공합니다.

통합 언어 환경(ILE*) 사전컴파일의 경우, 선택적으로 OPTION(*XREF) 및 OUTPUT(*PRINT)을 지정하여 사전컴파일 소스 및 상호 참조 리스팅을 인쇄할 수 있습니다. ILE이 아닌 사전컴파일의 경우, 선택적으로 *SOURCE 및 *XREF를 CRTSQLxxx(SQL 프로그램 작성) 명령의 OPTION 매개변수에 지정하여 소스 및 상호 참조 리스팅을 인쇄할 수 있습니다.

리스팅

169 페이지의 그림 17 내의 CRTSQLxxx(SQL 프로그램 작성) 명령에서 비롯한 리스팅은 다음과 같은 종류의 정보를 제공합니다.

- 사전컴파일 명령의 매개변수에 대해 제공된 값
- 프로그램 소스
- ID 중복 참조
- 사전컴파일로부터 비롯한 메시지

사전컴파일러 리스트

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

5763ST1 V3R1M0 940909          Create SQL ILE C Object          UPDATEPGM          04/19/94 14:30:10  Page  1
Source type.....C
Object name.....TST/UPDATEPGM
Source file.....*LIBL/QCSRC
Member.....*OBJ
Options.....*XREF
Listing option.....*PRINT
Target release.....*CURRENT
INCLUDE file.....*LIBL/*SRCFILE
Commit.....*CHG
Allow copy of data.....*YES
Close SQL cursor.....*ENDACTGRP
Allow blocking.....*READ
Delay PREPARE.....*NO
Generation level.....10
Margins.....*SRCFILE
Printer file.....*LIBL/QSYSVRT
Date format.....*JOB
Date separator.....*JOB
Time format.....*HMS
Time separator .....*JOB
Replace.....*YES
Relational database.....RCHASLKM
User .....*CURRENT
RDB connect method.....*DUW
Default Collection.....*NONE
Package name.....*OBJLIB/*OBJ
Created object type.....*PGM
Debugging view.....*NONE
Dynamic User Profile.....*USER
Sort Sequence.....*JOB
Language ID.....*JOB
IBM SQL flagging.....*NOFLAG
ANS flagging.....*NONE
Text.....*SRCMBRTXT
Source file CCSID.....37
Job CCSID.....65535
Source member changed on 04/19/94 14:25:33

```

```

5763ST1 V3R1M0 940909          Create SQL ILE C Object          UPDATEPGM          04/19/94 14:30:10  Page  2
Record  *...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8  SEQNBR  Last change
1  /*****/ 100
2  /* This program is called to update the DEPTCODE of file RWDS/DPT1 */ 200
3  /* to NULL. This is run once a month to clear out the old */ 300
4  /* data. */ 400
5  /* */ 500
6  /* NOTE: Because this program was compiled with an RDB name, it is */ 600
7  /* not necessary to do a connect, as an implicit connect will take */ 700
8  /* place when the program is called. */ 800
9  /*****/ 900
10 #include <stdio.h> 1000
11 #include <stdlib.h> 1100
12 exec sql include sqlca; 1200
13 1300
14 main() 1400
15 { 1500
16     /* Just update RWDS/DPT1, setting deptcode = NULL */ 1600
17     exec sql update RWDS/DPT1 1700
18         set deptcode = NULL; 1800
19 } 1900
* * * * * E N D O F S O U R C E * * * * *

```

그림 17. 사전컴파일러로부터의 리스트 (1/2)

```

5763ST1 V3R1M0 940909          Create SQL ILE C Object          UPDATEPGM          04/19/94 14:30:10  Page    3
CROSS REFERENCE
Data Names          Define    Reference
DEPTCODE           ****    COLUMN
18
DPT1                ****    TABLE IN RWDS
17
RWDS                ****    COLLECTION
17
5763ST1 V3R1M0 940909          Create SQL ILE C Object          UPDATEPGM          04/19/94 14:30:10  Page    4
DIAGNOSTIC MESSAGES
MSG ID SEV  RECORD  TEXT
SQL0088  0      17  Position 15 UPDATE applies to entire table.
SQL1103 10      17  Field definitions for file DPT1 in RWDS not found.
Message Summary
Total  Info  Warning  Error  Severe  Terminal
2      1      1         0       0         0
10 level severity errors found in source
19 Source records processed
***** END OF LISTING *****

```

그림 17. 사전컴파일러로부터의 리스트 (2/2)

CRTSQLPKG 리스팅

그림 18 내의 구조화 조회 언어 패키지 작성(CRTSQLPKG) 명령 명령에서 비롯한 리스팅은 두 가지 유형의 정보를 제공합니다.

- 명령 매개변수에 사용된 값
- 오류를 초래한 명령문(있을 경우)
- SQL(구조화 조회 언어) 패키지 작성(CRTSQLPKG) 명령을 실행한 결과 메시지

```

5763SS1 V3R1M0 940909          Create SQL package          04/19/94 14:30:31  Page    1
Record *...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8  SEQNBR Last change
Program name.....TST/UPDATEPGM
Relational database.....*PGM
User .....*CURRENT
Replace.....*YES
Default Collection.....*PGM
Generation level.....10
Printer file.....*LIBL/QSYSPRT
Object type.....*PGM
Module list.....*ALL
Text.....*PGMTXT
Source file.....TST/QCSRC
Member.....UPDATEPGM

```

그림 18. CRTSQLPKG에서 비롯한 리스팅 (1/2)

```

5763SS1 V3R1M0 940909          Create SQL package          04/19/94 14:30:31 Page 2
Record *...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 SEQNBR Last change
17 UPDATE RWDS / DPT1 SET deptcode = NULL
DIAGNOSTIC MESSAGES
MSG ID SEV RECORD TEXT
SQL0204 10      17 Position 17 DPT1 in RWDS type *FILE not found.
SQL5057          SQL Package UPDATEPGM in TST created at KC000 from
module UPDATEPGM.
Message Summary
Total      Info      Warning      Error      Severe      Terminal
1          0          1          0          0          0
10 level severity errors found in source
***** END OF LISTING *****

```

그림 18. CRTSQLPKG에서 비롯한 리스팅 (2/2)

SQLCODE 및 SQLSTATE

오류가 발생하면 SQL을 사용하는 프로그램 인터페이스가 어플리케이션 프로그램으로 오류 정보를 리턴합니다. SQLSTATE와 해당 SQLCODE는 SQLCA(SQL 통신 영역) 또는 SQL 진단 영역에서 어플리케이션 프로그램으로 리턴합니다. SQLCA는 데이터베이스 관리 시스템이 가장 최근에 실행된 SQL문에 대한 정보로 갱신한 변수의 컬렉션이며 어플리케이션이 제공한 공간의 제어 블록에 있습니다. SQL 진단 영역은 가장 최근에 실행한 SQL문에 대한 광범위한 정보를 전달하도록 되어 있는 보다 복잡한 기억장치 영역이며 데이터베이스 관리자가 제공하는 공간에 있습니다.

SQL 오류가 감지되면 SQLSTATE라고 하는 5자 글로벌 변수가 오류의 성격을 식별합니다. SQLSTATE뿐만 아니라 정수 SQLCODE도 사용할 수 있습니다. 그러나, SQLCODE는 현재 네 개의 IBM 관계형 데이터베이스(RDB) 제품간 똑같은 오류 조건에 대해 똑같은 리턴 코드를 리턴하지 않습니다. SQLSTATE는 어플리케이션이 연결되는 DB2 제품과 관계 없이 어플리케이션 프로그램이 특정 오류 조건이나 오류 종류를 테스트할 수 있도록 설계되었습니다.

SQL에서 명령문 처리 도중 심각한 오류가 발생할 경우, SQLCODE는 음수가 됩니다(예를 들어, SQLCODE -204). SQL에서 예외적이지만 유효한 조건(경고)이 생긴 경우, SQLCODE는 양수가 됩니다(예를 들어, SQLCODE +100). SQL에서 명령문 처리 도중 오류 또는 예외적인 조건이 생기지 않을 경우, SQLCODE는 0입니다. 모든 iSeries용 DB2 Universal Database SQLCODE는 라이브러리 QSYS 안의 메세지 파일 QSQLMSG 내에 상응하는 메세지가 있습니다. 예를 들어, SQLCODE -204는 메세지 ID SQL0204로 기록됩니다.

리턴된 오류 정보가 중요한 프로그램 진단 틀이기 때문에, 어플리케이션 프로그램에 리턴된 SQLCA 또는 SQL 진단 영역 내에 수록된 정보의 일부를 표시하기 위해 필수적인 지침을 포함시킬 것을 권장합니다. 아래에서 설명하는 메세지 토큰도 문제점을 진단하는 데 매우 유용합니다.

- SQLSTATE

리턴 코드

- SQLCODE(SQLCA) 또는 DB2_RETURNED_SQLCODE(SQL 진단 영역)

리턴 코드

- SQLERRD(3) (SQLCA) 또는 ROW_COUNT(SQL 진단 영역)

SQL에 의해 갱신, 삽입 또는 삭제된 행 수.

SQLCA 또는 SQL 진단 영역에 대한 자세한 정보는 SQL 참조서 주제의 SQLCA, SQL 진단 영역 및 SQLDA 제어 블록에 대한 정보를 참조하십시오.

SQL 메시지 및 코드 주제는 각각의 SQLCODE, 연관된 메시지 ID, 연관된 SQLSTATE 및 메시지의 텍스트를 나열합니다. 메시지 설명 표시(DSPMSGD) 명령을 사용하면 완전한 메시지를 온라인 상에서 볼 수 있습니다.

분산 관계형 데이터베이스 SQLCODE 및 SQLSTATE

다음 리스트는 분산 관계형 데이터베이스(RDB) 처리와 관련된 더 많은 일반적인 SQLCODE 및 SQLSTATE를 몇 개 나열한 것입니다. 모든 SQLCODE 및 SQLSTATE에 대한 내용은 SQL 프로그래밍 개념 책을 참조하십시오. 이러한 SQLCODE(및 연관된 SQLSTATE)에 대한 간략한 설명에서 메시지 자료 필드는 앰퍼샌드(&)와 숫자(예: &1)로 식별됩니다. 이러한 필드의 대체 텍스트는 어플리케이션 프로그램이 SQLCA를 사용할 경우 SQLERRM에 저장되고, 어플리케이션 프로그램이 SQL 진단 영역을 사용할 경우에는 DB2_ORDINAL_TOKEN_n(여기서 n은 토큰 번호임)에 저장됩니다. 메시지 설명 표시(DSPMSGD) 명령을 사용하면 임의의 SQLCODE에 대한 자세한 원인 및 복구 정보를 찾을 수 있습니다.

표 11. SQLCODE 및 SQLSTATE

SQLCODE	SQLSTATE	설명
+100	02000	이 SQLSTATE는 빈 표에 대한 SQL연산을 하거나, SQL UPDATE 또는 SQL DELETE문에서 식별되는 행이 없을 때 또는 SQL FETCH문의 커서가 결과표의 마지막 행을 가리킬 때 자료 없음 예외 경고를 보고합니다.
+114	0A001	관계형 데이터베이스(RDB) 이름 &1. 현재 서버와 똑같지 않은 경우 &2;
+304	01515	이 SQLSTATE는 호스트 변수가 검색된 값을 보유할 정도로 크지 않기 때문에 호스트 변수 리스트 또는구조에 FETCH 또는 SELECT를 넣는 일이 발생하면 이에 대한 경고를 보고합니다. FETCH 또는 SELECT는 표시된 SELECT 항목에 대해 자료를 리턴하지 않으며 인디케이터 변수는 -2로 설정되어 널값의 리턴을 표시하고 처리는 계속됩니다.
+331	01520	문자 변환이 실행될 수 없습니다.
+335	01517	문자 변환으로 대체 문자가 만들어졌습니다.
+551	01548	&2 유형 &3 내에 권한을 부여받은 오브젝트 &4가 없습니다.
+552	01542	&1에 대한 권한이 없습니다.

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
+595	01526	확약(commit) 레벨 &1이 &2로 절상되어 잠겼습니다.
+802	01519	이 SQLSTATE는 SQL SELECT문의 SELECT 리스트나 SELECT, UPDATE 또는 DELETE문의 탐색 조건 또는 UPDATE문의 SET절에 있는 SQL 산술 함수 또는 연산식을 처리하는 중에 발생한 산술 예외 경고를 보고합니다. 오류가 발생한 각 표현식에 대해 인디케이터 변수는 -2로 설정되어 널값의 리턴을 표시합니다. 연관된 자료 변수에는 아무 변화가 일어나지 않으며 처리는 계속됩니다.
+863	01539	SBCS 문자만이 관계형 데이터베이스(RDB) &1에 허용됩니다.
+990	01587	이 SQLSTATE는 이단계 프로세스 중에 적어도 하나의 프로세스 대상으로부터 응답이 지연되었거나 복합된 결과가 나왔음을 보고합니다.
+30104	01615	바인드 옵션이 무시되었습니다.
-114	42961	관계형 데이터베이스(RDB) &1. 현재 서버와 똑같지 않은 경우, &2
-144	58003	섹션 번호 &1이 유효하지 않습니다. 현재 최고 섹션 번호는 &3입니다. 이유는 &2입니다.
-145	55005	이질적인 어플리케이션 서버에 대해서는 순환이 지원되지 않습니다.
-175	58028	확약(commit) 작업이 실패했습니다.
-189	22522	코드화 문자 세트 ID &1이 유효하지 않습니다.
-191	22504	복합된 자료 값이 유효하지 않습니다.
-250	42718	디렉토리에 정의되지 않은 로컬 관계형 데이터베이스(RDB)입니다.
-251	2E000 42602	관계형 데이터베이스(RDB) 이름 &1 내의 문자가 유효하지 않습니다.
-300	22024	NUL로 종료된 입력 호스트 변수 또는 매개 변수가 NUL을 포함하지 않습니다.
-302	22001	입력 호스트 변수 &2에 대한 변환 오류
-330	22021	문자 변환이 실행될 수 없습니다.
-331	22021	문자 변환이 실행될 수 없습니다.
-332	57017	CCSID &1 및 CCSID &2 사이의 문자 변환이 유효하지 않습니다.
-334	22524	문자 변환으로 긴 이름에 대한 절단이 발생했습니다.

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
-351 -352	56084	지원되지 않는 SQLTYPE이 선택 리스트 또는 입력 리스트에서 발견되었습니다.
-426	2D528	어플리케이션 실행 환경에 유효하지 않은 조작입니다. 이 SQLSTATE는 동적 COMMIT 제한 환경에서 COMMIT를 실행하기 위해 EXCSQLIMM또는 EXCSQLSTT를 사용하려는 시도가 있었다고 보고합니다.
-427	2D529	어플리케이션 실행 환경에 유효하지 않은 조작입니다.
-501 -502 -507	24501	유효하지 않은 커서 상태로 인해 실행이 실패했습니다. 식별된 커서가 열리지 않습니다.
-510	42828	이 SQLSTATE는 블로킹 프로토콜을 이용하여 행을 폐지하는 커서에 대해 DELETE WHERE CURRENT OF CURSOR 또는 UPDATE WHERE CURRENT OF CURSOR 실행 시도를 보고합니다.
-525	51015	명령문 상의 오류입니다.
-551	42501	&2 유형 *&3 내의 오브젝트 &1에 대한 권한이 없습니다.
-552	42502	&1에 대한 권한이 없습니다.
-683	42842	FOR DATA절 또는 CCSID절이 지정된 유형에 대해 유효하지 않습니다.
-752	0A001	어플리케이션 처리가 연결 가능 상태가 아닙니다. 이유 코드는 &1입니다.
-802	22003 22012	숫자 값이 범위를 벗어났으며 피셋수가 0인 나눗셈은 유효하지 않습니다.
-805	51002	&2 내의 SQL 패키지 &1을 찾을 수 없습니다.
-818	51003	지속성 토큰이 일치하지 않습니다.
-842	08002	이미 연결되어 있습니다.
-862	55029	로컬 프로그램이 리모트 관계형 데이터베이스(RDB)에 대한 연결을 시도했습니다.
-871	54019	너무 많은 CCSID 값이 지정되었습니다.
-900	08003	연결되어 있지 않습니다.
-918	51021	SQL문은 어플리케이션 프로세스가 롤백 연산을 실행하기 전까지는 실행되지 않습니다.
-922	42505	이 SQLSTATE는 어플리케이션 서버에 연결 처리를 하는 중에 일반 사용자를 인증하는데 실패했음을 보고합니다.
-925 -926	2D521	현재 환경에서 SQL COMMIT 또는 ROLLBACK문은 유효하지 않습니다.
-950	42705	관계형 데이터베이스(RDB) &1이 관련 디렉토리에 존재하지 않습니다.

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
-952	57014	SQL문 처리가 ENDRDBRQS 명령으로 종료되었습니다.
-969	58033	어플리케이션 리퀘스터 드라이버 프로그램에 요구를 전달하는 중 오류가 발생했습니다
-7017	42971	확약 제어가 이미 DDM 목표에 대해 사용 중입니다.
-7018	42970	COMMIT HOLD 또는 ROLLBACK HOLD가 허용되지 않습니다.
-7021	57043	어플리케이션 서버 상에서 실행하려는 로컬 프로그램.
-30000	58008	분산 관계형 데이터베이스 구조(DRDA) 프로토콜 오류.
-30001	57042	분배 SQL 프로그램에 대한 호출이 허용되지 않습니다.
-30020	58009	분산 관계형 데이터베이스 구조(DRDA) 프로토콜 오류.
-30021	58010	리모트 서버에 의해 지원되지 못한 분산 관계형 데이터베이스(DRDB).
-30040	57012	관계형 데이터베이스(RDB)&1 내의 DDM 자원 &2를 사용할 수 없습니다.
-30041	57013	관계형 데이터베이스(RDB)&1 내의 DDM 자원을 사용할 수 없습니다.
-30050	58011	바인드 프로세스를 처리하는 동안 DDM 명령 &1이 유효하지 않습니다.
-30051	58012	지정된 패키지명 및 지속성 토큰의 바인드 프로세스가 활동 중이 아닙니다.
-30052	42932	프로그램 준비 전제가 부정확합니다.
-30053	42506	소유자 &1에 대해 패키지를 작성할 권한이 없습니다.
-30060	08004	관계형 데이터베이스(RDB) &1에 대해 권한이 없는 사용자입니다.
-30061	08004	관계형 데이터베이스(RDB) &1을 찾을 수 없습니다.
-30070	58014	DDM(분산 자료 관리) 명령 &1이 지원되지 않습니다.
-30071	58015	분산 자료 관리(DDM) 오브젝트 &1이 지원되지 않습니다.
-30072	58016	분산 자료 관리(DDM) 매개변수 &1이 지원되지 않습니다.
-30073	58017	분산 자료 관리(DDM) 매개변수 값 &1이 지원되지 않습니다.
-30074	58018	분산 자료 관리(DDM) 응답 메시지 &1이 지원되지 않습니다.

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
-30080	08001	분배 데이터베이스 처리 중 통신 오류가 발생했습니다.
-30082	08001	분배 데이터베이스 연결 시도 상 권한 실패
-30090	25000 2D528 2D529	읽기 전용 어플리케이션 서버에 대해 유효하지 않은 변경 요구.
-30104	56095	유효하지 않은 바인드 옵션. 이 SQLSTATE는 하나 이상의 바인드 옵션이 서버에서 유효하지 않음을 보고합니다. 바인드 연산이 종료됩니다. 오류가 생긴 첫 번째 바인드 옵션이 SQLERRMC에 보고됩니다.
-30105	56096	바인드 옵션 충돌입니다. 바인드 연산이 종료됩니다. 충돌이 발생한 바인드 옵션은 SQLERRMC에 보고됩니다.
AR에 의해 인식되지 않음	58020	오류 또는 경고에 대해 SQLSTATE 값이 지정되지 않았습니다.

시스템 및 통신 문제점

시스템 또는 통신에 문제가 발생하면 메시지가 생성됩니다. 시스템 감지 문제점은 자동적으로 문제점 기록부로 입력되며 이곳에서 문제점이 보여지고 분석됩니다.

자세한 정보는 iSeries 문제점 기록부를 참조하십시오.

iSeries 문제점 기록부

시스템 감지 문제점은 자동적으로 문제점 기록부로 입력됩니다. 문제점 기록부에 사용자 감지 문제점도 입력할 수 있습니다. 임의의 시스템 명령 행으로부터 문제점 분석(ANZPRB) 명령을 입력하여 언제든지 기록된 문제점에 대한 문제점 분석을 실행할 수 있습니다. 이 명령으로 분석 프로시저어를 실행하고 문제점 기록부 내의 추가적인 문제점 관련 정보를 저장할 수 있습니다.

문제점 기록부를 보기 위해서는 문제점에 대한 작업(WRKPRB) 명령을 사용하십시오. 다음의 화면들은 문제점 기록부의 두 가지 뷰를 나타내고 있습니다.

문제점에 대한 작업

시스템: KC000
 위치. 문제점 ID

옵션을 입력한 후 Enter를 누르십시오.
 2=변경 4=삭제 5=세부사항 표시 6=세부사항 인쇄
 8=문제점에 대한 작업 9=경고에 대한 작업 12=주 입력

Opt	문제점 ID	상태	문제점 설명
—	9114350131	READY	User detected a hardware problem on a differen
—	9114326436	OPENED	System cannot call controller . No lines avail
—	9114326281	OPENED	Line failed during insertion into the token-r
—	9114324416	OPENED	Device failed, recovery stopped.
—	9114324241	OPENED	System cannot call controller . No lines avail
—	9114324238	OPENED	System cannot call controller . No lines avail
—	9114324234	OPENED	System cannot call controller . No lines avail
—	9114324231	OPENED	System cannot call controller . No lines avail
—	9114324227	OPENED	System cannot call controller . No lines avail
—	9114324224	OPENED	System cannot call controller . No lines avail
—	9114324218	OPENED	System cannot call controller . No lines avail

계속...

F3=종료 F5=화면정리 F6=리스트 인쇄 F11=날짜 및 시간 표시
 F12=취소 F16=준비된 문제점 보고 F24=추가 키

다음 화면을 보기 위해서 처음 뷰에서 F11을 누르십시오.

문제점에 대한 작업

시스템: KC000
 위치. 문제점 ID

옵션을 입력한 후 Enter를 누르십시오.
 2=변경 4=삭제 5=세부사항 표시 6=세부사항 인쇄
 8=문제점에 대한 작업 9=경고에 대한 작업 12=주 입력

Opt	문제점 ID	날짜	시간	기점
—	9114350131	03/29/92	14:36:05	APPN.KC000
—	9114326436	03/29/92	07:41:59	APPN.KC000
—	9114326281	03/29/92	07:39:17	APPN.KC000
—	9114324416	03/29/92	07:06:42	APPN.KC000
—	9114324241	03/29/92	07:03:38	APPN.KC000
—	9114324238	03/29/92	07:03:35	APPN.KC000
—	9114324234	03/29/92	07:03:31	APPN.KC000
—	9114324231	03/29/92	07:03:27	APPN.KC000
—	9114324227	03/29/92	07:03:24	APPN.KC000
—	9114324224	03/29/92	07:03:20	APPN.KC000
—	9114324218	03/29/92	07:03:14	APPN.KC000

계속...

F3=종료 F5=화면정리 F6=리스트 인쇄 F11=설명 표시 F12=취소
 F14=신규 문제점 분석 F16=준비된 문제점 보고 F18=경고에 대한 작업

iSeries 문제점 기록부를 통해서 로컬 서버 상에 기록된 모든 문제점의 리스트를 표시할 수 있습니다. 또한 다음과 같이 특정한 문제점에 대한 자세한 정보도 표시할 수 있습니다.

- 문제점이 발생한 장치의 제품 유형 및 일련 번호
- 문제점이 발생한 날짜 및 시간
- 실패한 부분 및 그 위치
- 문제점 상태

또한 문제점 기록부에서 문제점을 분석하고, 보고하며 어떤 서비스 활동이 취해져야할지 판별할 수 있습니다.

실패 보고를 위한 자료 수집

다음 섹션에서는 iSeries 서버 상의 분산 관계형 데이터베이스(RDB) 내에서 문제점을 진단할 수 있도록 돕기 위해 인쇄할 수 있는 자료 종류를 설명합니다. 이 자료는 OS/400 프로그램에 의해 생산된 것입니다. 또한 시스템 오퍼레이터 메시지 및 어플리케이션 프로그램(그 자료와 함께)을 사용하여 문제점을 진단할 수 있습니다.

- 작업 기록부 인쇄
- TCP/IP 서버 사전시작 작업의 작업 기록부 찾기
- 제품 활동 기록부 인쇄
- 작업 추적
- 통신 추적

작업 기록부 인쇄

iSeries 서버 상의 모든 작업에는 그 작업에 대해 입력된 요구와 관련된 정보를 수록한 작업 기록부가 있습니다. 사용자가 어플리케이션 리퀘스터(AR)에서 문제점을 발견했을 때, 작업 기록부 내의 정보가 문제점 진단에 도움이 될 것입니다. 이 정보를 얻는 쉬운 방법 중 하나는 다음 명령어로 사용자 종료(Sign Off)를 실행하는 것입니다.

```
SIGNOFF *LIST
```

이 명령은 사용자의 작업 기록부 복사본을 인쇄하거나 인쇄를 위한 출력 대기행렬에 작업 기록부를 위치시킵니다.

작업 기록부를 인쇄하는 또다른 방법은 어플리케이션 작업 설명(JOBD) 상에 LOG(4 00 *SECLVL)를 지정하는 것입니다. 작업이 완료된 후, 모든 메시지가 특정한 작업에 대한 작업 기록부에 기록됩니다. 출력 대기행렬에 작업 기록부를 위치지정하거나 인쇄 프로시듀어를 실행해서 작업 기록부를 인쇄할 수도 있습니다. 서버 상에 작업 및 작업 기록부의 위치지정 방법에 대한 내용은 102 페이지의 『분산 관계형 데이터베이스의 작업 기록부를 사용한 요구 정보 트래킹』을 참조하십시오.

어플리케이션 서버(AS)에 대한 작업 기록부도 역시 문제점 진단에 도움이 될 것입니다. 어플리케이션 서버 작업에 대한 작업명을 찾는 방법에 대한 내용은 103 페이지의 『분산 관계형 데이터베이스 작업 찾기』를 참조하십시오.

TCP/IP 서버 사전시작 작업의 작업 기록부 찾기

DDM TCP/IP 서버와 관련된 QPWTSRVR 사전시작 작업이 진행된 연결이 종료되면, 사전시작 작업은 다른 연결에 의해 재순환되어 사용됩니다. 이 상황이 발생하면, 종료된 연결과 관련된 작업 기록부는 지워집니다. 하지만 어떤 경우에는, 작업 기록부가 지워지기 전에 프린터 파일로 스푼됩니다. 작업 기록부는 클라이언트 사용자 ID 및 암호가 유효화되지 않으면 스푼되지 않습니다. 유효화가 성공적인 경우, 사용자 ID 및 암호가 작업 기록부가 스푼되는 조건이 됩니다.

- V5R1 이상에서는 서버 작업의 메시지 기록 텍스트 레벨은 *SECLVL 또는 *MSG입니다.
- 연결을 종료시키려는 요구 처리 중 발생한 심각한 오류를 서버 요청 핸들러 라우팅 프로그램에서 감지한 경우.

- 사전시작 작업이 서비스된 경우(서비스 작업 시작(STRSRVJOB) 명령을 사용하여).
- 클라이언트 또는 서버의 QRWOPTIONS 자료 영역이 서버 작업이 충족시킨 작업 기록부 출력 조건을 지정한 경우. 자세한 정보는 QRWOPTIONS 자료 영역 사용법 주제를 참조하십시오.

몇몇 이유로 작업 기록부 정보를 얻으려 할 수 있습니다. 이는 분명 오류를 진단하는데 유용합니다. 또 성능에 관련된 문제점을 분석하는데도 유용합니다. 예를 들어, 디버그 하에서 실행될 때 생성되는 SQL optimizer 자료를 얻길 원하는 경우, 수동으로 서비스 작업을 시작하여 디버그 시작(STRDBG) 명령을 실행하거나 또는 QRWOPTIONS 자료 영역에서 적용할 수 있는 옵션을 하나 이상 설정하여 작업 기록부를 보유하도록 할 수 있습니다.

연결 단계에서 실패가 발생한 작업의 기록부는 위에서 기술한 프로세스에 의해 저장되지 않습니다. 프로세스에 의해 저장되는 작업은 사전시작 작업 ID 하에 저장됩니다. 작업 기록부를 발견하려면 다음의 명령을 실행하십시오.

```
WRKJOB userid/QPRTJOB
```

사용자 ID란 어플리케이션 서버(AS)에 대한 CONNECT 상에 사용된 사용자 ID입니다. 사용자 ID는 만일 그것을 모를 경우, 어플리케이션 서버 상에서 기록부 표시(DSPLOG) 명령을 사용하여 찾아낼 수 있습니다.

다음과 같이 매개변수 사용으로 원하지 않는 메시지를 필터링 할 수 있습니다.

```
DSPLOG PERIOD(('11:00')) MSGID(CPI3E34)
```

다음 메시지를 찾으십시오. 하지만 QRWOPTIONS 자료 영역이 이 메시지(CPI3E34)를 감추는데 사용되었다면 이력 기록부에는 나타나지 않는다는 점을 주의하십시오.

```
ddd, ttt에 yyy 사용자를 서버스하는 DDM 작업 xxxx
```

제품 활동 기록부 인쇄

iSeries 서버 상의 제품 활동 기록부는 기계 점검, 장치 오류 및 테이프와 디스켓 통계에 대한 레코드입니다. 오류 기록부에는 각 FFDC 덤프의 처음 1000바이트를 포함하는 FFDC 정보도 수록되어 있습니다. 이 오류를 검토하면 문제점의 본질을 판별할 수 있을 것입니다.

시작한 서버에 대한 제품 활동 기록부를 인쇄하기 위해서는 다음 단계를 따르십시오.

1. 임의의 명령 행에 오류 기록부 인쇄(PRTERLOG) 명령을 입력하고 F4(프롬프트) 키를 누르십시오. 오류 기록부 인쇄 화면이 나타납니다.
2. 인쇄를 원하는 기록부 정보 유형에 대한 매개변수를 입력하고 Enter를 누르십시오. 기록부 정보가 작업에 대해 식별된 출력 대기행렬에 송신됩니다.
3. 작업에 대한 작업(WRKJOB) 명령을 입력하십시오. 작업에 대한 작업 화면이 나타납니다.
4. 스폴 파일에 대한 작업 옵션을 선택하십시오. 스폴 파일에 대한 작업 화면이 나타납니다.
5. 스폴 파일 리스트 맨 아래 또는 아래 부분에 방금 작성한 기록부 파일을 검색하십시오.
6. 기록부 파일 다음의 *Opt* 열 내에 있는 인쇄 상태 옵션에 대한 작업을 입력하십시오. 인쇄 상태에 대한 작업 화면이 나타납니다.

- 7. 인쇄 상태에 대한 작업 화면 상에서 파일 상태를 변경하기 위해 상태 옵션 변경을 사용하십시오. 그리고 파일을 인쇄하기 위한 프린터를 지정하십시오.

작업 추적

때때로 어떤 문제점은 특정 프로그램에 대해 추적이 불가능합니다. 이러한 경우 모듈 흐름과 OS/400 자료 획득의 추적에 CL 명령을 포함하여 STRTRC(추적 시작)와 TRCJOB(작업 추적)를 사용할 수 있습니다. 이들 둘은 문제점 분석 프로시더가 문제점에 대한 충분한 정보를 공급하지 못할 때 사용되어야 합니다. 분산 데이터베이스 어플리케이션의 경우, 분산 데이터베이스 요구 및 응답 자료 스트림 캡처를 위해 이들 명령을 사용할 수 있습니다.

TRCJOB

TRCJOB는 두 가지 추적 틀 중에서 더 오래된 틀입니다. 추적 레코드가 생성됨에 따라 레코드가 내부 추적 기억장치 영역 내에 저장됩니다. 추적이 종료되면, 추적 레코드는 스펴 프린터 파일(QPSRVTRC)에 기록되거나 데이터베이스 출력 파일에 기록됩니다.

샘플 추적 시나리오는 다음과 같습니다.

```
TRCJOB SET(*ON) TRCTYPE(*ALL) MAXSTG(2000)
          TRCFULL(*WRAP) EXITPGM($SCFTRC)
CALL QCMD
TRCJOB SET(*OFF) OUTPUT(*PRINT)
WRKOUTQ output-queue-name
```

QPSRVTRC라는 이름의 스펴 파일을 발견할 것입니다. 스펴 파일에 추적 기록이 수록되어 있습니다. 추적 작업 사용에 대한 자세한 정보는 267 페이지의 부록 C 『추적 작업 및 FFDC 자료 해석』을 참조하십시오.

STRTRC

- 또한 STRTRC(추적 시작) 명령을 사용하여 추적을 수행할 수도 있습니다. STRTRC는 TRCJOB보다 유연하고 방해를 덜 받습니다. 여러 작업을 추적할 수 있으며 모듈 흐름에 대한 세부사항을 보여줍니다. 추적 레코드가 생성됨에 따라 레코드가 세션 ID에 의해 식별되는 내부 추적 기억장치 영역 내에 저장됩니다. ENDTRC(추적 종료)를 사용하여 추적을 종료하면 추적 레코드는 사용자 지정 라이브러리에 데이터베이스 파일 세트로 저장됩니다. 이러한 파일은 스펴 프린터 파일(QPSRVTRC)에 기록하거나 PRTRC를 발행하여 데이터베이스 출력 파일로 보낼 수 있습니다. 자세한 정보는 CL 프로그래밍 주제의 STRTRC 및 ENDTRC를 참조하십시오.

- 샘플 추적 시나리오는 다음과 같습니다.

- 주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```
STRTRC SSNID(DRDATRACE) JOB((*ALL/QUSER/QRWTSRVR)) MAXSTG(160000)
TRCFULL(*STOPTRC)
```

- 실패한 DRDA 시나리오 실행:

```
ENDTRC SSNID(DRDATRACE) DTALIB(TRACELIB)
PRTRC DTAMBR(DRDATRACE) DTALIB(TRACELIB)
```

- 1 어플리케이션 서버 작업의 작업 추적을 얻어야 할 경우, 서버에서 서비스 작업을 시작해야만 합니다. 187 페이지의 『어플리케이션 서버 문제점을 진단하기 위한 서비스 작업 시작』을 참조하십시오.

통신 추적

분산 관계형 데이터베이스(RDB)에 액세스하기 위해 DRDA를 사용할 때 CPF3Exx 범위 또는 CPF91xx 범위 내의 메시지를 얻었다면, 통신 추적을 실행해야만 합니다. 다음 리스트에서는 이 범위 내의 일반적인 메시지에 대한 것입니다.

표 12. 통신 추적 메시지

MSG ID	설명
CPF3E80	DDM 자료 스트림 구문 오류
CPF91xx	DDM 프로토콜 오류
CPF3E83	유효하지 않은 FD0:CA 설명자
CPF3E84	자료 불일치 오류

두 가지 유형의 통신 추적을 실행할 수 있습니다. 첫 번째는 표준 통신 추적입니다. 두 번째는 TRCTCPAPP 함수입니다. TRCTCPAPP는 IPSec 또는 보안 소켓 프로토콜이 자료 스트림을 암호화하는 위치에 대한 지능 추적을 제공합니다. 암호화 이전, 그리고 암호해제 이후 자료를 캡처합니다. 그리고 암호화되지 않은 자료 스트림의 추적 또한 훌륭히 완수합니다. LOOPBACK이 사용되는 시스템 내부의 DRDA 흐름을 추적하는 데 필수적입니다. 이 두 가지 유형의 추적 실행에 대한 지시에 대해서는 다음 아래의 섹션을 참조하십시오.

표준 통신 추적

통신 추적 기능으로 통신 구성 오브젝트에 대한 자료 추적을 시작하고 중단할 수 있습니다. 자료 추적 실행 이후, 인쇄 또는 보기에 대한 자료를 포맷할 수 있습니다. 출력 대기행렬 내에서만 프린터 파일을 볼 수 있습니다.

통신 추적 옵션은 시스템 서비스 툴(SST) 하에서 실행됩니다.통신 추적이 진행 중일때 SST로 구성 오브젝트를 사용할 수 있습니다. 분배 데이터베이스 네트워크 내에서 사용할 수 있는 통신 유형 모두에 대해 자료의 추적 및 포맷이 가능합니다.

서버에 연결된 임의의 화면으로부터 iSeries 통신 추적을 실행할 수 있습니다. *SERVICE에 대한 특수 권한 (SPCAUT)이 있다면 누구나 iSeries 서버에 대한 추적을 실행할 수 있습니다. 통신 추적은 모든 회선 속도를 지원합니다. 통신 제어기에 사용할 수 있는 프로토콜의 최대 집단 회선 속도는 V5R1 보충 매뉴얼 웹 사이트

에서 *Communications Management*  매뉴얼을 참조하십시오.

다음 상황에 대해서 통신 추적을 사용해야만 합니다.

- 문제점 분석 프로시더가 문제점에 대한 충분한 정보를 공급하지 못합니다.
- 프로토콜 손상이 문제점의 원인으로 판단됩니다.
- 회선 잡음이 문제점의 원인으로 판단됩니다.
- 시스템 네트워크 구조(SNA) bind 문제점이 있다는 오류 메시지를 수신했습니다.

통신 추적에 의해 생성된 자료를 정확히 해석하기 위해 사용하는 회선 프로토콜에 대해 자세히 알고 있어야 합니다. 분산 관계형 데이터베이스(DRDA) 자료 스트림에 대한 정보에 대해서는 268 페이지의 『예: RW 추적 자료 분석』을 참조하십시오.

가능하다면 언제든지 회선 연결변환(vary on) 이전에 통신 추적을 시작하십시오. 그러면 연결변환(vary on)이 진행됨에 따라 회선에 대한 가장 정확한 샘플을 얻을 수 있습니다.

APPC 추적을 실행하여 그 출력에 대해 작업하기 위해서는 반드시 실행 중인 회선, 제어기 및 장치가 무엇인지 알아야만 합니다. 만일 이 정보를 알지 못한다면, 『회선, 제어기 및 장치 설명 찾기』를 참조하십시오.

TCP/IP 추적의 출력에서 원하지 않는 자료의 형식을 정해 출력되는 것을 피하기 위해 소스 및 어플리케이션 소스(AS)의 IP 주소를 지정할 수 있습니다. 때로는 IP 주소를 지정하는 대신 더 쉬운 방법인 포트 번호를 지정하는 것만으로도 충분합니다.

다음 명령이 통신 추적을 시작, 중단, 인쇄 및 삭제합니다.

통신 추적 시작(STRCMNTRC) 명령

지정된 행 또는 네트워크 인터페이스 설명에 대한 통신 추적을 시작합니다. 추적 매개변수에서 바이트 수의 시작 바이트 값으로 *MAX를 지정하십시오. 통신 추적은 통신 추적 종료(ENDCMNTRC) 명령을 실행할 때까지 계속됩니다.

통신 추적 종료(ENDCMNTRC) 명령

지정된 행 또는 네트워크 인터페이스 설명에서 실행된 통신 추적을 종료합니다.

통신 추적 인쇄(PRTCMNTRC) 명령

지정된 행 또는 네트워크 인터페이스 설명에 대한 통신 추적 자료를 스폴 파일 또는 출력 파일에 이동시킵니다. 포맷 SNA(Systems Network Architecture) 자료 중 매개변수에만 *YES를 지정하십시오.

통신 추적 삭제(DLTCMNTRC) 명령

지정된 행 또는 네트워크 인터페이스 설명에 대한 통신 추적을 삭제합니다.

회선, 제어기 및 장치 설명 찾기: 어플리케이션 서버 작업이 시작된 제어기 및 장치를 발견하기 위해서 구성 상태에 대한 작업(WRKCFGSTS) 명령을 사용하십시오. 예를 들면, 다음과 같습니다.

```
WRKCFGSTS CFGTYPE(*DEV)
          CFGD(*LOC)
          RMTLOCNAME(DB2ESYS)
```

RMTLOCNAME 키워드에 대한 값은 어플리케이션 서버명입니다.

구성 상태에 대한 작업(WRKCFGSTS) 명령은 리모트 위치명으로 지정된 서버명을 갖는 장치 모두를 나타내고 있습니다. 한 번에 한 장치만 연결변환이 가능하기 때문에 어떤 장치가 사용 중인지 알 수 있습니다. 장치 설명에 대한 작업을 위해서 옵션 8을 사용하십시오. 그리고 옵션 5를 사용하여 화면을 표시하십시오. 접속된 제어기 필드에 제어기명이 있습니다. 제어기 및 장치 설명에 대한 작업을 위해 (WRKCFGSTS) 명령을 사용할 수 있습니다. 예를 들면, 다음과 같습니다.


```

WRKCFGSTS CFGTYPE(*CTL)
          CFGD(PCXZZ1205) /* workstation */
WRKCFGSTS CFGTYPE(*CTL)
          CFGD(LANSLKM) /* AS/400 on token ring */

```

CFGD 값은 이 섹션의 첫 번째 예 내의 장치 설명에서 확보한 제어기 명입니다.

구성 상태에 대한 작업(WRKCFGSTS) 명령로 인한 출력에는 통신 추적에 대한 작업 시 필요한 회선 설명의 이름도 포함됩니다. 만일 옵션 8을 선택하고 그 후에 옵션 5를 선택하여 제어기 설명을 표시했다면, 사용 중인 교환 회선(switched line) 매개변수가 회선 설명 이름을 표시합니다. LAN 리모트 어댑터 주소는 리모트 서버의 토큰-링 주소를 제공합니다.

회선명을 찾는 또다른 방법은 서버에 대한 모든 회선 설명을 나열하는 회선 설명에 대한 작업(WRKLIND) 명령을 사용하는 것입니다.

TRCTCPAPP 추적

이 함수는 통신 상 TCP/IP를 사용할 경우에만 유효합니다.

- | 이 추적 틀을 사용하는 하나의 예는 자료가 암호화될 때 환경에서 전송할 일반 텍스트를 표시하는 것입니다.
- | 추적 자료는 송신지측에서 암호화하기 전과 수신지측에서 암호화한 후에 캡처됩니다. 그러나 이 추적 틀은 다른 환경에서도 유용합니다.

TCP/IP 어플리케이션 추적(TRCTCPAPP) 명령을 사용하려면, *SERVICE 특수 권한을 부여받은 사용자 프로파일이 있어야만 합니다. 추적을 시작하려면 다음을 입력하십시오.

```
TRCTCPAPP *DDM
```

추적을 특정 포트에만 제한하려면 예를 들어, SSL에 대해 포트 448을 지정하려면 다음 예를 따르십시오.

```
TRCTCPAPP *DDM *ON RMTNETADR(*INET *N '255.255.255.255' 448)
```

추적 중인 통신이 완료되고 난 후에 다음 명령을 실행하고, 결과 스푼 파일을 참조하십시오.

```
TRCTCPAPP *DDM *OFF
```

- | 하나 이상의 연결을 추적할 경우에는 스푼 파일을 찾아서 각 QRWTSRVR 작업과 일치시켜야 합니다. 스푼 파일명은 QZBSTRC이고 작업은 QRWxxxxxx입니다. 여기서 xxxxxx는 스푼 파일의 사용자 자료에 넣은 작업 번호입니다.

***DDM 어플리케이션 사용에 대한 제약.**

*DDM 어플리케이션과 TCP/IP 어플리케이션 추적(TRCTCPAPP) 명령을 함께 사용할 때, 단일 송신 또는 수신 메시지에 대해 추적할

| TRCTCPAPP 추적 형식화

- | TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령은 논리 표시를 쉽게 읽도록 DRDA와 DDM 흐름을 구분하는 데 사용할 수 있습니다. 또한 상이한 환경에 도움이 될 수 있는 정보를 ASCII로 표시할 수도 있습니다. 이 형식화를 요구하려면 통신 추적을 종료할 때 다음을 입력하십시오.

| TRCTCPAPP APP(*DDM) SET(*OFF) ARGLIST('lv1=2')

| 다음은 이 매뉴얼의 너비에 맞게 편집한 형식화되지 않은 추적의 예입니다.

| 0080D0010001007A 200100162110D9C3 C8C1E2D5E3E24040
| 4040404040404040 *..}.....RCHASNTS *
| 0006210F2407000D 002FD8E3C4E2D8D3 F4F0F0000C112ED8
| E2D8F0F5F0F3F000 *.....QTDSQL400....QSQ05030.*
| 0A00350006119C00 2500062121241E00 062120241E0010D1
| 2A01000000000000 *.....J.....*
| 0000000000001621 35C1D7D7D54BD3D7 F0F6F6C1C2B9191C
| F706F90005213BF1 *.....APPN.LP066AB...7.9....1*

| 다음은 TRCTCPAPP를 사용하여 형식화한 동일한 추적입니다.

| -Datastream-----
| DATA: (ASCII) (EBCDIC)
| 0080D0010001007A 200100162110D9C3 .8'.....a...b.eā .0}.....RC
| C8C1E2D5E3E24040 4040404040404040 ç ë+ëë..... HASNTS
| 0006210F2407000D 002FD8E3C4E2D8D3 .Lb.f"....éèàëé<QTDSQL
| F4F0F0000C112ED8 E2D8F0F5F0F3F000éëë..... 400....QSQ05030.
| 0A00350006119C00 2500062121241E00 C...L.ó...LbbfK.ä.....
| 062120241E0010D1 2A01000000000000 LbafK..çk..... J.....
| 0000000000001621 35C1D7D7D54BD3D7b. &&+.<&APPN.LP
| F0F6F6C1C2B9191C F706F90005213BF1 ... â".c.L...bB. 066AB¼..7.9....1
| -Parsed-----

| RECV(AS) RQSDSS - Request Data Stream Structure
|
| LL: 128 CORR: 0001 CHAINED: n CONT ON ERR: n SAME CORR FOR NEXT DSS: n
|
| NM: ACCRDB - Access RDB
|
| LL: 122 CP: 2001
|
| NM: RDBNAM - Relational Database Name
|
| LL: 22 CP: 2110
|
| ASCII: eāç ë+ëë.....
|
| EBCDIC: RCHASNTS
|
| NM: RDBACCCL - RDB Access Manager Class
|
| LL: 6 CP: 210F
|
| CODE POINT DATA: 2407
|
| NAME: SQLAM - SQL Application Manager
|
| NM: TYPDEFNAM - Data Type Definition Name
|
| LL: 13 CP: 002F
|
| ASCII: éèàëé<...
|
| EBCDIC: QTDSQL400
|

```

| NM: PRDID - Product-Specific Identifier
|
| LL: 12 CP: 112E
|
| DATA:          (ASCII)      (EBCDIC)
|
| D8E2D8F0F5F0F3F0   éëé.....   QSQ05030
|
| NM: TYPDEFOVR - TYPDEF Overrides
|
| LL: 10 CP: 0035
|
| NM: CCSIDSBC - CCSID for Single-Byte Characters
|
| LL: 6 CP: 119C
|
| DATA:      (ASCII)      (EBCDIC)
|
| 0025      ..      ..
|
| NM: STTDECDEL - Statement Decimal Delimiter
|
| LL: 6 CP: 2121
|
| CODE POINT DATA: 241E
|
| NAME: DFTPKG - Package Default
|
| NM: STTSTRDEL - Statement String Delimiter
|
| LL: 6 CP: 2120
|
| CODE POINT DATA: 241E
|
| NAME: DFTPKG - Package Default
|
| NM: SXXPRDDTA - Extended Product Data
|
| LL: 16 CP: D12A
|
| DATA:          (EBCDIC)
|
| 0100000000000000   00000000 .....
|
| NM: CRRTKN - Correlation Token
|
| LL: 22 CP: 2135
|
| DATA:          (ASCII)      (EBCDIC)
|
| C1D7D7D54BD3D7F0   F6F6C1C2B9191CF7   &&+.<&... â".c. APPN.LP066AB¾..7
|
| 06F9 L. .9
|
| NM: TRGDFTRT - Target Default Value Return

```

첫 번째 실패 자료 캡처(FFDC) 자료 찾기

주: QSFWERRLOG 시스템 값이 *LOG로 설정되지 않는 한, FFDC 자료는 생성되지 않습니다.

다음은 iSeries 서버 상의 FFDC 자료 위치지정 방법에 대한 추가 정보입니다. 이 정보는 FFDC 자료 출력을 야기시키는 실패가 어플리케이션 서버(AS)에서 발생했을 때 특히 유용합니다. 대개 어플리케이션 리퀘스터(AR)에 대한 FFDC 자료는 어플리케이션 프로그램 실행 작업과 관련된 스펴 파일 중 하나에서 발견됩니다.

1. 메시지 표시(DSPMSG) QSYSOPR 명령을 실행하고 QSYSOPR 메시지 기록부 내의 Qccxyyyy 내에서 감지된 소프트웨어 문제점 메시지를 찾으십시오.(프로그램 명 내의 cc는 대개 RW입니다. 하지만 CN 또는 SQ가 되기도 합니다.) 이 메시지의 존재는 FFDC 자료가 생성되었다는 사실을 암시합니다. 메시지에 관계된 세부사항을 얻기 위해서 도움말 키를 사용할 수 있습니다. 메시지 도움말에서는 문제점 ID, 즉 문제점에 대한 작업(WRKPRB) 명령에 의해 제공된 리스트 내의 문제점을 식별하기 위해 사용하는 ID를 제공합니다. 문제점 레코드는 존재할 경우, 리스트의 맨 위 혹은 윗 부분에 있기 때문에 이 단계를 생략해도 상관없습니다.
2. 문제점에 대한 작업(WRKPRB) 명령을 입력하고 Qccxyyyy 내에서 감지된 소프트웨어 문제점 메시지로부터 프로그래밍(Qccxyyyy)을 지정하십시오. 프로그래밍을 사용하여 원치 않는 리스트 항목을 필터링 하십시오. 문제점 리스트가 제공된 때, 증후 스트링 및 오류 기록부 ID와 같은 문제점 세부사항을 더 많이 얻기 위해 문제점 ID가 수록된 행에서 옵션 5를 지정하십시오.
3. 오류 기록부 ID가 있다면, 시스템 서비스 툴 시작(STRSST) 명령을 입력하십시오. 첫 화면에서 서비스 툴 시작을 선택하십시오. 다음 화면에서, 오류 기록부 유틸리티를 선택하기 위해 1을 입력하십시오. 다음 화면에서, 오류 기록부 ID별 표시 또는 인쇄를 선택하기 위해 2를 입력하십시오. 다음 화면에서 다음 작업을 진행하십시오.
 - 오류 기록부 ID를 입력하십시오.
 - 16진 표시를 원한다면 Y를 입력하십시오.
 - 인쇄 또는 표시 옵션을 선택하십시오.

표시 옵션은 32바이트 대신 행 당 16바이트를 제공합니다. 이 옵션은 온라인 보기 및 80문자 워크스테이션 프린터에서 화면을 인쇄할 때 유용합니다. 표시 옵션을 선택했을 경우, Enter를 입력한 이후 16진 자료를 보기 원한다면 F6을 사용하십시오.

16진 자료는 몇 가지 다른 자료에 선행하는 FFDC 덤프 자료의 처음 1K바이트를 포함합니다. FFDC 자료의 시작은 FFDC 자료 색인에 의해 식별됩니다. 목표 작업(만일 어플리케이션 서버에서 발생할 경우)명은 자료 색인 앞에 있습니다. 만일 FFDC 덤프 스펴 파일이 삭제되지 않았다면, 완전히 규정된 작업명을 사용하여 스펴 파일을 찾으십시오. 만일 스펴 파일을 발견할 수 없다면, 다음 중 하나를 실행하십시오.

- 오류 기록부 내에 저장된 덤프 중 처음 1K를 사용하십시오.
- 만일 FFDC 자료 중 1K가 불충분하다면 문제점을 재작성하십시오.

오류 기록부에서 FFDC 자료를 해석할 때 스폴 파일 내의 자료 뿐 아니라 오류 기록부 내의 FFDC 자료도 읽을 수 있도록 포맷되지 않았습니다. 오류 기록부 내의 FFDC 덤프의 각 섹션은 4바이트 헤더의 접두부가 있습니다. 헤더의 처음 2바이트는 다음 섹션의 길이입니다(접두부를 뺀). 섹션 번호인 다음 2바이트는 색인 내의 섹션 번호에 상응합니다(277 페이지의 『FFDC 덤프 출력 설명』을 참조하십시오).

어플리케이션 서버 문제점을 진단하기 위한 서비스 작업 시작

어플리케이션이 DRDA를 사용할 때, SQL문은 어플리케이션 서버 작업 내에서 실행됩니다. 따라서 디버그 또는 OS/400 오퍼레이팅 시스템 상에 실행되는 어플리케이션 서버 작업에 대한 작업 추적을 시작할 필요가 있을 수도 있습니다. 이를 수행하기 위한 기술은 APPC를 사용하는냐 또는 TCP/IP를 사용하는냐에 따라 다릅니다. 서버 문제점을 진단하기 위한 서비스 작업 시작에 대한 자세한 정보는 다음 주제를 참조하십시오.

- APPC 서버에 대한 서비스 작업
- 사용자 자신의 TPN 작성 및 QCNTRVC 설정
- TCP/IP 서버에 대한 서비스 작업
- QRWOPTIONS 자료 영역 사용법

APPC 서버에 대한 서비스 작업

iSeries용 DB2 UDB 어플리케이션 서버가 특별한 트랜잭션 프로그램명(TPN)을 인식할 때, 어플리케이션 서버는 시스템 오퍼레이터에게 메시지를 송신하고 응답에 대해 대기합니다(1을 참조하십시오). 자세한 정보는 사용자 자신의 TPN 작성 및 QCNTRVC 설정을 참조하십시오. 이 대기 상태에 사용자는 어플리케이션 서버 작업을 위한 작업 추적 또는 디버그 시작을 허용하는 서비스 작업 시작(STRSRVJOB) 명령을 발행할 수 있습니다. 다음 단계를 통해 iSeries용 DB2 UDB 어플리케이션 서버 작업을 중단하고 디버그 모드에서 재시작할 수 있습니다.

1. QCNTRVC를 어플리케이션 리퀘스터의 트랜잭션 프로그램명(TPN)으로 지정하십시오. 이 지정 방법은 각 플랫폼에 따라 다릅니다. 다음 섹션은 그 다른 방법을 설명한 것입니다.
2. OS/400 어플리케이션이 QCNTRVC의 TPN을 수신할 때, CPF9188 메시지를 QSYSOPR로 송신하고 G(for go) 응답에 대기합니다.
3. G 응답을 입력하기 전에 서비스 작업 시작(STRSRVJOB) 명령을 사용하여 어플리케이션 서버 작업을 위한 서비스 작업을 시작하십시오. 그리고 디버그 모드에서 응답을 송신하십시오. (작업명을 표시하는 CPF9188 메시지에 대한 도움말을 요구하십시오.)
4. 디버그 시작(STRDBG) 명령을 입력하십시오.
5. 어플리케이션 서버 작업에 대한 디버그를 시작한 후 G로 QSYSOPR에 대해 응답하십시오.
6. G 응답을 수신한 후, 어플리케이션 서버는 정상 DRDA 처리를 계속합니다.
7. 어플리케이션 실행 후, SQL 디버그 메시지를 찾기 위해 어플리케이션 서버 작업 기록부를 참조할 수 있습니다.

사용자 자신의 TPN 작성 및 QCNTRSVC 설정

iSeries용 DB2 UDB 어플리케이션 리퀘스터 상에 TPN으로서 QCNTRSVC 설정

관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 또는 관계형 데이터베이스 디렉토리 항목 변경(CHGRDBDIRE) 명령의 TNSPGM 매개변수에서 QCNTRSVC를 지정하십시오.

디버깅을 종료할 때 설정을 다시 되돌릴 것을 기억하기 쉽게 하기 위해서 RDB 디렉토리 항목의 텍스트 내에 특별한 TPN에 대한 주를 만들어 두는 것이 좋습니다.

iSeries용 DB2 UDB 어플리케이션 서버(AS) 작업 디버깅을 위한 고유의 TPN 작성

종료할 때 디버그 명령문과 TFRCTL QSYS/QCNTEDDM 명령문을 수록한 CL 프로그램을 컴파일하여 자체 TPN을 작성할 수 있습니다. 이 작업으로 누릴 수 있는 장점으로는 연결할 때 수동으로 간섭할 필요가 전혀 없어진다는 것입니다. 프로그램의 예는 다음과 같습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```
PGM
  MONMSG CPF0000
    STRDBG UPDPROD(*YES) PGM(CALL/QRWTEXEC) MAXTRC(9999)
    ADBBKP STMT(CKUPDATE) PGMVAR((*CHAR (SQLDA@))) OUTFMT(*HEX) +
      LEN(1400)
    ADDTRC PGMVAR((DSLENGTH ( )) (LNTH ( )) (FDODTA_LNTH ( )))
    TRCJOB *ON TRCTYPE(*DATA) MAXSTG(2048) TRCFULL(*STOPTRC)
    TFRCTL QSYS/QCNTEDDM
ENDPGM
```

어플리케이션 리퀘스터(AR)의 RDB 디렉토리 항목 내의 TPN명은 사용자가 공급한 이름입니다. 텍스트 필드를 사용하여 특별한 TPN을 사용하고 있다는 경고를 삽입하십시오. 그리고 디버깅이 끝난 후 반드시 TPN명을 변경해야 합니다.

RDB의 TPN을 변경할 때, 다시 변경값을 되돌릴 때까지는 어플리케이션 리퀘스터로부터의 모든 연결이 새로운 TPN을 사용하여 발생한다는 사실을 명심해야 합니다. 이 설정은 성능 미달, 오퍼레이터 응답 장기 대기 및 디버그 자료로 가득 찬 기억장치 등 사용자가 예상치 못할 사태를 초래할 수도 있습니다.

VM 어플리케이션 리퀘스터를 위한 DB2 UDB 상에 TPN으로서 QCNTRSVC 설정

TPN 태그 내에 QCNTRSVC를 지정하기 위해 UCOMDIR NAMES 파일을 변경하십시오.

예를 들면, 다음과 같습니다.

```
:nick.RCHASLAI :tpn.QCNTRSVC
                :luname.VM4GATE RCHASLAI
                :modename.MODE645
                :security.NONE
```

그런 다음 SET COMDIR RELOAD USER를 발행하십시오.

z/OS 어플리케이션 리퀘스터를 위한 DB2 UDB 상에 TPN으로서 QCNTSRVC 설정

SYSIBM LOCATIONS 표를 갱신하여 iSeries용 DB2 UDB 어플리케이션 서버의 RDB-NAME을 수록하고 있는 행에 대한 TPN 열에 QCNTSRVC를 지정하십시오. 릴리스 5 이전의 시스템 실행 버전일 경우, 위의 지시대로 SYSIBM.SYSLOCATIONS 표와 LINKATTR 열을 대체하십시오.

DB2 연결 어플리케이션 리퀘스터 상에서 TPN으로서 QCNTSRVC 설정

DB2 Connect 및 Universal Database에 작업하고 있고 이 제품군에 대해 TPN을 설정하는 방법에 관한 지침을 알고 싶으면 지식 베이스: Windows, OS/2®, UNIX®용 DB2 Universal Database 및 DB2 Connect 웹 페이지를 참조하십시오. 여러 다른 버전별로 몇가지 책을 구할 수 있습니다(그러나 이 매뉴얼에 설명된 모든 기능이 모든 버전에서 지원되는 것은 아니라는 점을 주의하십시오).

TCP/IP 서버에 대한 서비스 작업

DDM TCP/IP 서버에서는 APPC 서버에서처럼 TPN을 사용하지 않습니다. 하지만, TCP/IP 서버에 의한 사전시작 작업을 사용하여 환경 내에서 서비스 작업을 시작할 수 있습니다. 그러나 QRWOPTIONS 자료 영역 사용법과 연관된 기능의 도입으로 많은 경우 서비스 작업을 시작할 필요가 없다는 점을 주의하십시오. 그 특성은 추적을 시작하고 다른 진단 기능을 수행할 수 있도록 해 줍니다. 작업의 연결 단계를 추적해야 할 경우에도 서비스 작업을 시작해야 합니다.

다음 문장이 참일 경우, 지정된 연결에 사용되고 있는 서버 작업의 이름을 보고하는 CPI3E34 메시지를 찾기 위해 기록부 표시(DSPLOG) 명령을 사용할 수 있습니다.

- 연결 조작을 하는 동안 서버의 조치를 추적할 필요가 없습니다.
- QRWOPTIONS 함수를 사용하지 않기로 합니다.
- 대화식 SQL에서처럼 서버에서 약간의 설정을 할 수 있을 때까지 어플리케이션 리퀘스터(AR) 작업의 실행을 지연시키는 능력이 있습니다.

그 후에 이전 섹션 내에 설명된 대로 서비스 작업 시작(STRSRVJOB) 명령을 사용할 수 있습니다.

만일 연결 명령문을 추적할 필요가 없거나, 또는 연결 이후 서버 상에 수동 설정을 위한 시간이 없다면, 연결이 발생하기 전에 연결을 위해 사용될 사전시작 작업이 무엇인지 예상해야 합니다. 예상을 위한 한 가지 방법은 가능하다면 테스트 시간 동안 다른 사용자가 연결하지 않도록 조치하고 하나를 제외하고 사전시작 작업 모두를 종료하면 됩니다.

QSYSWRK 내에서 실행되는 QRWTSRVR을 위한 사전시작 작업 항목 변경(CHGPIE) 명령에서 다음 매개변수를 다음과 같이 지정된 값으로 설정하면 사전시작 작업 수가 하나가 되도록 설정할 수 있습니다.

- 초기 작업 수: 1
- 임계값(threshold): 1
- 추가 작업 수: 0
- 최대 작업 수: 1

만일 이 방법을 사용하는 경우, 매개변수를 사용 환경에 합당한 값으로 반드시 되돌려야 합니다. 그렇지 않으면 사용자는 사전시작 작업 진행 중 연결을 시도할 때 '리모트 소켓 연결이 소켓에 의해 재설정되었습니다.'라는 메시지를 수신받습니다.

QRWOPTIONS 자료 영역 사용법

QRWOPTIONS 자료 영역

DDM/DRDA TCP/IP 서버 작업이 시작되면 사용자가 진단 및 다른 옵션을 지정할 수 있는 자료 영역을 찾습니다. 이름은 QRWOPTIONS이며 적용하려면 QGPL 라이브러리에 상주해야 합니다. 48자 스트링으로 이루어집니다.

주: 자료 영역에 있는 정보는 CCSID 37 또는 500에서 대문자로 입력되어야 합니다.

자료 영역의 형식은 다음과 같습니다.

표 13. 자료 영역 형식

열	목적
1-15	'I'가 전환 값으로 지정되었을 때(그렇지 않을 경우 무시됨) 사용할 점분할 십진 형식의 클라이언트 IP 주소입니다.
16	서버가 무시하는 예약된 영역(사용자를 위한 용도의 문자를 포함할 수 있음)
17-26	'U'가 전환 값으로 지정된 경우(그렇지 않을 경우 무시됨) 비교를 위한 사용자 프로파일(User profile)명입니다.
27	'A', 'I' 또는 'U'로 설정된 경우, 작업 기록부가 보관되도록 전환합니다(주 1 및 2 참조).
28	'A', 'I' 또는 'U'로 설정된 경우, DSPJOB 출력이 인쇄되도록 전환합니다(주 1 및 2 참조).
29	'A', 'I' 또는 'U'로 설정된 경우, 작업이 추적되도록 전환합니다(주 1 및 2 참조).
30	'A', 'I' 또는 'U'로 설정된 경우, 작업을 위해 디버그가 시작되도록 전환합니다(주 1 참조).
31	'A', 'I' 또는 'U'로 설정된 경우, QRYOPTLIB 값으로 조회 속성 변경 (CHGQRYA) 명령을 발생하도록 전환합니다. QRYOPTLIB 값은 해당 QAQQINI 파일이 들어 있는 라이브러리의 이름을 포함하는 39-48열에서 추출됩니다(주 1 참조). 주: 이 열에 'I' 또는 'A'가 지정되어 있으면, QUSER는 *JOBCTL 특수 권한이 있어야 효력을 발생할 수 있습니다.
32	'A', 'I' 또는 'U'로 설정된 경우, 클라이언트 디버그 옵션을 새도우하도록 전환합니다(주 1 참조).
33	'T'로 설정된 경우, 작업 추적을 위해 새 STRTRC 대신 오래된 TRCJOB을 사용하도록 전환하고 29열이 추적을 요구합니다. 주: 이 열이 'T'로 설정되어 있으면, TRCJOB은 작업 추적을 위해 사용됩니다. STRTRC를 사용하려면 공백이나 'S'로 설정하십시오.
34	'N'으로 설정하면 CPI3E34 메시지를 이력 기록부에 감추게 됩니다(V5R1에서는 PTF SI02613이 있어야 사용할 수 있습니다.)
35	'A', 'I' 또는 'U'로 설정된 경우, 특별한 서버루틴 추적을 시작하도록 전환합니다(주 1 및 2 참조).
36-38	예약
39-48	일반 자료 영역(조회 속성 변경(CHGQRYA) 명령이 31열에서 적절한 값으로 트리거되면 라이브러리명을 포함합니다.)

주:

- 이는 이 값이 나타나는 열에 해당되는 기능을 활성화하는 전환 값입니다.
 - ‘A’는 서버 작업을 모두 사용하기 위한 기능을 활성화합니다.
 - ‘T’는 1-15열에 지정된 클라이언트 IP 주소가 연결을 시도할 때 사용된 주소와 일치하면 기능을 활성화합니다.
 - ‘U’는 17-28열에 지정된 사용자 ID가 연결을 시도할 때 사용된 것과 일치하면 기능을 활성화합니다.
- 이 기능의 결과로 생긴 스푼 파일을 찾으려면 작업에 대한 작업 명령(WRKJOB 사용자 프로파일/QPRTJOB)을 사용하십시오. 여기서 사용자 프로파일은 연결 요구 시 사용된 사용자 ID입니다. 옵션 4를 선택하면 이 파일을 하나 이상 참조해야 합니다.

표 14. WRKJOB 사용자 프로파일/QPRTJOB 명령의 파일 리스트

파일	장치 또는 대기행렬	사용자 자료
QPJOBLOG	QEZJOBLOG	QRWTSRVR
QPDSPJOB	PRT01	
QPSRVTRC	PRT01	

- 특수 DRDA 서브루틴 추적이 들어 있는 파일이 QGPL 라이브러리에 QRWDBmmddy 형식의 이름으로 작성됩니다. 여기서 mm은 추적이 기록된 월, dd는 일, y는 연도의 마지막 숫자입니다. 모든 서버 프로그램이 추적되는 것은 아닙니다.

자세한 정보는 다음 예를 참조하십시오.

예: 자료 영역을 작성하는 CL 명령

```
CRTDTAARA DTAARA(QGPL/QRWOPTIONS) TYPE(*CHAR) LEN(48)
          VALUE('9.5.114.107 :MYUSERID AAUIU TN INILIBRARY')
          TEXT('DRDA TCP SERVER DIAGNOSTIC OPTIONS')
```

이 예는 표 15에 나타난 기능을 요구합니다.

주: 예에 나타난 적절한 띄어쓰기가 중요하기 때문에 VALUE 매개변수의 내용이 표 양식으로 반복됩니다.

표 15. CRTDTAARA 예의 VALUE 매개변수의 자료 요소에 관한 설명

열	목차	설명
1-11	9.5.114.107	30열에서 전환할 수 있는 사용 가능한 IP 주소
16	:	IP 주소 필드의 끝을 나타내는 문자
17-24	MYUSERID	29열과 31열에서 전환할 수 있는 사용 가능한 사용자 ID
27	A	서버 작업 기록부(QRWTSRVR용)를 모든 사용자에게 대해 스푼합니다.
28	A	DSPJOB 출력을 서버의 모든 사용에 대해 스푼합니다.
29	U	연결 요구 시 사용자 ID가 자료 영역의 17-26열(이 예에서 'MYUSERID')에 지정된 것과 일치하면, 명령(TRCJOB)으로 작업을 추적합니다.
30	I	클라이언트 IP 주소(이 예에서는 '9.5.114.107')와 자료 영역의 1-15열에 지정된 것과 일치하면 디버그 시작(STRDBG) 명령(아무런 프로그램도 지정하지 않음)으로 디버그를 시작합니다.

표 15. CRTDTAARA 예의 VALUE 매개변수의 자료 요소에 관한 설명 (계속)

열	목차	설명
31	U	연결 요구 시 사용자 ID가 자료 영역의 17-26열(이 예에서 'MYUSERID')에 지정된 것과 일치하면, 조회 속성 변경(CHGQRYA) QRYOPTLIB(INILIBRARY) 명령을 호출합니다. 주: 라이브러리명은 자료 영역의 39-48열에서 취할 수 있습니다.
32		클라이언트 디버그 옵션을 서버로 새도우하지 않습니다.
33	T	작업 추적을 위해 오래된 TRCJOB 기능을 사용합니다.
34	N	이력 기록부에 CPI3E34 메시지를 남기지 않습니다.
35		서브루틴 추적을 시작하지 않음
39-48	INILIBRARY	전환 31로 사용되는 라이브러리

제 10 장 분산 관계형 데이터베이스(DRDB) 어플리케이션 쓰기

프로그래머는 iSeries 분산 어플리케이션 프로그램에 SQL문을 사용하는 고급 언어 프로그램을 작성할 수 있습니다. 로컬 처리를 위해 작성한 프로그램의 기본 차이점은 리모트 데이터베이스에 연결하여 SQL 패키지를 작성하는 기능입니다. 어플리케이션 리퀘스터를 어플리케이션 서버에 연결하는 데 CONNECT SQL문을 사용하거나 프로그램이 내재적 연결을 허용하도록 작성될 때 관계형 데이터베이스의 이름을 지정할 수 있습니다. 또한 분산 작업 단위를 사용하는 어플리케이션의 연결을 관리하는 데 SET CONNECTION, RELEASE 및 DISCONNECT 명령문을 사용할 수 있습니다.

SQL 패키지는 분산 관계형 데이터베이스에만 사용되는 iSeries 오브젝트입니다. SQL의 사전컴파일 프로세스의 결과로 작성되거나 컴파일된 프로그램 오브젝트로부터 작성될 수 있습니다. SQL 패키지는 어플리케이션 서버에 있습니다. 어플리케이션 서버에는 SQL문, 호스트 변수 속성 및 어플리케이션 서버가 어플리케이션 리퀘스터의 요구를 처리하는 데 사용하는 액세스 계획이 들어 있습니다.

어플리케이션 프로그램은 여러 다른 서버에 연결할 수 있기 때문에 프로그래머는 서버 간의 자료 변환에 더 주의를 기울여야 합니다. iSeries 서버는 문자 정보의 관리를 위한 코드화 문자 세트 ID(CCSID) 지원을 비롯하여 여러 종류의 자료 변환을 제공합니다.

로컬 처리 어플리케이션에 사용하는 것과 같은 방법으로 SQL 언어를 사용하여 iSeries 서버에 분산 관계형 데이터베이스(RDB) 프로그램을 작성하고 유지보수할 수 있습니다. 또한 다음 중 하나 이상의 고급 언어를 사용하여 정적 및 동적 SQL문을 삽입할 수 있습니다.

- iSeries PL/I
- ILE C/400*
- COBOL/400
- ILE COBOL/400
- FORTRAN/400*
- RPG/400®
- ILE RPG/400

분산 어플리케이션 개발 프로세스는 로컬 처리 SQL 어플리케이션 개발 프로세스와 유사합니다. 차이점은 분산 처리용 어플리케이션은 연결하고 있는 관계형 데이터베이스의 이름을 지정해야 한다는 것입니다. 이름 지정은 프로그램을 사전컴파일할 때나 어플리케이션 내에서 실행됩니다.

오브젝트 하나만 제외하고 로컬 및 분산 어플리케이션 모두에 사용된 동일한 SQL 오브젝트만 분산 관계형 데이터베이스(RDB) 지원에 사용됩니다. CRTSQLxxx(SQL 프로그램 작성) 명령을 사용하여 프로그램을 작성합니다. 이 명령에서 xxx는 호스트 언어 CI, CBL, CBLI, FTN, PLI, RPG 또는 RPGI를 나타냅니다. SQL 패키지는 이 프로세스에서 사전컴파일 제품일 수 있습니다. SQL 패키지 작성(CRTSQLPKG) 명령은 기존 분산 SQL 프로그램의 SQL 패키지를 작성합니다.

SQL문이 있는 프로그램을 사전컴파일하려면 DB2 UDB 조회 관리자 및 SQL 개발 킷 사용권 프로그램이 설치되어 있어야 합니다. 그러나 서버에 컴파일된 프로그램만 설치되어 있는 기존 분산 SQL 프로그램으로부터 SQL 패키지를 작성할 수 있습니다. DB2 UDB 조회 관리자 및 SQL 개발 킷 사용권 프로그램을 사용하면 대화식 SQL 프로그램을 사용하여 분산 관계형 데이터베이스(RDB)에 액세스할 수 있습니다. 이 기능은 프로그램을 사전컴파일하거나 컴파일하지 않고 SQL문을 테스트할 수 있기 때문에 프로그램을 디버깅할 때 도움이 됩니다.

이 장에서는 분산 관계형 데이터베이스(RDB)의 프로그래밍 문제의 개요를 설명합니다. 다음 주제에 관한 자세한 정보는 iSeries Information Center의 SQL 프로그래밍 개념 주제를 참조하십시오.

- 분산 관계형 데이터베이스(DRDB) 어플리케이션에 대한 프로그래밍 고려사항
- 분산 관계형 데이터베이스 프로그램 작성
- SQL 패키지에 대한 작업

분산 관계형 데이터베이스(DRDB) 어플리케이션에 대한 프로그래밍 고려사항

iSeries 서버의 분산 관계형 데이터베이스(RDB) 어플리케이션 프로그래밍 고려사항은 로컬 서버에서 지원되는 기능을 다루는 범주와 다른 서버에 연결한 결과의 범주 등 두 가지 기본 범주로 분리됩니다. 이 섹션에서는 다음 사항을 설명하면서 이 두 가지 범주를 모두 설명합니다.

- 명명 규칙
- 다른 서버에 연결
- 분산 SQL문과 공존
- DRDA 작업 단위 종료
- 코드화 문자 세트 ID(CCSID)
- 자료 해석
- 분산 자료 관리(DDM) 파일과 SQL 프로그램

18 페이지의 『추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계』는 분산 관계형 데이터베이스 어플리케이션을 설계할 때 고려해야 할 추가 정보를 제공합니다.

| DRDA 프로토콜의 새로운 레벨

| IBM DB2 제품군의 다른 멤버는 버전 8 릴리스에서 새 레벨의 프로토콜과 자료 구조(DRDA 레벨 5)가 필요한 새 기능을 도입했습니다. OS/400 V5R3은 새로 구조화된 기능의 서브세트를 사용하고 호환성을 유지하기 위해 레벨 5에서 작동하도록 설계했습니다. OS/400 V5R3에 포함된 기능은 다음과 같습니다.

- | • 사용자에게 보다 친숙한 Kerberos 구성. Kerberos 구성에 대한 자세한 정보는 네트워크 인증 서비스 주제를 참조하십시오.
- | • 실행 준비와 동시에 SQL문의 속성을 지정할 수 있음. SQL 참조서 주제에서 PREPARE문을 참조하십시오.

- | • SQL문 완료 시 사용할 수 있는 새롭고 완전한 진단 정보. 이 자료 콜렉션은 기존 SQLCA 제어 블록을 대체하지만 함께 사용할 수도 있습니다. 모든 DB2 제품군 멤버가 추가된 정보를 초기에 제공하는 것은 아닙니다. 자세한 정보는 SQL 참조서에서 GET DIAGNOSTICS문을 참조하십시오.
- | • 저장 프로시저가 조회 결과 세트를 리턴하면 후속 호출에서 조회의 새 인스턴스를 열 때 한 프로시저 호출의 조회가 열린 상태를 유지할 수 있습니다. 자세한 정보는 SQL 프로그래밍 개념 주제에서 DRDA 저장 프로시저 고려사항을 참조하십시오.
- | • 2단계 요약 프로세스를 수행할 때 조회 도중 트리거와 기능에 의해 수행된 데이터베이스 갱신이 고려됩니다. 자세한 정보는 SQL 프로그래밍 개념 주제에서 분산 지원을 참조하십시오.
- | • 제한된 이중 화면이동 커서 기능 세트는 iSeries가 어플리케이션 서버 역할을 할 때 지원됩니다(읽기 전용 액세스). LOB 필드(예: BLOB 및 CLOB)의 검색은 화면이동 커서를 지원하지 않습니다. 자세한 정보는 SQL 프로그래밍 개념 주제에서 화면이동 커서를 참조하십시오.

분산 관계형 데이터베이스 오브젝트 명명

SQL 오브젝트는 iSeries 서버 오브젝트로 작성되고 유지보수됩니다.

iSeries용 DB2 Universal Database 프로그래밍의 두 가지 명명 규칙인 시스템(*SYS)과 SQL(*SQL) 중 하나를 사용할 수 있습니다. 사용하는 명명 규칙은 파일과 표 이름을 규정하는 방법에 영향을 줍니다. 또한 대화식 SQL 표시장치에 사용되는 보안과 용어에도 영향을 줍니다. 분산 관계형 데이터베이스 어플리케이션은 명명 규칙을 사용하여 다른 iSeries 서버의 오브젝트에 액세스할 수 있습니다. 그러나, 프로그램이 비iSeries 서버의 관계형 데이터베이스에 액세스하면 SQL 이름만 사용될 수 있습니다. STRSQL(SQL 시작) 명령의 NAMING 매개변수나 CRTSQLxxx 명령의 하나인 OPTION 매개변수를 사용하여 명명 규칙을 선택하십시오.

시스템(*SYS) 명명 규칙

시스템 명명 규칙을 사용할 때 파일은 *library/file* 형식으로 라이브러리 이름에 의해 규정됩니다. 이 명명 규칙으로 작성된 표는 작성되어 있는 라이브러리의 공용 권한을 가정합니다. 표 이름이 명시적으로 규정되어 있지 않고 디폴트 콜렉션 명이 CRTSQLxxx 또는 CRTSLQPKG 명령의 DFTRDBCOL 매개변수로 사용되면 디폴트 콜렉션 명은 정적 SQL문에 사용됩니다. 파일 이름이 명시적으로 규정되지 않고 디폴트 콜렉션 명이 지정되지 않으면 다음 규칙이 적용됩니다.

- 특정 CREATE 명령문을 제외한 모든 SQL문은 SQL이 규정되지 않은 파일의 라이브러리 리스트(*LIBL)를 검색하도록 합니다.
- CREATE 명령문은 규정되지 않은 오브젝트를 다음과 같이 해결합니다.
 - CREATE TABLE: 표 이름은 명시적으로 규정되어야 합니다.
 - CREATE VIEW: 뷰는 부속 선택에서 참조된 첫 번째 라이브러리에서 작성됩니다.
 - CREATE INDEX: 색인은 색인을 빌드 중인 표가 있는 콜렉션이나 라이브러리에서 작성됩니다.

SQL(*SQL) 명명 규칙

SQL 명명 규칙을 사용할 때 표는 *collection.table* 형식으로 콜렉션 명에 의해 규정됩니다. 표 이름이 명시적으로 규정되지 않고 디폴트 콜렉션 명이 CRTSQLxxx 또는 SQL 패키지 작성(CRTSQLPKG) 명령의 디폴트

관계형 데이터베이스 콜렉션(DFTRDBCOL) 매개변수에 지정되면 디폴트 콜렉션 명이 사용됩니다. 표 이름이 명시적으로 규정되지 않고 디폴트 콜렉션 명이 지정되지 않으면 다음 규칙이 적용됩니다.

- 정적 SQL의 경우, 디폴트 규정자는 프로그램 소유자의 사용자 프로파일입니다.
- 동적 SQL 또는 대화식 SQL의 경우, 디폴트 규정자는 명령문을 실행 중인 작업의 사용자 프로파일입니다.

디폴트 콜렉션명

프로그램을 사전컴파일할 때 CRTSQLxxx 명령의 DFTRDBCOL 매개변수에 이 이름을 제공하여 SQL 프로그램이 사용할 디폴트 콜렉션명을 지정할 수 있습니다. DFTRDBCOL 매개변수는 *SYS 명명 규칙이 사용된 경우에는 규정되지 않은 파일의 라이브러리, 또는 *SQL 명명 규칙이 사용된 경우에는 규정되지 않은 표의 콜렉션으로 프로그램에 콜렉션 명을 제공합니다. 프로그램을 사전컴파일할 때 디폴트 콜렉션 명을 지정하지 않으면 위에 설명한 대로 각 명명 규칙에 규정되지 않은 이름의 규칙이 적용됩니다. 디폴트 관계형 데이터베이스 콜렉션 명만 정적 SQL문에 적용됩니다.

또한 SQL 패키지 작성(CRTSQLPKG) 명령의 DFTRDBCOL 매개변수를 사용하여 패키지의 디폴트 콜렉션을 변경할 수 있습니다. SQL 프로그램을 컴파일한 후에 새 SQL 패키지를 작성하여 디폴트 콜렉션을 변경할 수 있습니다. (CRTSQLPKG) 명령의 모든 매개변수에 대한 설명은 220 페이지의 『SQL 패키지 작성 (CRTSQLPKG) 명령 사용』을 참조하십시오.

분산 관계형 데이터베이스(DRDB)에 연결

분산 관계형 데이터베이스 어플리케이션이 분산하는 것은 다른 서버의 관계형 데이터베이스에 연결하는 기능입니다.

구문은 같지만 다른 의미를 갖는 두 유형의 CONNECT 명령문이 있습니다.

- CONNECT(유형 1)는 리모트 작업 단위에 사용됩니다.
- CONNECT(유형 2)는 분산 작업 단위에 사용됩니다.

프로그램이 사용하는 CONNECT 유형은 CRTSQLxxx 명령의 RDBCNNMTH 매개변수로 표시됩니다.

DRDA 리모트 작업 단위

리모트 작업 단위 기능은 리모트 준비와 SQL문의 실행을 제공합니다. 컴퓨터 서버 A의 활성 그룹은 컴퓨터 서버 B의 어플리케이션 서버에 연결할 수 있습니다. 그리고 하나 이상의 작업 단위 내에서 해당 활성 그룹은 B에 있는 오브젝트를 참조하는 많은 수의 정적 또는 동적 SQL문을 실행할 수 있습니다. B의 작업 단위를 종료한 후에 활성 그룹은 컴퓨터 서버 C의 어플리케이션 서버에 연결할 수 있습니다.

대부분의 SQL문에는 다음 제한사항이 있고 리모트로 준비되고 실행될 수 있습니다.

- 단일 SQL문에서 참조한 모든 오브젝트는 같은 어플리케이션 서버에 의해 관리되어야 합니다.
- 작업 단위에 있는 모든 SQL문은 같은 어플리케이션 서버에 의해 실행되어야 합니다.

DRDA 리모트 작업 단위 연결 관리: 활성 그룹의 상태는 항상 다음 세 가지 중 하나입니다.

연결 가능 및 연결된 상태

연결 불가능 및 연결된 상태

연결 가능 및 연결되지 않은 상태

다음 다이어그램은 상태 트랜지션입니다.

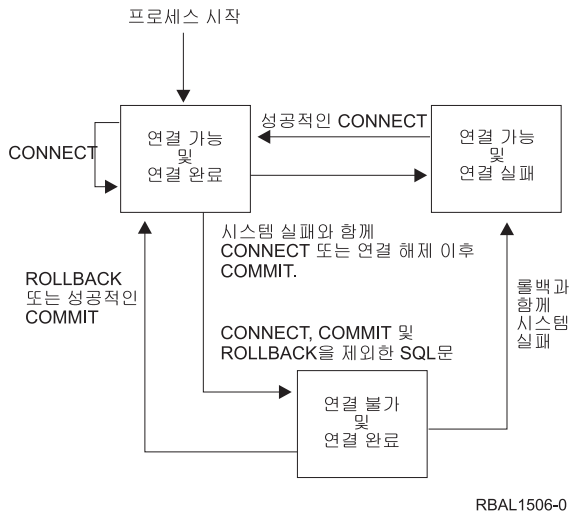


그림 19. 리모트 작업 단위 활성 그룹 연결 상태 트랜지션

활성 그룹의 초기 상태는 연결 가능과 연결된 상태입니다. 활성 그룹이 연결된 어플리케이션 서버는 CRTSQLxxx와 STRSQL 명령의 RDB 매개변수에 의해 판별되며 내재 CONNECT 조작을 포함할 수 있습니다. 내재 CONNECT 조작은 내재 또는 명시 CONNECT 조작이 이미 성공했거나 아직 성공하지 못했을 경우에는 발생하지 않습니다. 따라서 활성 그룹은 어플리케이션 서버에 한 번 이상 내재적으로 연결될 수 없습니다.

연결 가능 및 연결된 상태: 활성 그룹은 어플리케이션 서버에 연결되며 CONNECT 명령문이 실행될 수 있습니다. 활성 그룹은 연결 불가능과 연결 상태에서부터 롤백이나 성공적인 확약을 완료할 때 또는 CONNECT 명령문이 연결 가능 및 연결되지 않은 상태에서부터 성공적으로 실행될 때 이 상태가 됩니다.

연결 불가능 및 연결된 상태: 활성 그룹은 어플리케이션 서버에 연결되지만 CONNECT 명령문을 성공적으로 실행하여 어플리케이션 서버를 변경할 수 없습니다. 활성 그룹은 CONNECT, COMMIT 또는 ROLLBACK 이외의 SQL문을 실행할 때 연결 가능 및 연결 상태에서 이 상태가 됩니다.

연결 가능 및 연결되지 않은 상태: 활성 그룹은 어플리케이션 서버에 연결되지 않습니다. 실행될 수 있는 유일한 SQL문은 CONNECT입니다.

활성 그룹은 다음 경우에 이 상태가 됩니다.

- 연결이 이전에 릴리스되었고 COMMIT이 성공적으로 실행된 경우.
- SQL DISCONNECT 명령문을 사용하여 연결을 단절한 경우.
- 연결이 연결 가능 상태에 있었지만 CONNECT 명령문이 성공하지 못한 경우.

연속 CONNECT 명령문은 CONNECT가 연결 가능 상태에서 활성 그룹을 제거하지 않기 때문에 성공적으로 실행될 수 없습니다. 활성 그룹이 현재 연결되어 있는 어플리케이션 서버의 CONNECT는 다른 CONNECT 명령문처럼 실행됩니다.

CONNECT, COMMIT, DISCONNECT, SET CONNECTION, RELEASE 또는 ROLLBACK(COMMIT (*NONE)으로 실행하지 않는 한) 이외의 SQL문이 선행되지 않으면 CONNECT는 성공적으로 실행될 수 없습니다. 오류를 피하려면 CONNECT 명령문을 실행하기 전에 확약 또는 롤백 작업을 실행하십시오.

어플리케이션 지향 분산 작업 단위

어플리케이션 지향의 분산 작업 단위 기능은 리모트 작업 단위와 같은 방식으로 SQL문의 리모트 준비와 실행을 제공합니다. 리모트 작업 단위와 마찬가지로, 컴퓨터 서버 A의 활성 그룹은 컴퓨터 서버 B의 어플리케이션 서버에 연결할 수 있으며 작업 단위를 종료하기 전에 B의 오브젝트를 참조하는 많은 정적 또는 동적 SQL문을 실행할 수 있습니다. 단일 SQL문에서 참조한 모든 오브젝트는 같은 어플리케이션 서버에 의해 관리되어야 합니다. 그러나, 리모트 작업 단위와는 달리 많은 어플리케이션 서버가 같은 작업 단위에 참여할 수 있습니다. 확약이나 롤백 작업이 작업 단위를 종료합니다.

어플리케이션 지향 분산 작업 단위 연결 관리: 언제든지

- 활성 그룹은 항상 연결 또는 연결되지 않은 상태에 있으며 0 이상의 연결 세트를 갖고 있습니다. 각 활성 그룹의 연결은 연결의 어플리케이션 서버 이름에 의해 고유하게 식별됩니다.
- SQL 연결 상태는 항상 다음 중 하나입니다.
 - 현재 및 보류
 - 현재 및 해제
 - 정지 및 보류
 - 정지 및 해제

활성 그룹의 초기 상태: 활성 그룹은 초기에는 연결 상태에 있고 정확히 하나의 연결을 갖습니다. 연결의 초기 상태는 현재 및 보류입니다.

다음 다이어그램은 상태 트랜지션입니다.

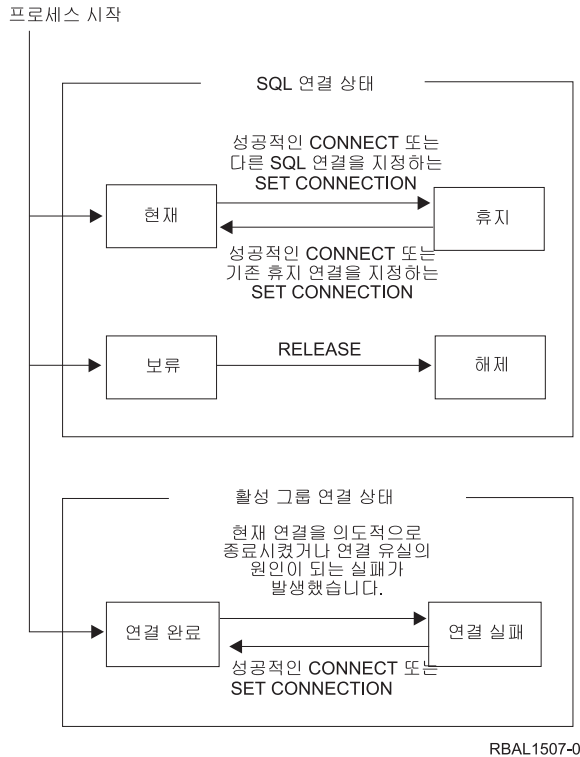


그림 20. 어플리케이션 지향의 분산 작업 단위 연결 및 활성화 그룹 연결 상태 트랜지션

연결 상태: 어플리케이션이 CONNECT 명령문을 실행하고 서버 이름이 어플리케이션 리퀘스터에게 알려져 있으며 활성화 그룹의 기존 연결 세트에 있지 않으면

- 현재 연결은 정지 상태와 보류 상태가 됩니다.
- 서버 이름이 연결 세트에 추가되고 새 연결이 현재 및 보류 상태가 됩니다.

서버 이름이 이미 활성화 그룹의 기존 연결 세트에 있으면 오류가 발생합니다.

SET CONNECTION 명령문을 사용하여 정지 상태의 연결을 현재 상태로 합니다. 연결이 현재 상태가 되면 이전 현재 연결이 있을 경우, 정지 상태가 됩니다. 활성화 그룹의 기존 연결 세트에 있는 하나의 연결만 언제든지 현재 상태가 될 수 있습니다. 현재에서 정지로 또는 정지에서 현재로 연결 상태를 변경해도 보류나 해제 상태에는 영향을 주지 않습니다.

연결은 RELEASE 명령문에 의해 해제 상태가 됩니다. 활성화 그룹이 예약 작업을 실행하면 활성화 그룹의 모든 해제 연결이 종료됩니다. 보류에서 해제로 연결 상태를 변경해도 현재나 정지 상태에는 영향을 주지 않습니다. 따라서 해제 상태의 연결은 다음 예약 작업 때까지 사용될 수 있습니다. 연결 상태를 해제에서 보류로 변경하는 방법은 없습니다.

활성 그룹 연결 상태: CONNECT 명령문의 명시적 또는 내재적 실행으로 다른 어플리케이션 서버를 빌드하는 데 다음 규칙이 적용됩니다.

- 활성화 그룹은 같은 어플리케이션 서버에 동시에 하나 이상 연결될 수 없습니다.
- 활성화 그룹이 SET CONNECTION 명령문을 실행하면 지정된 위치명은 활성화 그룹 연결 세트의 기존 연결이어야 합니다.

- 활성 그룹이 CONNECT 명령문을 실행하면 지정된 서버명은 활성 그룹 연결 세트의 기존 연결이어야 합니다.

활성 그룹이 현재 연결되어 있으면 활성 그룹은 연결 상태입니다.

CURRENT SERVER 특수 레지스터에는 현재 연결의 어플리케이션 서버 이름이 있습니다. 활성 그룹은 어플리케이션 서버가 관리하는 오브젝트를 참조하는 SQL문을 실행할 수 있습니다.

연결되지 않은 상태의 활성 그룹은 CONNECT 또는 SET CONNECTION 명령문을 실행하면 연결된 상태가 됩니다.

활성 그룹이 현재 연결되어 있지 않으면 활성 그룹은 연결되지 않은 상태입니다. CURRENT SERVER 특수 레지스터 내용은 공백입니다. 실행될 수 있는 유일한 SQL문은 CONNECT, DISCONNECT, SET CONNECTION, RELEASE, COMMIT 및 ROLLBACK입니다.

연결 상태의 활성 그룹은 현재 연결이 의도적으로 종료되거나 어플리케이션 서버에서의 롤백 작업이나 연결이 끊기는 원인이 되는 장애 때문에 SQL문이 실행되지 못했을 때 연결되지 않은 상태가 됩니다. 활성 그룹이 확약 작업을 실행하고 연결이 해제 상태에 있거나 어플리케이션 프로세스가 DISCONNECT 명령문을 성공적으로 실행할 때 연결은 의도적으로 종료됩니다.

연결이 종료되는 시기: 연결이 종료되면 연결을 통해 활성 그룹이 확보한 모든 자원과 연결을 작성하고 유지 보수하는 데 사용된 모든 자원이 더 이상 할당되지 않습니다. 예를 들어, 활성 그룹이 RELEASE 명령문을 실행하면 다음 확약 작업을 실행하는 동안 연결이 종료될 때 열린 커서는 닫힙니다.

연결은 활성 그룹이 연결되지 않은 상태가 될 경우, 통신 실패의 결과로 종료될 수 있습니다. 활성 그룹이 종료될 때 활성 그룹의 모든 연결이 종료됩니다.

RUW와 DUW 연결 관리로 실행: RUW 연결 관리로 컴파일된 프로그램은 DUW 연결 관리로 컴파일된 프로그램에 의해 호출될 수 있습니다. SET CONNECTION, RELEASE 및 DISCONNECT 명령문은 활동 연결에 대한 작업의 RUW 연결 관리로 컴파일된 프로그램에 의해 사용될 수 있습니다. 그러나, DUW 연결 관리로 컴파일된 프로그램이 RUW 연결 관리로 컴파일된 프로그램을 호출하면, RUW 연결 관리로 컴파일된 프로그램에서 실행되는 CONNECT는 CONNECT의 부분으로 활성 그룹에 대해 모든 활동중인 연결의 종료를 시도합니다.

이 CONNECT는 활동 연결에 사용된 대화가 보호된 대화를 사용할 경우 실패합니다. 게다가 보호된 대화가 비활동 연결에 사용되었고 DDMCNV 작업 속성이 *KEEP이면, 사용되지 않은 DDM 대화는 RUW 연결 관리로 컴파일된 프로그램의 연결을 실패하도록 만들 수 있습니다. 이 상황을 피하려면 CONNECT를 실행하는 RUW 연결 관리로 컴파일된 프로그램을 호출하기 전에 DDMCNV(*DROP)를 실행하고 RELEASE와 COMMIT을 실행하십시오.

마찬가지로, RUW 연결 관리로 컴파일된 프로그램용 패키지를 작성한 후에 DUW 연결 관리로 컴파일된 프로그램용 패키지를 작성할 때 DUW 연결 관리로 컴파일된 프로그램용 패키지를 작성한 다음, DDMCNV(*DROP)를 실행하거나 RCLDDMCNV를 실행하십시오.

DUW 연결 관리로 컴파일된 프로그램은 RUW 연결 관리로 컴파일된 프로그램에 의해 호출될 수 있습니다. DUW 연결 관리로 컴파일된 프로그램이 CONNECT를 실행하면 RUW 연결 관리로 컴파일된 프로그램이 실행한 연결은 단절되지 않습니다. 이 연결은 DUW 연결 관리로 컴파일된 프로그램이 사용할 수 있습니다.

디폴트 활성 그룹에 대한 내재적 연결 관리

어플리케이션 리퀘스터는 어플리케이션 서버에 내재적으로 연결할 수 있습니다. 어플리케이션 리퀘스터가 디폴트 활성 그룹의 첫 번째 활성 SQL 프로그램에 의해 발행되는 SQL문을 감지하고 다음 항목이 참이면 내재적으로 연결됩니다.

- 발행되는 SQL문은 매개변수가 있는 CONNECT 명령문이 아닙니다.
- SQL은 디폴트 활성 그룹에서 활동 중이 아닙니다.

분산 프로그램의 경우, 내재적 연결은 RDB 매개변수에 지정된 관계형 데이터베이스입니다. 분산 프로그램이 아닌 경우, 내재적 연결은 로컬 관계형 데이터베이스입니다.

SQL이 활동 중이 아니면 SQL은 디폴트 활성 그룹의 활동 연결을 종료합니다. SQL은 다음 경우에 활동 중이 아닙니다.

- 어플리케이션 리퀘스터가 프로세스의 첫 번째 활동 중인 SQL 프로그램을 감지하고 다음 사항이 모두 참인 경우.
 - 지연 중인 SQL 변경이 없는 경우.
 - 보호된 대화를 사용하는 연결이 없는 경우.
 - SET TRANSACTION 명령문이 활동 중이 아닌 경우.
 - CLOSQLCSR(*ENDJOB)로 사전컴파일된 프로그램이 실행되고 있지 않는 경우.

지연 중인 변경, 보호된 대화 그리고 활동 중인 SET TRANSACTION 명령문이 있으면 SQL은 종료된 상태가 됩니다. CLOSQLCSR(*ENDJOB)로 사전컴파일된 프로그램이 실행되면 SQL은 작업이 종료될 때까지 디폴트 활성 그룹에 대해 사용 중입니다.

- SQL이 종료된 상태면 작업 단위의 종료 시. SQL 프로그램 외부에서 사용자가 COMMIT 또는 ROLLBACK 명령을 발행할 때 발생합니다.
- 작업의 종료 시.

디폴트가 아닌 활성 그룹에 대한 내재적 연결 관리

어플리케이션 리퀘스터는 어플리케이션 서버에 내재적으로 연결할 수 있습니다. 어플리케이션 리퀘스터가 활성 그룹에 대해 발행된 첫 번째 SQL문을 감지하고 매개변수가 있는 CONNECT 명령문이 아니면 내재적 연결이 발생합니다.

분산 프로그램의 경우, 내재적 연결은 RDB 매개변수에 지정된 관계형 데이터베이스로 작성됩니다. 분산 프로그램이 아닌 경우, 내재적 연결은 로컬 관계형 데이터베이스로 작성됩니다.

프로세스의 다음 부분에서 내재적 단절이 발생할 수 있습니다.

- 활성 그룹이 종료될 때 예약 제어가 사용 중이 아니면 활성 그룹 레벨 예약 제어가 사용 중이거나 작업 레벨 예약 정의가 작업 단위 경계에 있습니다.

작업 레벨 예약 제어가 사용 중이고 작업 단위 경계에 있지 않으면 SQL은 종료된 상태가 됩니다.

- SQL이 종료된 상태이고 작업 레벨 예약 정의가 예약되었거나 롤백인 경우.
- 작업의 종료 시.

다음 예제 프로그램은 분배되지 않습니다(연결이 필요하지 않습니다). 지역 수리 정보를 보고서에 수집하는 spiffy사의 지역 사무소에서 실행되는 프로그램입니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

CRTSQLxxx PGM(SPIFFY/FIXTOTAL) COMMIT(*CHG) RDB(*NONE)

```
PROC: FIXTOTAL;
.
.
.
SELECT * INTO :SERVICE      A
FROM REPAIRTOT;
EXEC SQL
COMMIT;
.
.
.
END FIXTOTAL;
```

A 명령문은 로컬 관계형 데이터베이스에서 실행됩니다.

다음 예와 같은 다른 프로그램은 캔자스 시에 있는 spiffy사의 특약점로부터 같은 정보를 수집할 수 있습니다. 이 프로그램은 내재적으로 연결되고 단절되는 분산 프로그램의 예입니다.

CRTSQLxxx PGM(SPIFFY/FIXES) COMMIT(*CHG) RDB(KC101) RDBCNMTH(*RUW)

```
PROC: FIXES;
.
.
.
EXEC SQL
SELECT * INTO :SERVICE      B
FROM SPIFFY.REPAIR1;

EXEC SQL C
COMMIT;
.
.
.
END FIXES; D
```

B 어플리케이션 서버(AS)의 내재적 연결. 명령문은 AS에서 실행됩니다.

C 작업 단위의 종료. 어플리케이션 리퀘스터(AR)는 COMMIT이 성공하면 연결 가능 및 연결 상태가 됩니다.

D SQL 프로그램 종료 시 내재적 단절.

명시적 CONNECT

CONNECT문은 어플리케이션 리퀘스터(AR)를 식별된 어플리케이션 서버(AS)에 명시적으로 연결하는 데 사용됩니다. 이 SQL문은 어플리케이션 프로그램에 삽입될 수 있고 사용자는 대화식 SQL을 사용하여 SQL문을 발행할 수 있습니다. CONNECT 명령문은 TO 또는 RESET절과 함께 사용됩니다. TO절이 있는 CONNECT 명령문을 사용하면 특정 AS 관계형 데이터베이스에 대한 연결을 지정할 수 있습니다. RESET절이 있는 CONNECT 명령문은 로컬 관계형 데이터베이스에 대한 연결을 지정합니다.

사용자가(또는 프로그램이) TO 또는 RESET절이 있는 CONNECT 명령문을 발행하면 식별된 AS는 관계형 데이터베이스 디렉토리에 설명되어야 합니다. 이 디렉토리로 작업하는 방법에 대한 자세한 정보는 75 페이지의 『관계형 데이터베이스 디렉토리 사용』을 참조하십시오. 또한 CONNECT 명령문이 성공하려면 AR이 연결 가능 상태에 있어야 합니다.

CONNECT 명령문은 사용하는 연결 관리 방법에 따라 다른 효과를 가집니다. RUW 연결 관리의 경우, CONNECT 명령문은 다음 효과를 가집니다.

- TO 또는 RESET절이 있는 CONNECT 명령문이 성공하면 다음이 발생합니다.
 - 어플리케이션 프로세스가 COMMIT HOLD 또는 ROLLBACK HOLD SQL문을 사용하여 연결 가능 상태로 되었거나 COMMIT(*NONE)를 실행 중이면, 열린 커서가 닫히고 준비된 명령문이 삭제되며 보류 자원이 이전 AS로부터 해제됩니다.
 - 어플리케이션 프로세스는 이전 AS로부터 단절되고 식별된 AS에 연결됩니다.
 - AS의 이름은 현재 서버 특수 레지스터에 있습니다.
- 오류를 리턴한 시스템 모듈을 식별하는 정보는 SQLCA(SQL 통신 영역)의 SQLERRP 필드 또는 SQL 진단 영역의 DB2_MODULE_DETECTING_ERROR에 있습니다.
- CONNECT 명령문이 어떤 이유로든 성공하지 못하면 어플리케이션은 연결 가능하지만 연결되지 않은 상태로 유지됩니다. 연결 가능하지만 연결되지 않은 상태의 어플리케이션은 CONNECT 명령문만 실행할 수 있습니다.
- 연속 CONNECT 명령문은 CONNECT가 연결 가능 상태에서 AR을 제거하지 않기 때문에 성공적으로 실행될 수 있습니다. AR이 현재 연결되어 있는 AS에 대한 CONNECT는 다른 CONNECT 명령문처럼 실행됩니다.
- CONNECT, SET CONNECTION, COMMIT, ROLLBACK, DISCONNECT 또는 RELEASE 외의 SQL문이 선행될 때 확약 제어로 실행 중이면 CONNECT 명령문은 성공적으로 실행될 수 없습니다. 오류를 피하려면 CONNECT 명령문을 실행하기 전에 COMMIT 또는 ROLLBACK 작업을 실행하십시오. 확약 제어없이 실행하면 CONNECT 명령문은 항상 허용됩니다.

DUW 연결 관리의 경우, CONNECT 명령문은 다음 효과를 가집니다.

- TO 또는 RESET절이 있는 CONNECT 명령문이 성공하면 다음이 발생합니다.
 - AS의 이름은 현재 서버 특수 레지스터에 있습니다.
- 오류를 리턴한 시스템 모듈을 식별하는 정보는 SQLCA(SQL 통신 영역)의 SQLERRP 필드 또는 SQL 진단 영역의 DB2_MODULE_DETECTING_ERROR에 있습니다.

- | - 또한 연결 유형에 대한 정보는 SQLCA와 SQL 진단 영역에 있습니다. 여기에서 인코드된 정보는 다음과 같습니다.
- | - 어플리케이션이 연결된 상태인지 연결되지 않은 상태인지의 여부는 SQLCA의 SQLERRD(5) 또는 SQL 진단 영역의 DB2_CONNECTION_STATE에서 알 수 있습니다.
- | - 리모트 연결이 보호된 대화를 사용하는지 또는 보호되지 않은 대화를 사용하는지 여부는 SQLCA의 SQLERRD(4) 또는 SQL 진단 영역의 DB2_CONNECTION_TYPE에서 알 수 있습니다.
- | - 연결이 항상 읽기 전용인지, 항상 갱신가능한지의 여부와 갱신 기능이 각 작업 단위 사이에서 변경될 수 있는지의 여부는 SQLCA 내의 SQLERRD(4) 또는 SQL 진단 영역 내의 DB2_CONNECTION_STATUS에서 찾을 수 있습니다.
- | SQLCA의 SQLERRD 필드에 대한 자세한 정보와 SQL 진단 영역의 연결 정보는 SQL 프로그래밍 개념 주제를 참조하십시오.
- AR이 연결 가능 상태가 아니고 *server-name*이 로컬 관계형 데이터베이스 디렉토리에 나열되어 있지 않기 때문에 TO 또는 RESET절이 있는 CONNECT 명령문이 성공하지 못하면 AR의 연결 상태는 변경되지 않습니다.
- 현재 연결된 AS에 연결하면 오류가 발생합니다.
- TO 또는 RESET절이 없는 연결은 현재 연결에 대한 정보를 얻는 데 사용될 수 있습니다. 여기에는 다음 정보가 포함되어 있습니다.
 - | - 상태를 리턴한 시스템 모듈을 식별하는 정보는 SQLCA(SQL 통신 영역)의 SQLERRP 필드 또는 SQL 진단 영역의 DB2_MODULE_DETECTING_ERROR에 있습니다.
 - | - 다른 상태 정보는 SQLERRD(4)와 SQLERRD(5)의 내용 및 대응하는 SQL 진단 영역의 정보를 설명한 섹션에 기술되어 있습니다.

어플리케이션 프로세스가 실행하는 첫 번째 SQL문이 CONNECT 명령문이 되는 것이 좋습니다. 그러나, 프로그램에 CONNECT 명령문을 삽입할 때 프로그램이 하나 이상의 AS에 연결되면 AS 이름을 동적으로 변경할 수 있습니다. 다중 서버에서 어플리케이션을 실행할 경우, 아래와 같이 호스트 변수가 있는 CONNECT 명령문을 지정하여 프로그램이 관계형 데이터베이스 이름을 전달하도록 할 수 있습니다.

```
CONNECT TO : host-variable
```

CONNECT 명령문이 없으면 AS를 변경할 때 사용자가 해야 할 일은 새로운 관계형 데이터베이스 이름을 사용하여 프로그램을 다시 컴파일하는 것입니다.

다음 예는 어플리케이션 프로그램에 있는 두 가지 형식의 CONNECT 명령문(**1** 과 **2**)입니다.

```
CRTSQLxxx PGM(SPIFFY/FIXTOTAL) COMMIT(*CHG) RDB(KC105)
```

```
PROC: FIXTOTAL;
EXEC SQL CONNECT TO KC105; 1

:
EXEC SQL
SELECT * INTO :SERVICE
FROM REPAIRTOT;
```

```

:
EXEC SQL COMMIT;

:
EXEC SQL CONNECT TO MPLS03 USER :USERID USING :PW; 2

:
EXEC SQL SELECT ...

:
EXEC SQL COMMIT;

:
END FIXTOTAL;

```

예(2)는 CONNECT 명령문 USER/USING 양식의 사용을 표시합니다. CONNECT 명령문의 해당 양식을 프로그램에 삽입할 때 호스트 변수로 사용자 ID와 암호를 지정해야 합니다. TCP/IP를 사용하는 경우, 연결 시 서버 인증 항목 추가(ADDSVRAUTE) 명령을 저장할 적절한 매개변수와 함께 사용하면 사용자 ID 및 암호를 보안 오브젝트로부터 추출할 수 있습니다.

다음 예는 대화식 SQL의 두 가지 CONNECT 명령문 형식입니다. 암호는 작은 따옴표로 표시한 것에 주의하십시오.

```

SQL 명령문을 입력하고 Enter를 누르십시오.
현재 연결은 관계형 데이터베이스(RDB) KC105입니다.
CONNECT TO KC000 _____
:
COMMIT _____
====> CONNECT TO MPLS03 USER JOE USING 'X47K' _____
_____
_____

```

분산 관계형 데이터베이스 및 SQL CALL의 특정 SQL

분산 iSeries용 DB2 UDB 어플리케이션의 사전컴파일 프로세스가 실행되는 동안 OS/400 프로그램은 어플리케이션 서버(AS)에서 실행될 SQL 패키지를 빌드합니다. 컴파일 이후에 분산 SQL 프로그램과 패키지는 어플리케이션 수신자와 어플리케이션 서버로 사용되는 서버와 호환되어야 합니다. 213 페이지의 『분산 관계형 데이터베이스 프로그램 작성』은 사용자에게 사전컴파일 프로세스로의 변경과 SQL 패키지의 추가에 대한 자세한 내용을 제공합니다.

이 섹션에서는 분산 관계형 데이터베이스(RDB) 지원에 사용되는 SQL문과 다른 서버와의 공존에 대해 고려할 몇 가지 사항에 대한 개요를 제공합니다. 이 주제에 대한 자세한 정보는 iSeries Information Center의 SQL 참조서 및 SQL 프로그래밍 개념 주제를 참조하십시오.

분산 관계형 데이터베이스명령문

다음 명령문에는 특히 분산 관계형 데이터베이스(RDB)를 지원하는 SQL 언어가 들어 있습니다.

- CONNECT
- SET CONNECTION

- RELEASE
- DISCONNECT
- DROP PACKAGE
- GRANT EXECUTE ON PACKAGE
- REVOKE EXECUTE ON PACKAGE

SQL CALL 명령문은 로컬로 사용될 수 있지만 주요 목적은 리모트 서버에서 프로시저어를 호출하는 것입니다.

196 페이지의 『분산 관계형 데이터베이스(DRDB)에 연결』은 어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 사이의 연결을 관리하기 위한 CONNECT, SET CONNECTION, RELEASESQL 패키지에 사용자 권한을 부여하거나 취소하기 위해 SQL GRANT EXECUTE ON PACKAGE 및 REVOKE EXECUTE ON PACKAGE 명령문을 사용하는 방법은 65 페이지의 『분산 관계형 데이터베이스 오브젝트에 권한』에서 설명됩니다.

SQL 패키지를 제거하는 데 사용하는 SQL DROP PACKAGE 명령문은 220 페이지의 『SQL 패키지에 대한 작업』에서 설명됩니다.

SQL CALL 명령문(저장된 프로시저어)

주: iSeries용 DB2 UDB는 V5R1 이전에는 저장된 프로시저어에서 결과 세트 리턴을 지원하지 않았습니다. V5R1에서는 비iSeries 클라이언트로부터 저장 프로시저어 호출을 위해 DRDA 서버에 지원이 추가되었습니다. V5R2에서는 SQL을 위한 CLI 인터페이스를 사용하는 어플리케이션을 위해 iSeries 클라이언트 측 지원이 추가되었습니다. 그러나 저장 프로시저어 결과 세트를 V5R2 iSeries 클라이언트에 리턴할 수 있도록 하려면 PTF를 V5R1 iSeries 서버에 적용해야 합니다. PTF의 설명은 "저장 프로시저어 결과 세트를 V5R2 iSeries DRDA 클라이언트로 리턴하는 것을 지원하는 V5R1 DRDA 서버 PTF"입니다. 이 기능에 필요한 V5R1 PTF에 대한 자세한 정보는 정보 APAR ii13348을 참조하십시오.

결과 세트는 SQL SELECT문과 연관된 하나 또는 그 이상의 SQL 커서를 열어서 저장 프로시저어에 생성될 수 있습니다. 또 최대 하나의 배열 결과 세트도 리턴됩니다. 결과 세트를 리턴하는 저장 프로시저어 쓰기에 대한 자세한 정보는 iSeries Information Center의 SQL 참조 주제에서 SET RESULT SETS 및 CREATE PROCEDURE문의 설명을 참조하십시오.

SQL CALL 명령문이 실제로 분산 관계형 데이터베이스에 특정된 것은 아니지만 기본 값이 분산 어플리케이션 논리와 처리에 있기 때문에 이에 대한 설명이 이 곳에 포함되었습니다. CALL 명령문은 RPC(리모트 프로시저어 호출) 메커니즘이 개방 소프트웨어 기초**(OSF**) 분산 컴퓨팅 환경(DCE)에서 실행하는 것과 상당히 유사한 DRDA 환경에서 기능을 제공합니다. 사실상, 리모트 관계형 데이터베이스에 있는 프로그램에 대한 SQL CALL은 실제 리모트 프로시저어 호출입니다. 이 유형의 RPC에는 인터페이스의 컴파일이 필요하지 않고 스태브 프로그램 작성도 필요하지 않다는 장점이 있습니다.

다음과 같은 이유로 사용자는 가끔 저장된 프로시저어라고 하는 SQL CALL을 사용합니다.

- 주어진 기능을 실행하여 어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 사이의 메시지 흐름의 수를 줄이기 위해. SQL 연산 세트를 실행하면 서버측 프로그램에 명령문과 상호 연결 논리가 있는 것이 훨씬 효과적입니다.
- 원래 데이터베이스 작업을 리모트 위치에서 실행되도록 하기 위해.
- 비데이터베이스 작업(예를 들어, 메시지 전송이나 자료 대기행렬 연산 실행)을 SQL을 사용하여 실행하기 위해.

주: 데이터베이스 작업과 달리 이 작업은 서버의 확약 제어에 의해 보호되지 않습니다.

- 리모트 서버의 서버 API(어플리케이션 프로그래밍 인터페이스)에 액세스.

저장된 프로시저와 어플리케이션 프로그램은 동일하거나 다른 활성 그룹에서 실행될 수 있습니다. AR의 어플리케이션 프로그램과 AS의 저장된 프로시저 사이의 일관성을 유지하려면 지정된 ACTGRP(*CALLER)로 저장된 프로시저를 컴파일할 것을 권장합니다. 저장 프로시저가 결과 세트를 리턴하도록 설계되어 있다면, *NEW 활성 그룹에서 실행하도록 작성하면 안됩니다. 만일 그렇게 하면, 결과 세트와 연관된 커서는 프로시저가 호출자에게 리턴되고 활성 그룹이 파괴될 때 불완전하게 닫힐 수 있습니다.

저장된 프로시저가 조회 메시지를 발행하도록 호출되면 메시지는 QSYSOPR 메시지 대기행렬로 송신됩니다. 저장된 프로시저는 조회 메시지에 대한 응답을 기다립니다. 저장된 프로시저가 조회 메시지에 응답하도록 하려면 응답 리스트 항목 추가(ADDRPYLE) 명령을 사용하고 저장된 프로시저의 작업 변경(CHGJOB) 명령의 INQMSGRPY 매개변수에 *SYSRPYL을 지정하십시오.

디폴트 활성 그룹의 AS 작업에서 COMMIT 또는 ROLLBACK이 실행되면 저장된 프로시저에서 COMMIT 또는 ROLLBACK을 실행할 수 없습니다. 저장된 프로시저와 어플리케이션 프로그램이 다른 확약 정의에서 실행될 때 어플리케이션 프로그램의 COMMIT과 ROLLBACK 명령문은 자신의 확약 정의에만 영향을 미칩니다. 다른 방법으로 저장된 프로시저의 변경을 확약해야 합니다.

SQL CALL에 대한 자세한 내용은 iSeries Information Center의 SQL 참조서 주제를 참조하십시오.

DB2 Universal Database(UDB)에 대해 SQL CALL을 사용하여 저장된 프로시저 호출: DB2 UDB를 실행하는 플랫폼에서 호출되어 C로 작성된 저장 프로시저는 argc와 argv를 매개변수로 사용할 수 없습니다(즉, 유형 main()이 될 수 없습니다). 이는 argc와 argv를 사용해야 하는 iSeries 저장 프로시저와는 다릅니다. DB2 UDB 플랫폼의 저장 프로시저 예는 \SQLLIB\SAMPLES(또는 /sqllib/samples) 서브디렉토리를 참조하십시오. C 서브디렉토리에서 outsrv.sqc와 outcli.sqc를 검색하십시오.

iSeries 서버가 호출한 UDB 저장 프로시저의 경우, 프로시저명은 반드시 대문자여야 합니다. 현재 iSeries 서버는 프로시저명으로 대문자를 사용합니다. 소문자로 된 같은 프로시저명이 UDB 서버에 있으면 프로시저명을 찾을 수 없습니다. iSeries 서버의 저장 프로시저의 경우, 프로시저명은 대문자입니다.

iSeries 서버의 저장 프로시저가 같은 활성 그룹에서 호출 프로그램으로 실행되도록 작성되면 그 안에 COMMIT을 가질 수 없습니다(적절한 작성 방법). UDB에서 저장된 프로시저는 COMMIT을 가질 수 있지만 어플리케이션 설계자는 확약이 발생하는 iSeries용 DB2 UDB 부분에 대해 알 수 없습니다.

iSeries용 DB2 UDB 공존

SQL 언어를 사용하여 분산 관계형 데이터베이스(RDB) 프로그램을 작성하고 유지보수할 때 분산 관계형 데이터베이스(RDB) 네트워크에 있는 다른 서버를 고려해야 합니다. 사용자가 작성하고 유지보수하는 프로그램은 다음과 호환되어야 합니다.

- 기타 iSeries 서버
- 이전 iSeries 서버 릴리스
- iSeries 서버가 아닌 서버

분산 SQL 프로그램의 SQL문은 어플리케이션 서버(AS)에서 실행됨을 기억하십시오. 프로그램이 어플리케이션 리퀘스터(AR)에서 실행되더라도 SQL문은 AS에서 실행될 SQL 패키지에 있습니다. 이 명령문은 AS가 지원해야 하며 AS에 있는 컬렉션, 표 및 뷰와 호환되어야 합니다. 또한 AR에서 프로그램을 실행하는 사용자는 AS에 있는 SQL 패키지와 다른 SQL 오브젝트에 대한 권한을 부여 받아야 합니다.

CRTSQLxxx 명령을 사용하고 AS의 관계형 데이터베이스명(RDB 매개변수)을 지정하여 다시 프로그램을 작성함으로써 비분산 삽입된 SQL 프로그램을 분산 삽입된 SQL 프로그램으로 변환할 수 있습니다. iSeries용 DB2 Universal Database의 분산 관계형 데이터베이스 지원을 사용하여 다시 프로그램을 컴파일하고 AS에서 필요한 SQL 패키지를 작성합니다.

사용자는 iSeries 서버가 아닌 어플리케이션 서버에서 실행되는 iSeries용 DB2 UDB 프로그램을 작성할 수 있으며 다른 플랫폼은 몇 가지 SQL 기능을 지원할 수 있습니다. iSeries용 DB2 UDB AR에서 지원되지 않는 명령문은 AS가 기능을 지원할 때 서버에서 사용되거나 컴파일될 수 있습니다. iSeries 서버 AS에서만 실행되도록 작성된 SQL 프로그램은 이 안내서에 설명된 지원 레벨을 제공합니다. 제공하는 기능의 레벨을 판별하려면 다른 시스템용 지원 설명서를 참조하십시오.

DRDA 작업 단위 종료

확약되지 않은 작업이 있는 SQL 프로그램 종료를 주의해야 합니다. 프로그램이 확약되지 않은 작업과 함께 종료되면 관계형 데이터베이스에 대한 연결은 여전히 사용 중입니다(그러나 system-named 활성 그룹에서 실행 중인 프로그램의 경우, 시스템은 프로그램이 종료될 때 자동으로 확약을 실행합니다).

이 작동은 OS/400 오퍼레이팅 시스템에서, COMMIT과 ROLLBACK을 명령 행이나 CL 프로그램에서 명령으로 사용할 수 있기 때문에 다른 시스템과는 다릅니다. 그러나, 앞의 시나리오는 이 상황을 계획하지 않는 한, 다음 SQL 프로그램이 실행될 때 예상치 못한 결과를 가져올 수 있습니다. 예를 들어, 다음에 대화식 SQL(STRSQL 명령)을 실행하면 대화식 세션은 확약되지 않은 작업이 있는 이전 어플리케이션 서버(AS)에 연결된 상태에서 시작됩니다. 다른 예를 들면, 이전 시나리오를 따를 경우, 내재적 연결을 실행하는 두 번째 SQL 프로그램을 시작하고 마지막으로 사용한 AS에서 패키지를 찾아 실행을 시도합니다. 이 상황은 사용자가 AS에서 의도하지 않은 것일 수 있습니다. 이 상황을 피하려면 어플리케이션 프로그램을 종료하기 전에 마지막 작업 단위를 항상 확약하거나 롤백하십시오.

저장 프로시듀어, 사용자 정의 기능(UDF) 및 약속 제어

대화식 SQL과 같은 애플리케이션이 DRDA 연결에서 활동 약속 제어(COMMIT(*NONE))없이 실행되면 호출된 저장 프로시듀어 또는 사용자 정의 기능(UDF)이 iSeries 서버에서 약속 제어를 시작할 수 있습니다. 그러면 클라이언트와 서버간 약속 제어가 일치하지 않아서 애플리케이션이 종료할 때 미약약 갱신이 발생할 수 있습니다.

이 상황은 피해야 합니다. 그러나 구현할 경우 한 가지 해결 방법은 약속 제어에서 실행되는 저장 프로시듀어 또는 UDF가 모든 데이터베이스 갱신을 명시적으로 약속하는 것입니다. 그렇게 하지 않으면 V5R3부터 서버가 단절 프로세스에서 지연 중인 갱신을 감지하여 자동으로 지연 중인 작업을 약속합니다.

코드화 문자 세트 ID(CCSID)

어떤 국가의 자국어 지원은 최소 문자 세트에 대한 적절한 처리를 요구합니다. 문자 정보 관리의 공통 시스템 지원은 IBM 문자 자료 표시 구조(CDRA)로 제공됩니다. CDRA는 코드화 문자 세트 ID(CCSID) 값을 정의 하여 문자를 표현하는 데 사용된 코드점을 식별하고 필요에 따라 문자의 의미를 보존하기 위해 이 코드(문자 자료)를 변환합니다.

CDRA와 같은 구조와 관련 변환 프로토콜의 사용은 다음 상황에서 중요합니다.

- iSeries 서버에 하나 이상의 자국어 버전이 설치된 경우
- 여러 iSeries 서버가 다른 1차 자국어 버전을 갖는 다른 국가에서 시스템 간의 자료를 공유하는 경우
- iSeries 서버와 비iSeries 서버가 1차 자국어 버전을 갖는 다른 국가에서 시스템 간의 자료를 공유하는 경우

태그는 코드화 그래픽 문자에 의미를 할당하는 주요한 수단입니다. 태그는 자료 오브젝트와 관련된 자료 구조에 있거나(명시 태그) 작업이나 시스템 자체와 같은 오브젝트에서 계승될 수 있습니다(내재 태그).

iSeries용 DB2 UDB는 CCSID를 사용하여 문자 열에 태그를 붙입니다. **CCSID**는 코드화 체계 ID, 문자 세트 ID, 코드 페이지 ID 및 사용된 코드화 그래픽 문자 표시를 고유하게 식별하는 추가 코드 관련 정보의 특정 세트를 식별하는 16비트 숫자입니다. 애플리케이션을 실행하면 자료를 다른 시스템으로 송신할 때 CCSID와 함께 태그가 붙어 송신되고 자료가 변환됩니다. 자료를 태그하는 방법이 다를 경우, 수신하는 작업은 자료를 자신의 CCSID로 자동 변환합니다.

CDRA는 CCSID에 대한 다음 범위의 값을 정의합니다.

00000 다음 단계의 CCSID 사용

00001부터 28671

IBM이 등록한 CCSID

28672부터 65533

예약

65534 낮은 단계의 CCSID 참조

65535 변환이 실행되지 않음

OS/400 CCSID 리스트는 iSeries Information Center의 자국어 지원 주제를 참조하고, CDRA CCSID에 대한 자세한 리스트는 문자 자료 표시 구조 - 레벨 1, 레지스트리를 참조하십시오. CCSID 처리에 대한 자세한 정보는 iSeries Information Center의 SQL 참조서 및 SQL 프로그래밍 개념 주제를 참조하십시오.

다음 일러스트레이션은 일부 CCSID입니다.

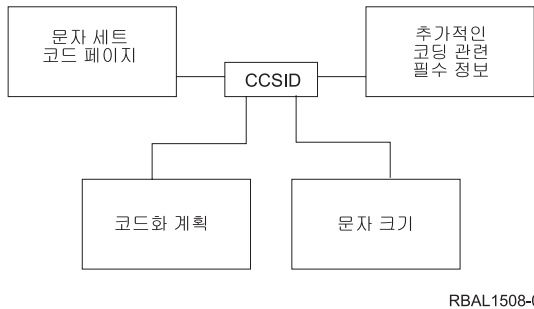


그림 21. 코드화 문자 세트 ID(CCSID)

iSeries 서버 지원

iSeries 서버에서의 작업에 대한 디폴트 CCSID는 작업 변경(CHGJOB) 명령을 사용하여 지정됩니다. CCSID를 이 방법으로 지정하지 않으면 작업 CCSID는 사용자 프로파일의 CCSID 속성으로부터 얻습니다. CCSID가 사용자 프로파일에 지정되어 있지 않으면 시스템은 CCSID를 QCCSID 시스템 값에서 가져 옵니다. QCCSID 값은 처음에 65535로 설정됩니다. 사용자의 서버가 상이한 시스템의 분산 관계형 데이터베이스에 있으면 CCSID 65535를 사용할 수 없습니다. 상이한 환경에서 운영할 때 고려해야 할 사항은 253 페이지의 부록 B 『DRDA를 사용하는 공통 플랫폼 액세스』를 참조하십시오.

어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 사이에 흐르는 모든 제어 정보는 CCSID 500(DRDA 표준)에 있습니다. 이 정보는 콜렉션 명, 표 이름과 일부 설명 텍스트와 같은 정보입니다. 제어 정보의 변형 문자를 사용하면 이름이 변환되어 성능에 영향을 줄 수 있습니다. 패키지 이름이 CCSID 500에 전송될 수 있습니다. 패키지 이름에 변형 문자를 사용하면 패키지 이름이 변환됩니다. 따라서 실행 시간에 패키지를 찾지 못합니다.

작업을 시작한 후에 작업 변경(CHGJOB) 명령을 사용하여 작업 CCSID를 변경할 수 있습니다. 이렇게 하려면

1. 작업에 대한 작업(WRKJOB) 명령을 입력하여 작업에 대한 작업을 표시하십시오.
2. 옵션 2(작업 정의 속성 표시)를 표시하십시오.

현재 CCSID 값을 찾아 작업을 원래의 CCSID 값으로 재설정할 수 있습니다.

3. 새로운 CCSID 값으로 작업 변경(CHGJOB) 명령을 입력하십시오.

새 CCSID 값은 작업에서 즉시 반영됩니다. 그러나, 사용자가 변경한 작업 CCSID가 AR 작업이면 새 CCSID는 다음 CONNECT까지 실행 중인 작업에 영향을 주지 않습니다.

경고: AS 작업의 CCSID를 변경하면 결과는 예상할 수 없습니다.

소스 파일의 소스 실제 파일 작성(CRTSRCPF)이나 실제 파일 작성(CRTPF) 명령에 CCSID가 명시적으로 지정되지 않으면 소스 파일은 작업 CCSID로 태그됩니다. 외부적으로 설명된 데이터베이스 파일과 표는 CCSID가 자료 서술 스펙(DDS), 대화식 자료 정의 유틸리티(IDDU) 또는 CREATE TABLE SQL문에 명시적으로 지정되지 않을 경우 작업 CCSID로 태그됩니다.

소스와 외부 서술 파일의 경우, 작업 CCSID가 65535이면 오퍼레이팅 시스템 언어에 근거한 디폴트 CCSID가 사용됩니다. 프로그램 서술 파일은 CCSID 65535로 태그됩니다. 뷰는 대응하는 표 태그나 열 레벨 태그의 CCSID로 태그됩니다. 뷰가 여러 표에 정의되어 있으면 열 레벨에서 태그되고 기본 열의 태그를 가집니다. 뷰는 CCSID로 외부적으로 태그될 수 없습니다. 시스템은 CCSID가 동일하지 않고 어떤 CCSID도 65535가 아니면 작업과 표 사이의 자료를 자동으로 변환합니다.

태그된 표의 CCSID를 변경하면 열 레벨에서 태그될 수 없거나 뷰를 정의할 수 없습니다. 태그된 표의 CCSID를 변경하려면 실제 파일 변경(CHGPF) 명령을 사용하십시오. 열 레벨 태그가 있는 표를 변경하려면 다시 작성한 다음, 파일 복사(CPYF) 명령의 FMT(*MAP)를 사용하여 자료를 새 표에 복사해야 합니다. 표에 하나 이상의 뷰가 정의되어 있으면 표를 변경하기 위해 다음을 실행해야 합니다.

1. 뷰와 표를 액세스 경로와 함께 저장하십시오.
2. 뷰를 삭제하십시오.
3. 표를 변경하십시오.
4. 작성된 표에서 뷰와 액세스 경로를 복원하십시오.

태그되지 않았거나 65535로 내재적으로 태그된 소스 파일과 iSeries용 DB2 Universal Database에 마이그레이트된 외부 서술 파일은 설치된 오퍼레이팅 시스템 언어에 근거하여 디폴트 CCSID로 태그됩니다. 따라서 iSeries용 DB2 Universal Database에 복원된 새 릴리스와 파일이 설치될 때 시스템에 있는 파일이 포함됩니다.

AR과 AS 사이에 송신된 모든 자료는 변환되지 않습니다. 또한 CCSID도 송신됩니다. 자료를 태그하는 방법이 다를 경우, 수신하는 작업은 자료를 자신의 CCSID로 자동 변환합니다. 예를 들어, 특약점 시스템 KC105에서 실행되는 다음 어플리케이션을 참고하십시오.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```
CRTSQLxxx PGM(PARTS1) COMMIT(*CHG) RDB(KC000)
```

```
PROC :PARTS1;
.
.
EXEC SQL
  SELECT * INTO :PARTAVAIL
  FROM INVENTORY
  WHERE ITEM = :PARTNO;
.
.
END PARTS1;
```

위의 예에서 로컬 시스템(KC105)은 QCCSID 시스템 값을 CCSID 37에 설정합니다. 리모트 지역 센터(KC000)는 CCSID 937을 사용하고 모든 표는 CCSID 937로 태그됩니다. CCSID 처리는 다음과 같습니다.

- KC105 시스템은 CCSID 37의 입력 호스트 변수(:PARTNO)를 송신합니다.(DECLARE VARIABLE SQL 문은 작업의 CCSID가 호스트 변수에 적합하지 않을 경우, 사용될 수 있습니다.)
- KC000 시스템은 :PARTNO를 CCSID 937로 변환하고, 필요한 자료를 선택하며 자료를 CCSID 937의 KC105로 되돌려 보냅니다.
- KC105가 자료를 가져오면 CCSID 37로 변환하고 로컬 사용으로 :PARTAVAIL에 둡니다.

기타 DRDA 자료 변환

때때로 리모트 시스템에서 처리를 실행할 때 사용자 프로그램은 한 시스템에서 자료를 변환하여 다른 시스템에서 사용할 수 있도록 합니다. iSeries 서버의 DRDA 지원은 DRDA 지원을 사용하는 다른 시스템 간에 자료를 자동으로 변환합니다. iSeries용 DB2 Universal Database 어플리케이션 리퀘스터 (AR)가 어플리케이션 서버(AS)에 연결될 때, 그 유형을 식별하는 정보를 송신합니다. 마찬가지로 AS는 프로세서 유형(즉, S/390* 호스트 또는 iSeries 서버)을 식별하는 정보를 서버로 다시 보냅니다. 그런 다음, 두 시스템은 연결에 정의된 대로 두 시스템 사이의 자료를 자동으로 변환합니다. 따라서 두 시스템의 구조적인 차이점에 대해 프로그램할 필요가 없습니다.

DRDA 지원이 되는 IBM 시스템 간의 자료 변환에는 다음과 같은 자료 유형이 포함됩니다.

- 부동 소수점 표시
- 존(Zone) 십진 표시
- 바이트 반전
- 혼합 자료 유형
- 다음과 같은 iSeries 특정 자료 유형
 - DBCS 전용
 - DBCS 선택
 - 정밀도와 척도가 있는 정수

DDM 파일 및 SQL

SQL 분산 관계형 데이터베이스(RDB) 지원을 사용하는 프로그램 내에서 일부 분산 관계형 데이터베이스(RDB) 작업을 실행하는데 도움이 되는 iSeries DDM 지원을 사용할 수 있습니다. 예를 들어, 많은 수의 레코드를 가져오는 데는 SQL FETCH 명령문보다는 DDM과 파일 복사(COPY) 명령을 사용하는 것보다 빠릅니다. 또한, DDM은 분산 관계형 데이터베이스(RDB) 어플리케이션에 사용하기 위해 컴파일하는 동안 가져온 리모트 시스템 자료의 외부 파일 설명을 가져오는 데 사용할 수 있습니다. 이렇게 하려면 제 3 장 『iSeries 분산 관계형 데이터베이스의 통신』 제 5 장 『iSeries 분산 관계형 데이터베이스 설정』에 설명된 대로 DDM을 사용해야 합니다.

다음 예는 같은 작업은 어플리케이션 서버(AS) 및 어플리케이션 리퀘스터(AR)에서 사용할 수 있도록 관계형 데이터베이스 디렉토리 항목을 추가하고 DDM 파일을 작성하는 방법을 설명합니다.

주: 대화를 공유하려면 두 연결을 모두 보호하거나 두 연결 모두 보호하지 말아야 합니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

관계형 데이터베이스 디렉토리

```
ADDRDBDIRE  RDB(KC000) +
             RMTLOCNAME(KC000)
             TEXT('Kansas City regional database')
```

DDM File:

```
CRTDDMF  FILE(SPIFFY/UPDATE)
         RMTFILE(SPIFFY/INVENTORY)
         RMTLOCNAME(KC000)
         TEXT('DDM file to update local orders')
```

다음은 리모트 서버의 같은 작업에서 관계형 데이터베이스 디렉토리 항목과 DDM 파일을 모두 사용하는 샘플 프로그램입니다.

```
CRTSQLxxx PGM(PARTS1) COMMIT(*CHG) RDB(KC000) RDBCNNMTH(*RUW)
```

```
PROC :PARTS1;
OPEN SPIFFY/UPDATE;
.
.
.
CLOSE SPIFFY/UPDATE;
.
.
.
EXEC SQL
  SELECT * INTO :PARTAVAIL
  FROM INVENTORY
  WHERE ITEM = :PARTNO;
EXEC SQL
  COMMIT;
.
.
.
END PARTS1;
```

iSeries DDM 지원을 사용하는 방법에 대한 자세한 내용은 iSeries Information Center의 분산 자료 관리 주제를 참조하십시오.

분산 관계형 데이터베이스 프로그램 작성

SQL 언어를 사용하여 프로그램을 작성할 때는 호스트 프로그램에 SQL문을 삽입할 수 있습니다. 호스트 프로그램은 iSeries PL/I, ILE C/400®, COBOL/400, ILE COBOL/400, FORTRAN/400, RPG/400 또는 ILE RPG/400 프로그래밍 언어와 같은 호스트 언어 중 하나로 작성한 SQL문이 들어 있는 프로그램입니다. 호스트 프로그램에서는 호스트 변수라고 하는 변수를 사용합니다. 이 변수는 호스트 프로그램으로 식별되는 SQL 문에서 사용되는 변수입니다. RPG에서는 이 변수를 필드명이라 하고, FORTRAN, PL/I 및 C에서는 변수, COBOL에서는 자료 항목이라고 합니다.

분산되지 않은 iSeries용 DB2 UDB 프로그램을 코딩하는 것과 유사한 방법으로 분산 iSeries용 DB2 Universal Database 프로그램을 코딩할 수 있습니다. 호스트 언어를 사용하여 호스트 변수가 있는 SQL문을 삽입합니다. 또한, 분산되지 않은 iSeries용 DB2 UDB 프로그램처럼 분산 iSeries용 DB2 UDB 프로그램은 다음 프로세스를 사용하여 준비됩니다.

- 사전컴파일
- 테스트 및 디버깅
- 어플리케이션 바인딩
- 어플리케이션 프로그램 컴파일

그러나, 분산 iSeries용 DB2 UDB 프로그램은 SQL 패키지가 자료에 액세스하기 위해 어플리케이션 서버(AS)에서 작성될 것을 요구합니다.

이 섹션에서는 분산 iSeries용 DB2 UDB 프로그램의 차이점을 설명하면서 프로세스의 단계를 설명합니다.

SQL문을 사용한 프로그램 사전컴파일

사용자는 삽입된 SQL문이 들어 있는 어플리케이션 프로그램을 실행하기 전에 사전컴파일하고 컴파일해야 합니다. 프로그램이 SQL 사전컴파일러에 의해 실행될 수 있도록 사전 컴파일. SQL 사전컴파일러는 어플리케이션 프로그램의 각 명령문을 스캔하고 다음을 실행합니다.

- SQL문과 호스트 변수명 정의를 검색
- 각 SQL문이 유효하고 구문 오류가 없는지 확인
- 데이터베이스의 설명을 사용하여 SQL문 확인
- 호스트 언어의 컴파일에 대한 SQL문 준비
- 사전컴파일된 SQL문에 대한 정보 생성

어플리케이션 프로그래밍 명령문과 삽입된 SQL문은 SQL 사전컴파일러에 대한 1차 입력입니다. SQL 사전컴파일러는 호스트 언어 명령문이 구문상 정확하다고 가정합니다. 호스트 언어 명령문이 구문 상 정확하지 않으면 사전컴파일러는 SQL문과 호스트 변수 선언을 정확하게 식별할 수 없습니다.

SQL 사전컴파일 프로세스는 리스트와 임시 소스 파일 멤버를 생성합니다. 또한 사전컴파일러 명령의 OPTION과 RDB 매개변수에 지정된 내용에 따라 SQL 패키지를 생성할 수 있습니다. 이 매개변수에 대한 자세한 내용은 216 페이지의 『어플리케이션 프로그램 컴파일』을 참조하십시오.

리스트

출력 리스트는 CRTSQLxxx 명령의 PRTFILE 매개변수에 지정된 프린터 파일로 송신됩니다. 다음 항목은 프린터 파일에 기록됩니다.

- 사전 컴파일러 옵션

다음은 CRTSQLxxx 명령으로 지정된 모든 옵션과 소스 멤버를 마지막으로 변경한 날짜의 리스트입니다.

- 사전컴파일러 소스

이 출력은 *SOURCE 옵션이 비ILE 사전컴파일러에 사용되거나 OUTPUT(*PRINT) 매개변수가 ILE 사전 컴파일러로 지정되면 생성됩니다. 각 사전컴파일러 소스 명령문을 사전컴파일러에 의해 할당된 레코드 번호, 소스 입력 유틸리티(SEU)를 사용할 때 나타나는 순번(SEQNBR) 및 레코드가 마지막으로 변경된 날짜와 함께 나타내고 있습니다.

- 사전컴파일러 공통 참조

이 출력은 *XREF가 OPTION 매개변수에 지정된 경우에 생성됩니다. 호스트 변수의 이름이나 SQL 엔티티(예: 표와 열), 이름이 정의된 레코드 번호, 정의된 이름 및 이름이 생기는 레코드 번호를 나타내고 있습니다.

- 사전컴파일러 진단 리스트

이 출력은 오류에서 명령문의 사전컴파일러 레코드 번호를 나타내는 진단 메시지를 나타내고 있습니다.

임시 소스 파일 멤버

사전컴파일러가 처리한 소스 명령문은 QTEMP 라이브러리의 QSQLTEMP에 기록됩니다(CRTSQLRPGI를 사용하여 작성된 프로그램의 QTEMP 라이브러리의 QSQLTEMP1). 사전컴파일러로 변경된 소스 코드에서, SQL 문은 SQL 인터페이스 모듈 QSQRROUTE, QSQLOPEN, QSQLCLSE 및 QSQLCMIT에 대한 주석과 호출로 변환됩니다. 임시 소스 파일 멤버의 이름은 CRTSQLxxx의 PGM 매개변수에 지정된 이름과 같습니다. 이 멤버는 컴파일러에 대한 입력으로 사용되기 전에는 변경될 수 없습니다.

QSQLTEMP 또는 QSQLTEMP1은 나중에 컴파일하고 싶으면 사전컴파일 후에 영구 라이브러리로 이동될 수 있습니다. 임시 소스 파일 멤버의 레코드를 변경하면 나중에 시도한 컴파일은 실패합니다.

SQL 패키지 작성

SQL 패키지를 호출한 오브젝트는 CRTSQLxxx 명령을 컴파일할 때 사전컴파일 프로세스의 부분으로 작성될 수 있습니다. 이 프로세스의 부분으로 패키지를 작성할 때 영향을 미치는 상황에 대한 내용은 216 페이지의 『어플리케이션 프로그램 컴파일』과 217 페이지의 『어플리케이션 바인딩』을 참조하십시오. 패키지에 대한 작업에 사용할 수 있는 SQL 패키지와 명령에 대한 자세한 정보는 220 페이지의 『SQL 패키지에 대한 작업』을 참조하십시오.

사전컴파일러 명령

DB2 UDB 조회 관리자 및 SQL 개발 킷 프로그램에는 각 호스트 언어마다 하나씩 7개의 사전컴파일러 명령이 있습니다.

호스트 언어	명령
iSeries PL/I	CRTSQLPLI
ILE C/400 언어	CRTSQLCI
COBOL/400 언어	CRTSQLCBL
ILE COBOL/400 언어	CRTSQLCBLI
FORTRAN/400 언어	CRTSQLFTN
RPG III(RPG/400 언어의 부분)	CRTSQLRPG
ILE RPG/400 언어	CRTSQLRPGI

각 언어에 대한 개별 명령이 존재하므로 각 언어는 해당 언어에만 적용되는 매개변수를 가질 수 있습니다. 예를 들어, 옵션 *APOST 및 *QUOTE는 COBOL에만 유일하게 적용됩니다. 이 옵션은 다른 언어에 대한 명령에는 포함되지 않습니다. 사전컴파일러는 SQL 사전컴파일러 명령 중 하나에 의해 호출될 때 지정된 매개변수에 의해 제어됩니다. 매개변수는 입력 처리 방법과 출력 제공 방법을 지정합니다.

사용자는 CRTSQLxxx 명령의 PGM 매개변수(비ILE 사전컴파일용)나 OBJ 매개변수(ILE 사전컴파일용)로 프로그램 소스 명령문이 들어 있는 멤버의 이름 이외의 것은 지정하지 않고 프로그램을 사전컴파일할 수 있습니다. SQL은 모든 사전컴파일러 매개변수에 대한 디폴트 값을 할당합니다(그러나, 사용자가 명시적으로 지정하는 값으로 대체될 수 있습니다).

다음은 분산 관계형 데이터베이스를 지원하는 데 사용되는 모든 CRTSQLxxx 명령에 공통된 매개변수를 간단히 설명합니다. 매개변수 및 지원되는 값에 대한 구문 및 자세한 설명은 SQL 프로그래밍 개념 책을 참조하십시오.

RDB

SQL 패키지 옵션을 작성할 관계형 데이터베이스 이름을 지정합니다. *NONE이 지정되면 프로그램이나 모듈은 분산 오브젝트가 아니며 SQL 패키지 작성(CRTSQLPKG) 명령을 사용할 수 없습니다. 관계형 데이터베이스 이름은 로컬 데이터베이스 이름이 될 수 있습니다.

RDBCNNMTH

CONNECT 명령문에 사용되는 의미의 유형, 즉 리모트 작업 단위(RUW) 또는 분산 작업 단위(DUW) 의미를 지정합니다.

SQLPKG

SQL 패키지의 이름과 라이브러리를 지정합니다.

USER

대화를 시작할 때 리모트 서버에 송신하는 사용자 이름을 지정합니다. 이 매개변수는 대화가 사전컴파일 프로세스의 부분으로 시작될 경우에만 사용됩니다.

PASSWORD

대화를 시작할 때 리모트 서버에 사용될 암호를 지정합니다. 이 매개변수는 대화가 사전컴파일 프로세스의 부분으로 시작될 경우에만 사용됩니다.

REPLACE

사전컴파일 프로세스의 부분으로 작성된 오브젝트가 기존 오브젝트를 대체해야 할 경우에 지정합니다.

다음 예는 INVENT라는 COBOL 프로그램을 작성하고 SPIFFY라는 라이브러리에 저장합니다. SQL 명령 규칙이 선택되고 지정된 표에서 선택된 모든 행은 회복 단위가 종료될 때까지 잠깁니다. 프로그램과 같은 이름의 SQL 패키지는 KC000라는 리모트 관계형 데이터베이스에 작성됩니다.

```
CRTSQLCBL PGM(SPIFFY/INVENT) OPTION(*SRC *XREF *SQL)
          COMMIT(*ALL) RDB(KC000)
```

어플리케이션 프로그램 컴파일

iSeries용 DB2 Universal Database 사전컴파일러는 *NOGEN 사전컴파일러 옵션을 지정하지 않는 한, 사전컴파일러가 성공적으로 완료된 후에 호스트 언어 컴파일러를 자동으로 호출합니다. 컴파일러 명령은 프로그래밍,

소스 파일 이름, 소스 멤버 이름이 작성된 사전컴파일러, 텍스트 및 사용자 프로파일 지정이 실행됩니다. 다른 매개변수는 호스트 언어에 따라 컴파일러로 전달되기도 합니다.

이 매개변수에 대한 자세한 내용은 iSeries Information Center의 SQL 프로그래밍 개념 주제를 참조하십시오.

어플리케이션 바인딩

어플리케이션 프로그램을 실행하기 전에 프로그램, 참조하는 표 및 뷰 사이의 관계를 설정해야 합니다. 이 프로세스를 바인딩이라고 합니다. 바인딩의 결과는 액세스 계획입니다. 액세스 계획은 각 SQL 요구를 만족시키는 데 필요한 조치를 설명하는 제어 구조입니다. 액세스 계획에는 프로그램과 프로그램이 사용할 자료에 대한 정보가 들어 있습니다. 분산 관계형 데이터베이스(RDB) 작업의 경우, 액세스 계획은 SQL 패키지에 저장되며 SQL 패키지와 함께 서버가 관리합니다. SQL 패키지에 대한 자세한 정보는 220 페이지의 『SQL 패키지에 대한 작업』을 참조하십시오.

SQL은 성공적인 컴파일의 결과가 프로그램이나 서비스 프로그램 오브젝트일 때 액세스 계획을 자동으로 바인딩하고 작성하도록 시도합니다. 컴파일이 성공하지 못하거나 컴파일 결과가 모듈 오브젝트일 경우에는 액세스 계획이 작성되지 않습니다. 실행 시간에 데이터베이스 관리자가 액세스 계획이 유효하지 않다거나, 성능을 개선할 수 있는 데이터베이스에 대한 변경이 발생(예: 색인의 추가)했음을 감지하면 새 액세스 계획이 자동으로 작성됩니다. 어플리케이션 서버(AS)가 iSeries 서버가 아니면 SQL 패키지 작성(CRTSQLPKG) 명령을 사용하여 바인딩을 다시 실행해야 합니다. 바인딩은 세 가지 사항을 실행합니다.

- 데이터베이스의 설명을 사용하여 SQL문 재확인.

바인딩이 처리되는 동안 SQL문은 표, 뷰 및 열 이름이 유효한지 확인합니다. 표나 뷰에 대한 참조가 사전 컴파일이나 컴파일할 때 존재하지 않으면 확인은 실행 시간에 실행됩니다. 표나 뷰가 실행 시간에 존재하지 않으면 음수 SQLCODE가 리턴됩니다.

- 사용자 프로그램이 처리할 자료에 액세스하는 데 필요한 액세스 경로를 선택.

SQL이 액세스 계획을 빌드할 때 액세스 경로, 색인, 표 크기 및 기타 요소의 선택을 고려합니다. 바인딩 프로세스는 자료를 액세스하는 데 사용할 수 있는 모든 색인을 참고하고 자료에 대한 경로를 선택할 때 사용할 색인을 결정합니다.

- 액세스 계획 빌드 시도.

모든 SQL문이 유효하면 바인딩 프로세스는 프로그램에 액세스 계획을 빌드하고 저장합니다.

사용자 프로그램이 액세스하는 표나 뷰의 특성이 변경되면 액세스 계획은 더 이상 유효하지 않습니다. 유효하지 않은 액세스 계획을 사용하려고 하면 서버는 자동으로 액세스 계획의 재설정을 시도합니다. 액세스 계획을 재설정할 수 없으면 음수 SQLCODE가 리턴됩니다. 이 경우에 사용자는 상황을 수정하기 위해 프로그램의 SQL 문을 변경하고 CRTSQLxxx 명령을 재발행해야 하는 경우가 있습니다.

예를 들어, 프로그램에 TABLEA의 COLUMNNA를 참조하는 SQL문이 들어 있고 COLUMNNA가 더 이상 존재하지 않도록 TABLEA를 삭제하고 다시 작성하면, 프로그램을 호출할 때 COLUMNNA가 더 이상 존재하지 않기 때문에 자동 리바인드는 성공하지 못합니다. 프로그램 소스를 변경하고 CRTSQLxxx 명령을 재발행해야 합니다.

테스트 및 디버깅

분산 SQL 프로그램의 테스트와 디버깅은 로컬 SQL 프로그램의 테스트와 디버깅과 유사하지만 프로세스의 특정 측면은 다릅니다.

결국, 테스트 용으로 하나 이상의 서버가 필요하게 됩니다. 프로그램을 다시 컴파일하거나, 프로그램에 대한 입력 매개변수를 변경하거나, 프로그램 소스를 약간 수정하여 관계형 데이터베이스명을 쉽게 변경할 수 있도록 어플리케이션을 코딩하면, 하나의 서버를 사용하여 대부분의 테스트를 실행할 수 있습니다.

로컬 자료에 대해 프로그램을 테스트한 후에 프로그램은 분산 관계형 데이터베이스(RDB) 네트워크의 최종 테스트에 사용될 수 있습니다. 분산 환경으로 테스트를 이동시킬 때 프로그램만 이동시키면 되도록 리모트 연결에서 어플리케이션을 테스트할 때 어플리케이션 서버(AS)가 있는 서버에서 로컬로 어플리케이션을 테스트하는 것을 참고하십시오.

분산 SQL 프로그램의 디버깅은 로컬 SQL 프로그램 디버깅과 같은 기술을 사용합니다. 디버그 시작(STRDBG) 명령을 사용하여 디버거를 시작하고 어플리케이션을 디버그 모드에 둡니다. 중단점을 추가하고, 명령문을 추적하거나 변수의 내용을 표시할 수 있습니다.

그러나, 분산 SQL 프로그램을 디버그하려면 UPDPROD 매개변수에 대해 *YES의 값을 지정해야 합니다. 그 이유는 OS/400 분산 관계형 데이터베이스(RDB) 지원이 라이브러리 QSYS에 있는 파일을 사용하고 QSYS는 기간계 라이브러리이기 때문입니다. 기간계 라이브러리에 있는 자료를 어플리케이션 리퀘스터(AR)에서 변경할 수 있습니다. AR에서 디버그 시작(STRDBG) 명령을 발행하면 AR 작업이 디버그 모드로 되어 AR에서의 자료 조작 기능은 변경되지 않습니다.

AR의 디버그 모드에서 각 SQL문을 실행하기 위해 작업 기록부에 정보 메시지를 입력합니다. 이 메시지는 각 SQL문의 결과에 대한 정보를 제공합니다. SQL 리턴 코드 리스트와 분산 관계형 데이터베이스(RDB)의 오류 메시지 리스트는 제 9 장 『분산 관계형 데이터베이스(DRDB) 처리』에 제공됩니다.

또한 서버가 SQL문의 처리 효율을 극대화하는 방법에 대한 정보 메시지가 디버그 모드의 결과로 발행됩니다. 극대화는 AS에서 발생하기 때문에 이 메시지 유형은 AR 작업 기록부에는 나타나지 않습니다. 이 정보를 얻으려면 AS 작업을 디버그 모드로 두어야 합니다.

TCP/IP를 사용하는 경우, 서버에서 디버그 모드를 시작하는 비교적 쉬운 방법은 QRWOPTIONS 자료 영역을 사용하는 것입니다. 하지만 특정 프로그램을 이 기능으로 디버그하도록 지정할 수는 없습니다. 설정에 관한 세부사항은 QRWOPTIONS 자료 영역 사용법을 참조하십시오. 자료 영역은 디버거를 시작하는 것 뿐 아니라 작업 추적을 시작하고 작업 기록부를 요구하며 작업 출력을 표시하고 기타 다른 작업을 하는데도 사용됩니다. iSeries AR에서 QRWOPTIONS 설정도 할 수 있고 옵션이 iSeries 서버로 새도우되도록 할 수 있습니다.

AR과 AS가 모두 iSeries 서버이고 APPC로 연결되어 있으면 리모트 명령 제출(SBMRMTCMD) 명령을 사용하여 AS 작업에서 디버그 모드를 시작할 수 있습니다. 85 페이지의 『DDM 파일 설정』의 설명대로 DDM 파일을 작성하십시오. DDM 파일의 통신 정보는 액세스되는 관계형 데이터베이스에 대한 관계형 데이터베이스 디렉토리 항목의 정보와 일치해야 합니다. 그런 다음 다음 명령을 발행하십시오.

```
SBMRMTCMD CMD('STRDBG UPDPROD(*YES)') DDMFILE(ddmfile name)
```

(SBMRMTCMD) 명령은 AS 작업이 아직 존재하지 않고 해당 작업에서 디버그 모드를 시작하는 경우, AS 작업을 시작합니다. 97 페이지의 『관계형 데이터베이스 활동 모니터』에서 설명한 방법을 사용하여 AS 작업 기록부를 검사하여 작업을 찾으십시오.

보다 자세한 정보는 206 페이지의 『SQL CALL 명령문(저장된 프로시저)』을 참조하십시오.

AS 작업을 디버그 모드로 설정하는 다음 방법은 AR과 iSeries용 DB2 Universal Database AS로 작동되지만 약간의 제한이 있습니다. 이는 어플리케이션이 설정을 하기 위한 연결을 한 후 멈출 수 있는지의 여부에 따라 결정됩니다. 이는 또한 사용자가 추적 또는 디버그하려는 것이 연결이 성립된 후에 발생한다고 가정합니다.

- AS로 사인 온하고 AS 작업 찾기.
- 다음과 같은 대화식 작업(AS 작업을 찾기 위해 사용하는 작업)에서 서비스 작업 시작(STRSRVJOB) 명령을 발행.

```
STRSRVJOB (job-number/user-ID/job-name)
```

(STRSRVJOB) 명령의 작업 이름은 AS 작업의 이름입니다. 이 명령을 발행하면 AS 작업에 영향을 주는 대화식 작업에서 특정 명령을 발행할 수 있습니다. 이 명령 중 하나가 디버그 시작(STRDBG) 명령입니다.

- 대화식 작업의 UPDPROD 매개변수에 대한 *YES의 값을 사용하여 (STRDBG) 명령을 발행합니다. AS 작업을 디버그 모드로 설정하여 AS 작업 기록부에 디버그 메시지를 생성합니다.

이 디버그 세션을 종료하려면 대화식 작업을 종료하거나 서비스 작업 종료(ENDSRVJOB) 명령 다음에 디버그 종료(ENDDBG) 명령을 사용하십시오.

AS 작업은 SQL문을 실행하기 전에 디버그로 설정되어야 하기 때문에 AS에서 디버그를 설정할 시간을 허용하기 위해 어플리케이션을 변경해야 하는 경우도 있습니다. AS 작업은 어플리케이션이 AS에 연결되면 시작됩니다. AS에 연결한 후에 AS에서 디버그가 시작될 때까지 대기 상태에 들어 가도록 어플리케이션을 코딩할 수 있습니다.

오직 하나의 대기 작업만 있고 다른 클라이언트로부터의 방해가 없는 경우처럼 작업이 발생하기 전에 TCP/IP 연결로 사용될 사전시작 작업에 대비할 수 있으면, 지연할 필요가 없습니다.

프로그램 참조

프로그램을 작성할 때 OS/400 사용권 프로그램은 참조되는 컬렉션, 표, 뷰, SQL 패키지 및 색인에 대한 모든 정보를 SQL 프로그램에 있는 SQL문에 저장합니다.

프로그램 참조 표시(DSPPGMREF) 명령을 사용하여 프로그램의 모든 오브젝트 참조를 표시할 수 있습니다. SQL 명명 규칙을 사용하면 라이브러리명은 다음 세 가지 방법 중 하나로 저장됩니다.

- SQL 이름이 완전히 규정되면 컬렉션 명이 이름 식별자로 저장됩니다.
- SQL 이름이 완전히 규정되지 않고 DFTRDBCOL 매개변수가 지정되지 않으면 명령문의 권한 부여 ID가 이름 규정자로 저장됩니다.
- SQL 이름이 완전히 규정되지 않고 DFTRDBCOL 매개변수가 지정되면 DFTRDBCOL 매개변수에 지정된 컬렉션 명이 이름 규정자로 저장됩니다.

서버 명명 규칙을 사용하면 라이브러리명은 다음 세 가지 방법 중 하나로 저장됩니다.

- 오브젝트 이름이 완전히 규정되면 라이브러리 이름이 이름 식별자로 저장됩니다.
- 오브젝트 이름이 완전히 규정되지 않고 DFTRDBCOL 매개변수가 지정되지 않으면 *LIBL이 저장됩니다.
- SQL 이름이 완전히 규정되지 않고 DFTRDBCOL 매개변수가 지정되면 DFTRDBCOL 매개변수에 지정된 컬렉션 명이 이름 규정자로 저장됩니다.

SQL 패키지에 대한 작업

SQL 패키지는 분산 관계형 데이터베이스 어플리케이션이 특별히 사용하는 SQL 오브젝트입니다. **Application Server (AS)**에 있는 자료에 액세스하는 SQL문에 대한 제어 구조가 있습니다. 이 제어 구조는 어플리케이션 프로그램이 SQL문이 사용하는 자료를 요구하는 실행 시간에 AS가 사용합니다.

사용자에게는 SQL 패키지 작성용 SQL문이 없기 때문에 제어 언어(CL) 명령을 사용하여 SQL 패키지를 작성해야 합니다. SQL 패키지를 두 가지 방법으로 작성할 수 있습니다.

- RDB 매개변수에 지정된 관계형 데이터베이스 이름으로 CRTSQLxxx 명령 사용. 214 페이지의 『SQL문을 사용한 프로그램 사전컴파일』을 참조하십시오.
- SQL 패키지 작성(CRTSQLPKG) 명령 사용.

SQL 패키지를 작성하는 것 외에 다음 작업도 할 수 있습니다.

- 221 페이지의 『SQL 패키지 관리』
- DLTSQLPKG 명령을 사용하여 SQL 패키지 삭제
- SQL DROP PACKAGE 명령문 사용

SQL 패키지 작성(CRTSQLPKG) 명령 사용

어플리케이션 서버(AS)에서 SQL 패키지를 작성하는 데 DB2 UDB 조회 관리자 및 SQL 개발 킷 사용권 프로그램이 필요하지 않습니다. SQL 패키지 작성(CRTSQLPKG) 명령 사용을 입력하여 컴파일된 분산 관계형 데이터베이스(RDB) 프로그램에서 SQL 패키지를 작성할 수 있습니다. 또한 이 명령을 사용하여 이전에 작성한 SQL 패키지를 대체할 수 있습니다. 새 SQL 패키지는 RDB 매개변수로 정의된 관계형 데이터베이스에 작성됩니다. 새 SQL 패키지는 동일한 이름을 가지며 CRTSQLxxx 명령의 PKG 매개변수에 지정된 것과 같은 라이브러리에 저장됩니다.

자세한 정보는 CL 주제의 SQL 패키지 작성 명령을 참조하십시오.

SQL 패키지 관리

SQL 패키지를 작성한 후에 몇 가지 제한사항으로 iSeries 서버의 다른 오브젝트를 관리하는 것과 같은 방법으로 SQL 패키지를 관리할 수 있습니다. 패키지를 저장 및 복원하고, 다른 서버로 송신하고, 패키지에 대한 사용자 권한을 부여 및 취소할 수 있습니다. 또한 SQL 패키지 삭제(DLTSQLPKG) 명령이나 DROP PACKAGE SQL문을 입력하여 패키지를 삭제할 수 있습니다.

분산 SQL 프로그램을 작성하면 SQL 패키지의 이름과 내부 일관성 토큰이 프로그램에 저장됩니다. SQL 패키지의 이름과 일관성 토큰은 SQL 패키지를 찾고 SQL 패키지가 이 프로그램에 올바른지 확인하기 위해 실행 시간에 사용됩니다. SQL 패키지의 이름은 분산 SQL 프로그램을 실행하는 데 중요하기 때문에 SQL 패키지를 다른 라이브러리로 이동, 이름 변경, 복제 또는 복원할 수 없습니다.

DLTSQLPKG(SQL 패키지 삭제) 명령

SQL 패키지 삭제(DLTSQLPKG) 명령을 사용하여 하나 이상의 구조화 패키지를 삭제할 수 있습니다. 삭제되는 SQL 패키지가 있는 iSeries 서버에서 (DLTSQLPKG) 명령을 입력해야 합니다.

자세한 정보는 CL 주제의 SQL 패키지 삭제 명령을 참조하십시오.

사용자는 SQL 패키지에 대한 *OBJEXIST 권한과 패키지가 있는 컬렉션에 대해 적어도 *EXECUTE 권한을 가져야 합니다.

패키지를 제거하는 여러 가지 SQL 메소드가 있습니다.

- DB2 UDB 조회 관리자 및 SQL 개발 킷 사용권 프로그램이 설치되어 있으면 대화식 SQL을 사용하여 어플리케이션 서버(AS)에 연결한 다음 SQL DROP PACKAGE 명령문을 사용하여 패키지를 드롭(drop)하십시오.
- 연결하는 SQL 프로그램을 실행한 다음 패키지를 제거하십시오.
- 조회 관리를 사용하여 연결하고 패키지를 제거하십시오.

다음 명령은 SPIFFY 컬렉션에 있는 SQL 패키지 PARTS1을 삭제합니다.

```
DLTSQLPKG SQLPKG(SPIFFY/PARTS1)
```

리모트 iSeries 서버에 있는 SQL 패키지를 삭제하려면 리모트 명령 제출(SBMRMTCMD) 명령을 사용하여 리모트 서버에서 SQL 패키지 삭제(DLTSQLPKG) 명령을 실행하십시오. 또한 리모트 서버를 사인 온하는 데 표시장치 pass-through를 사용하여 SQL 패키지를 삭제할 수 있습니다. 리모트 서버가 iSeries 서버에 없으면 리모트 작업 장치 프로그램을 사용하여 해당 서버를 통과시킨 다음 해당 서버에 대해 SQL 패키지 삭제 명령을 제출하십시오.

SQL DROP PACKAGE 명령문

DROP PACKAGE 명령문에는 분산 관계형 데이터베이스(RDB)에 대한 PACKAGE 매개변수가 있습니다. 프로그램에 삽입하거나 대화식 SQL을 사용하여 DROP PACKAGE 명령문을 발행할 수 있습니다. DROP

PACKAGE 명령문을 발행하면 SQL 패키지와 설명이 어플리케이션 서버(AS)에서 삭제됩니다. 로컬 서버에 입력한 SQL 패키지 삭제(DLTSQLPKG) 명령과 결과는 같습니다. SQL 패키지에 종속적인 다른 오브젝트는 이 명령문 실행 결과 삭제되지 않습니다.

패키지를 삭제하려면 사용자는 SQL 패키지에 대해 다음 권한을 가져야 합니다.

- 참조된 콜렉션에 대한 시스템 권한 *EXECUTE
- SQL 패키지에 대한 시스템 권한 *OBJEXIST

다음은 DROP PACKAGE 명령문 발행 방법에 대한 예입니다.

```
DROP PACKAGE SPIFFY.PARTS1
```

프로그램은 현재 사용 중인 SQL 패키지에 대해 DROP PACKAGE 명령문을 발행할 수 없습니다.

부록 A. 어플리케이션 프로그래밍 예

이 부록은 RPG/400, COBOL/400 및 ILE C/400 프로그래밍 언어로 작성된 분산 관계형 데이터베이스 사용에 대한 RUW 어플리케이션 예입니다. 이 예는 기능적 스펙 TASK에 대한 분산 관계형 데이터베이스(RDB) 사용법을 나타내고 있습니다.

분산 관계형 데이터베이스에 대한 업무 요구사항 예

이 예에 있는 분산 관계형 데이터베이스(RDB) 어플리케이션은 자동차 대리점 네트워크의 재고 관리 부분입니다.

이 프로그램은 로컬 부품 재고표의 각 부품에 대한 재고 수준을 검사합니다. 이 재고 수준이 재주문점 아래인 경우에는 프로그램이 중앙에 있는 표를 검사하여 지불되지 않은 기존 주문이 있는지와 각 주문에 대해 적재해야 할 주문량을 조사합니다.

그래도 순 주문량(로컬 재고, 플러스 주문, 마이너스 선적)이 재주문점 이하인 경우에는 중앙 서버의 해당 표에 행을 삽입하여 주문을 이 부분에 놓습니다. 보고서는 로컬 서버에서 인쇄됩니다.

기술적 주의사항

확약 제어

이 프로그램은 로컬 및 리모트 논리 작업 장치(LUW) 개념을 사용합니다. 이 프로그램은 리모트 작업 장치를 사용하기 때문에, 다른 서버에서 새작업 창치를 시작하려면 먼저 서버(COMMIT)에서 현재 LUW를 닫아야 합니다.

커서 위치 재지정

LUW가 확약되고 어플리케이션이 다른 데이터베이스에 연결되면, 커서가 모두 닫히게 됩니다. 이 어플리케이션에는 다음 부품 번호를 다시 열 수 있는 부품 재고 파일을 읽는 커서가 필요합니다. 따라서 커서는 부품 번호가 현재 부품 번호 값보다 큰 지점에서 시작하여 부품 번호순으로 정의되어야 합니다.

주: 이 기술은 똑같은 부품 번호에 중복 행이 있으면 작동하지 않습니다.

이 예에 대한 자세한 정보는 다음 주제를 참조하십시오.

- 컬렉션 및 표 작성 예
- 표에 자료 삽입 예
- RPG 프로그램 예
- COBOL 프로그램 예
- C 프로그램 예
- 프로그램 출력 예

컬렉션 및 표 작성 예

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:16:50          PAGE    1
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . CRTDB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100      IDENTIFICATION DIVISION.                                03/29/92
200      PROGRAM-ID. CRTDB.                                       03/29/92
300      ENVIRONMENT DIVISION.                                     03/29/92
400      DATA DIVISION.                                         03/29/92
500      WORKING-STORAGE SECTION.                                 03/29/92
600          EXEC SQL INCLUDE SQLCA END-EXEC.                     03/29/92
700      PROCEDURE DIVISION.                                       03/29/92
800      MAIN.                                                     03/29/92
900      * -----
1000     * LOCATION TABLE                                         03/29/92
1100     * -----*/--
1200         EXEC SQL
1300             CREATE COLLECTION DRDA
1400             END-EXEC.                                           03/29/92
1500         EXEC SQL
1600             CREATE TABLE DRDA/PART_STOCK                       03/29/92
1700                 (PART_NUM CHAR(5) NOT NULL,
1800                  PART_UM CHAR(2) NOT NULL,
1900                  PART_QUANT INTEGER NOT NULL WITH DEFAULT,    03/29/92
2000                  PART_ROP INTEGER NOT NULL,                   03/29/92
2100                  PART_EOQ INTEGER NOT NULL,                   03/29/92
2200                  PART_BIN CHAR(6) NOT NULL WITH DEFAULT       03/29/92
2300                 ) END-EXEC.                                     03/29/92
2400         EXEC SQL
2500             CREATE UNIQUE INDEX DRDA/PART_STOCI                 03/29/92
2600             ON DRDA/PART_STOCK
2700             (PART_NUM ASC) END-EXEC.                             03/29/92
2800         EXEC SQL
2900             CREATE TABLE DRDA/PART_ORDER                       03/29/92
3000                 (ORDER_NUM SMALLINT NOT NULL,
3100                  ORIGIN_LOC CHAR(4) NOT NULL,
3200                  ORDER_TYPE CHAR(1) NOT NULL,
3300                  ORDER_STAT CHAR(1) NOT NULL,
3400                  NUM_ALLOC SMALLINT NOT NULL WITH DEFAULT,
3500                  URG_REASON CHAR(1) NOT NULL WITH DEFAULT,
3600                  CREAT_TIME TIMESTAMP NOT NULL,
3700                  ALLOC_TIME TIMESTAMP,
3800                  CLOSE_TIME TIMESTAMP,
3900                  REV_REASON CHAR(1)                             03/29/92
4000                 ) END-EXEC.                                     03/29/92
4100         EXEC SQL
4200             CREATE UNIQUE INDEX DRDA/PART_ORDEI                 03/29/92
4300             ON DRDA/PART_ORDER
4400             (ORDER_NUM ASC) END-EXEC.                             03/29/92
4500         EXEC SQL
4600             CREATE TABLE DRDA/PART_ORDLN                       03/29/92
4700                 (ORDER_NUM SMALLINT NOT NULL,
4800                  ORDER_LINE SMALLINT NOT NULL,
4900                  PART_NUM CHAR(5) NOT NULL,
5000                  QUANT_REQ INTEGER NOT NULL,                   03/29/92
5100                  LINE_STAT CHAR(1) NOT NULL                     03/29/92
5200                 ) END-EXEC.                                     03/29/92
5300         EXEC SQL
5400             CREATE UNIQUE INDEX PART_ORDLI                       03/29/92
5500             ON DRDA/PART_ORDLN
5600             (ORDER_NUM ASC,
5700             ORDER_LINE ASC) END-EXEC.                             03/29/92

```

그림 22. 컬렉션 및 표 작성 (1/2)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:16:50          PAGE    2
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . CRTDB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
5800          EXEC SQL                                          03/29/92
5900          CREATE TABLE DRDA/SHIPMENTLN                    03/29/92
6000          (SHIP_NUM SMALLINT NOT NULL,
6100          SHIP_LINE SMALLINT NOT NULL,
6200          ORDER_LOC CHAR(4) NOT NULL,
6300          ORDER_NUM SMALLINT NOT NULL,
6400          ORDER_LINE SMALLINT NOT NULL,
6500          PART_NUM CHAR(5) NOT NULL,
6600          QUANT_SHIP INTEGER NOT NULL,                    03/29/92
6700          QUANT_RECV INTEGER NOT NULL WITH DEFAULT        03/29/92
6800          ) END-EXEC.                                       03/29/92
6900          EXEC SQL                                          03/29/92
7000          CREATE UNIQUE INDEX SHIPMENTLI                   03/29/92
7100          ON DRDA/SHIPMENTLN                                03/29/92
7200          (SHIP_NUM ASC,                                     03/29/92
7300          SHIP_LINE ASC) END-EXEC.                           03/29/92
7400          EXEC SQL                                          03/29/92
7500          COMMIT      END-EXEC.                             03/29/92
7600          STOP RUN.                                         03/29/92
***** END OF SOURCE *****

```

그림 22. 콜렉션 및 표 작성 (2/2)

표에 자료 삽입 예

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:16:54          PAGE    1
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . INSDB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100      IDENTIFICATION DIVISION.                                03/29/92
200      PROGRAM-ID. INSDB.                                       03/29/92
300      ENVIRONMENT DIVISION.                                    03/29/92
400      DATA DIVISION.                                         03/29/92
500      WORKING-STORAGE SECTION.                                03/29/92
600          EXEC SQL INCLUDE SQLCA END-EXEC.                    03/29/92
700      PROCEDURE DIVISION.                                     03/29/92
800      MAIN.                                                    03/29/92
900
1000
1100
1200      *-----*
1300      * PART_STOCK TABLE                                     03/29/92
1400      *-----*/--
1500
1600          EXEC SQL                                           03/29/92
1700              INSERT INTO PART_STOCK                          03/29/92
1800              VALUES                                         03/29/92
1900              ('14020','EA',038,050,100,' ') END-EXEC.      03/29/92
2000          EXEC SQL                                           03/29/92
2100              INSERT INTO PART_STOCK                          03/29/92
2200              VALUES                                         03/29/92
2300              ('14030','EA',043,050,050,' ') END-EXEC.    03/29/92
2400          EXEC SQL                                           03/29/92
2500              INSERT INTO PART_STOCK                          03/29/92
2600              VALUES                                         03/29/92
2700              ('14040','EA',030,020,030,' ') END-EXEC.    03/29/92
2800          EXEC SQL                                           03/29/92
2900              INSERT INTO PART_STOCK                          03/29/92
3000              VALUES                                         03/29/92
3100              ('14050','EA',010,005,015,' ') END-EXEC.    03/29/92
3200          EXEC SQL                                           03/29/92
3300              INSERT INTO PART_STOCK                          03/29/92
3400              VALUES                                         03/29/92
3500              ('14060','EA',110,045,090,' ') END-EXEC.    03/29/92
3600          EXEC SQL                                           03/29/92
3700              INSERT INTO PART_STOCK                          03/29/92
3800              VALUES                                         03/29/92
3900              ('14070','EA',130,080,160,' ') END-EXEC.    03/29/92
4000          EXEC SQL                                           03/29/92
4100              INSERT INTO PART_STOCK                          03/29/92
4200              VALUES                                         03/29/92

```

그림 23. 자료를 포에 입력 (1/4)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:16:54          PAGE    2
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . INSDB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
4300          ('18020','EA',013,025,050,' ') END-EXEC.          03/29/92
4400          EXEC SQL          03/29/92
4500          INSERT INTO PART_STOCK          03/29/92
4600          VALUES          03/29/92
4700          ('18030','EA',015,005,010,' ') END-EXEC.          03/29/92
4800          EXEC SQL          03/29/92
4900          INSERT INTO PART_STOCK          03/29/92
5000          VALUES          03/29/92
5100          ('21010','EA',029,030,050,' ') END-EXEC.          03/29/92
5200          EXEC SQL          03/29/92
5300          INSERT INTO PART_STOCK          03/29/92
5400          VALUES          03/29/92
5500          ('24010','EA',025,020,040,' ') END-EXEC.          03/29/92
5600          EXEC SQL          03/29/92
5700          INSERT INTO PART_STOCK          03/29/92
5800          VALUES          03/29/92
5900          ('24080','EA',054,050,050,' ') END-EXEC.          03/29/92
6000          EXEC SQL          03/29/92
6100          INSERT INTO PART_STOCK          03/29/92
6200          VALUES          03/29/92
6300          ('24090','EA',030,025,050,' ') END-EXEC.          03/29/92
6400          EXEC SQL          03/29/92
6500          INSERT INTO PART_STOCK          03/29/92
6600          VALUES          03/29/92
6700          ('24100','EA',020,015,030,' ') END-EXEC.          03/29/92
6800          EXEC SQL          03/29/92
6900          INSERT INTO PART_STOCK          03/29/92
7000          VALUES          03/29/92
7100          ('24110','EA',052,050,080,' ') END-EXEC.          03/29/92
7200          EXEC SQL          03/29/92
7300          INSERT INTO PART_STOCK          03/29/92
7400          VALUES          03/29/92
7500          ('25010','EA',511,300,600,' ') END-EXEC.          03/29/92
7600          EXEC SQL          03/29/92
7700          INSERT INTO PART_STOCK          03/29/92
7800          VALUES          03/29/92
7900          ('36010','EA',013,005,010,' ') END-EXEC.          03/29/92
8000          EXEC SQL          03/29/92
8100          INSERT INTO PART_STOCK          03/29/92
8200          VALUES          03/29/92
8300          ('36020','EA',110,030,060,' ') END-EXEC.          03/29/92
8400          EXEC SQL          03/29/92
8500          INSERT INTO PART_STOCK          03/29/92
8600          VALUES          03/29/92
8700          ('37010','EA',415,100,200,' ') END-EXEC.          03/29/92
8800          EXEC SQL          03/29/92
8900          INSERT INTO PART_STOCK          03/29/92
9000          VALUES          03/29/92
9100          ('37020','EA',010,020,040,' ') END-EXEC.          03/29/92
9200          EXEC SQL          03/29/92
9300          INSERT INTO PART_STOCK          03/29/92
9400          VALUES          03/29/92
9500          ('37030','EA',154,055,060,' ') END-EXEC.          03/29/92
9600          EXEC SQL          03/29/92
9700          INSERT INTO PART_STOCK          03/29/92
9800          VALUES          03/29/92
9900          ('37040','EA',223,120,120,' ') END-EXEC.          03/29/92
10000         EXEC SQL          03/29/92
10100         INSERT INTO PART_STOCK          03/29/92
10200         VALUES          03/29/92
10300         ('43010','EA',110,020,040,' ') END-EXEC.          03/29/92
10400         EXEC SQL          03/29/92
10500         INSERT INTO PART_STOCK          03/29/92
10600         VALUES          03/29/92
10700         ('43020','EA',067,050,050,' ') END-EXEC.          03/29/92
10800         EXEC SQL          03/29/92

```

그림 23. 자료를 표에 입력 (2/4)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:16:54          PAGE    3
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . INSD8
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
10900          INSERT INTO PART_STOCK                      03/29/92
11000          VALUES                                      03/29/92
11100          ('48010','EA',032,030,060,' ') END-EXEC.    03/29/92
11200                                                  03/29/92
11300          *-----*
11400          * PART_ORDER TABLE                          03/29/92
11500          *-----*/--                                03/29/92
11600                                                  03/29/92
11700                                                  03/29/92
11800                                                  03/29/92
11900          EXEC SQL                                     03/29/92
12000          INSERT INTO PART_ORDER                       03/29/92
12100          VALUES                                      03/29/92
12200          (1,'DB2B','U','0',0,' ','1991-03-12-17.00.00',NULL,NULL,NULL) 03/29/92
12300          END-EXEC.                                    03/29/92
12400          EXEC SQL                                     03/29/92
12500          INSERT INTO PART_ORDER                       03/29/92
12600          VALUES                                      03/29/92
12700          (2,'SQLA','U','0',0,' ','1991-03-12-17.01.00', 03/29/92
12800          NULL,NULL,NULL)                               03/29/92
12900          END-EXEC.                                    03/29/92
13000          EXEC SQL                                     03/29/92
13100          INSERT INTO PART_ORDER                       03/29/92
13200          VALUES                                      03/29/92
13300          (3,'SQLA','U','0',0,' ','1991-03-12-17.02.00', 03/29/92
13400          NULL,NULL,NULL)                               03/29/92
13500          END-EXEC.                                    03/29/92
13600          EXEC SQL                                     03/29/92
13700          INSERT INTO PART_ORDER                       03/29/92
13800          VALUES                                      03/29/92
13900          (4,'SQLA','U','0',0,' ','1991-03-12-17.03.00', 03/29/92
14000          NULL,NULL,NULL)                               03/29/92
14100          END-EXEC.                                    03/29/92
14200          EXEC SQL                                     03/29/92
14300          INSERT INTO PART_ORDER                       03/29/92
14400          VALUES                                      03/29/92
14500          (5,'DB2B','U','0',0,' ','1991-03-12-17.04.00', 03/29/92
14600          NULL,NULL,NULL)                               03/29/92
14700          END-EXEC.                                    03/29/92
14800                                                  03/29/92
14900          *-----*
15000          * PART_ORDLN TABLE                          03/29/92
15100          *-----*/--                                03/29/92
15200                                                  03/29/92
15300                                                  03/29/92
15400          EXEC SQL                                     03/29/92
15500          INSERT INTO PART_ORDLN                       03/29/92
15600          VALUES                                      03/29/92
15700          (1,1,'24110',005,'0') END-EXEC.              03/29/92

```

그림 23. 자료를 표에 입력 (3/4)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:16:54          PAGE    5
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . INSDB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
15800          EXEC SQL                                          03/29/92
15900          INSERT INTO PART_ORDLN                            03/29/92
16000          VALUES                                          03/29/92
16100          (1,2,'24100',021,'0') END-EXEC.                  03/29/92
16200          EXEC SQL                                          03/29/92
16300          INSERT INTO PART_ORDLN                            03/29/92
16400          VALUES                                          03/29/92
16500          (1,3,'24090',018,'0') END-EXEC.                  03/29/92
16600          EXEC SQL                                          03/29/92
16700          INSERT INTO PART_ORDLN                            03/29/92
16800          VALUES                                          03/29/92
16900          (2,1,'14070',004,'0') END-EXEC.                  03/29/92
17000          EXEC SQL                                          03/29/92
17100          INSERT INTO PART_ORDLN                            03/29/92
17200          VALUES                                          03/29/92
17300          (2,2,'37040',043,'0') END-EXEC.                  03/29/92
17301          EXEC SQL                                          03/29/92
17302          INSERT INTO PART_ORDLN                            03/29/92
17303          VALUES                                          03/29/92
17304          (2,3,'14030',015,'0') END-EXEC.                  03/29/92
17305          EXEC SQL                                          03/29/92
17306          INSERT INTO PART_ORDLN                            03/29/92
17307          VALUES                                          03/29/92
17308          (3,2,'14030',025,'0') END-EXEC.                  03/29/92
17400          EXEC SQL                                          03/29/92
17500          INSERT INTO PART_ORDLN                            03/29/92
17600          VALUES                                          03/29/92
17700          (3,1,'43010',003,'0') END-EXEC.                  03/29/92
17800          EXEC SQL                                          03/29/92
17900          INSERT INTO PART_ORDLN                            03/29/92
18000          VALUES                                          03/29/92
18100          (4,1,'36010',013,'0') END-EXEC.                  03/29/92
18200          EXEC SQL                                          03/29/92
18300          INSERT INTO PART_ORDLN                            03/29/92
18400          VALUES                                          03/29/92
18500          (5,1,'18030',005,'0') END-EXEC.                  03/29/92
18600          03/29/92
18700          03/29/92
18800          *-----*
18900          * SHIPMENTLN TABLE                                03/29/92
19000          *-----*/
19100          03/29/92
19200          03/29/92
19300          EXEC SQL                                          03/29/92
19400          INSERT INTO SHIPMENTLN                            03/29/92
19500          VALUES                                          03/29/92
19600          (1,1,'DB2B',1,1,'24110',5,5) END-EXEC.            03/29/92
19700          EXEC SQL                                          03/29/92
19800          INSERT INTO SHIPMENTLN                            03/29/92
19900          VALUES                                          03/29/92
20000          (1,2,'DB2B',1,2,'24100',10,1) END-EXEC.          03/29/92
20100          EXEC SQL                                          03/29/92
20200          INSERT INTO SHIPMENTLN                            03/29/92
20300          VALUES                                          03/29/92
20400          (2,1,'SQLA',2,1,'14070',4,4) END-EXEC.            03/29/92
20500          EXEC SQL                                          03/29/92
20600          INSERT INTO SHIPMENTLN                            03/29/92
20700          VALUES                                          03/29/92
20800          (2,2,'SQLA',2,2,'37040',45,25) END-EXEC.          03/29/92
20801          EXEC SQL                                          03/29/92
20802          INSERT INTO SHIPMENTLN                            03/29/92
20803          VALUES                                          03/29/92
20804          (2,3,'SQLA',2,3,'14030', 5,5) END-EXEC.          03/29/92
20805          EXEC SQL                                          03/29/92
20806          INSERT INTO SHIPMENTLN                            03/29/92
20807          VALUES                                          03/29/92
20808          (3,1,'SQLA',2,3,'14030', 5,5) END-EXEC.          03/29/92
20900          03/29/92
21000          EXEC SQL COMMIT END-EXEC.                          03/29/92
21100          STOP RUN.                                          03/29/92
* * * * * E N D   O F   S O U R C E * * * * *

```

그림 23. 자료를 표에 입력 (4/4)

RPG 프로그램 예

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.


```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    1
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100 *****
200 *
300 * DESCRIPTIVE NAME = D-DB SAMPLE APPLICATION *
400 * REORDER POINT PROCESSING *
500 * AS/400 *
600 *
700 * FUNCTION = THIS MODULE PROCESS THE PART_STOCK TABLE AND *
800 * FOR EACH PART BELOW THE ROP (REORDER POINT) *
900 * CREATES A SUPPLY ORDER AND PRINTS A REPORT. *
1000 *
1100 *
1200 * INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION: *
1300 *
1400 * LOCADB LOCAL DB NAME *
1500 * REMODB REMOTE DB NAME *
1600 *
1700 * TABLES = PART-STOCK - LOCAL *
1800 * PART_ORDER - REMOTE *
1900 * PART_ORDLN - REMOTE *
2000 * SHIPMENTLN - REMOTE *
2100 *
2200 * INDICATORS = *IN89 - '0' ORDER HEADER NOT DONE *
2300 * '1' ORDER HEADER IS DONE *
2400 * *IN99 - '1' ABNORMAL END (SQLCOD<0) *
2500 *
2600 * TO BE COMPILED WITH COMMIT(*CHG) RDB(remotedbname) *
2700 *
2800 * INVOKE BY : CALL DDBPT6RG PARM(localdbname remotedbname) *
2900 *
3000 * CURSORS WILL BE CLOSED IMPLICITLY (BY CONNECT) BECAUSE *
3100 * THERE IS NO REASON TO DO IT EXPLICITLY *
3200 *
3300 *****
3400 *
3500 FQPRINT 0 F 33 OF PRINTER
3600 F*
3700 I*
3800 IMISC DS
3900 I B 1 20SHORTB
4000 I B 3 60LONGB
4100 I B 7 80INDNUL
4200 I 9 13 PRTTBL
4300 I 14 29 LOCTBL
4400 I I 'SQLA' 30 33 LOC
4500 I*
4600 I*
4700 C*
4800 C *LIKE DEFN SHORTB NXTORD NEW ORDER NR
4900 C *LIKE DEFN SHORTB NXTORL ORDER LINE NR
5000 C *LIKE DEFN SHORTB RTCOD1 RTCOD NEXT_PART
5100 C *LIKE DEFN SHORTB RTCOD2 RTCOD NEXT_ORD_
5200 C *LIKE DEFN SHORTB CURORD ORDER NUMBER
5300 C *LIKE DEFN SHORTB CURORL ORDER LINE
5400 C *LIKE DEFN LONGB QUANTI FOR COUNTING
5500 C *LIKE DEFN LONGB QTYSTC QTY ON STOCK
5600 C *LIKE DEFN LONGB QTYORD REORDER QTY

```

그림 24. RPG 프로그램 예 (1/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    2
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
5700 C          *LIKE   DEFN LONGB  QTYROP          REORDER POINT 03/29/92
5800 C          *LIKE   DEFN LONGB  QTYREQ          QTY ORDERED   03/29/92
5900 C          *LIKE   DEFN LONGB  QTYREC          QTY RECEIVED  03/29/92
6000 C*
6100 C*
6200 C*****
6300 C*   PARAMETERS                                     *           03/29/92
6400 C*****
6500 C*
6600 C          *ENTRY  PLIST
6700 C          PARM          LOCADB 18          LOCAL DATABASE 03/29/92
6800 C          PARM          REMODB 18          REMOTE DATABASE 03/29/92
6900 C*
7000 C*
7100 C*****
7200 C*   SQL CURSOR DECLARATIONS                         *           03/29/92
7300 C*****
7400 C*
7500 C* NEXT PART WHICH STOCK QUANTITY IS UNDER REORDER POINTS QTY 03/29/92
7600 C/EXEC SQL
7700 C+   DECLARE NEXT_PART CURSOR FOR
7800 C+       SELECT PART_NUM,
7900 C+           PART_QUANT,
8000 C+           PART_ROP,
8100 C+           PART_EQQ
8200 C+       FROM PART_STOCK
8300 C+       WHERE PART_ROP > PART_QUANT
8400 C+           AND PART_NUM > :PRTTBL
8500 C+       ORDER BY PART_NUM ASC
8600 C/END-EXEC
8700 C*
8800 C* ORDERS WHICH ARE ALREADY MADE FOR CURRENT PART
8900 C/EXEC SQL
9000 C+   DECLARE NEXT_ORDER_LINE CURSOR FOR
9100 C+       SELECT A.ORDER_NUM,
9200 C+           ORDER_LINE,
9300 C+           QUANT_REQ
9400 C+       FROM PART_ORDLN A,
9500 C+           PART_ORDER B
9600 C+       WHERE PART_NUM = :PRTTBL
9700 C+           AND LINE_STAT <> 'C'
9800 C+           AND A.ORDER_NUM = B.ORDER_NUM
9900 C+           AND ORDER_TYPE = 'R'
10000 C/END-EXEC
10100 C*
10200 C*****
10300 C*   SQL RETURN CODE HANDLING                         *           03/29/92
10400 C*****
10500 C/EXEC SQL
10600 C+   WHENEVER SQLERROR GO TO DBERRO
10700 C/END-EXEC
10800 C/EXEC SQL
10900 C+   WHENEVER SQLWARNING CONTINUE
11000 C/END-EXEC

```

그림 24. RPG 프로그램 예 (2/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    3
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
11100      C*                                03/29/92
11200      C*                                03/29/92
11300      C*****                                03/29/92
11400      C*   PROCESS - MAIN PROGRAM LOGIC          *          03/29/92
11500      C*   MAIN PROCEDURE WORKS WITH LOCAL DATABASE *          03/29/92
11600      C*****                                03/29/92
11700      C*                                03/29/92
11800      C*CLEAN UP TO PERMIT RE-RUNNING OF TEST DATA 03/29/92
11900      C          EXSR CLEANU                    03/29/92
12000      C*                                03/29/92
12100      C*                                03/29/92
12200      C          RTCOD1   DOUEQ100                03/29/92
12300      C*                                03/29/92
12400      C/EXEC SQL                                03/29/92
12500      C+          CONNECT TO :LOCADB                03/29/92
12600      C/END-EXEC                                03/29/92
12700      C/EXEC SQL                                03/29/92
12800      C+          OPEN          NEXT_PART            03/29/92
12900      C/END-EXEC                                03/29/92
13000      C/EXEC SQL                                03/29/92
13100      C+          FETCH          NEXT_PART            03/29/92
13200      C+          INTO          :PRTTBL,              03/29/92
13300      C+          :QTYSTC,                            03/29/92
13400      C+          :QTYROP,                            03/29/92
13500      C+          :QTYORD                            03/29/92
13600      C/END-EXEC                                03/29/92
13700      C          MOVE SQLCOD   RTCOD1                03/29/92
13800      C/EXEC SQL                                03/29/92
13900      C+          COMMIT                                03/29/92
14000      C/END-EXEC                                03/29/92
14100      C          RTCOD1   IFNE 100                  03/29/92
14200      C          EXSR CHECKO                        03/29/92
14300      C          ENDIF                                03/29/92
14400      C*                                03/29/92
14500      C          ENDDO                                03/29/92
14600      C*                                03/29/92
14700      C          GOTO SETLR                          03/29/92
14800      C*                                03/29/92
14900      C*                                03/29/92
15000      C*****                                03/29/92
15100      C*   SQL RETURN CODE HANDLING ON ERROR SITUATIONS *          03/29/92
15200      C*****                                03/29/92
15300      C*                                03/29/92
15400      C          DBERRO   TAG                        03/29/92
15500      C*          *-----*                          03/29/92
15600      C          EXCPTERRLIN                          03/29/92
15700      C          MOVE *ON          *IN99              03/29/92
15800      C/EXEC SQL                                03/29/92
15900      C+          WHENEVER SQLERROR CONTINUE          03/29/92
16000      C/END-EXEC                                03/29/92

```

그림 24. RPG 프로그램 예 (3/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    4
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
16100      C/EXEC SQL                                03/29/92
16200      C+          ROLLBACK                        03/29/92
16300      C/END-EXEC                                03/29/92
16400      C/EXEC SQL                                03/29/92
16500      C+          WHENEVER SQLERROR GO TO DBERRO  03/29/92
16600      C/END-EXEC                                03/29/92
16700      C*                                          03/29/92
16800      C*                                          03/29/92
16900      C          SETLR      TAG                  03/29/92
17000      C*          *-----*                      03/29/92
17100      C/EXEC SQL                                03/29/92
17200      C+          CONNECT  RESET                  03/29/92
17300      C/END-EXEC                                03/29/92
17400      C          MOVE *ON      *INLR              03/29/92
17500      C*                                          03/29/92
17600      C*****                                03/29/92
17700      C*    THE END OF THE PROGRAM                  *          03/29/92
17800      C*****                                03/29/92
17900      C*                                          03/29/92
18000      C*                                          03/29/92
18100      C*****                                03/29/92
18200      C* SUBROUTINES TO WORK WITH REMOTE DATABASES *          03/29/92
18300      C*****                                03/29/92
18400      C*                                          03/29/92
18500      C*                                          03/29/92
18600      C          CHECKO    BEGSR                  03/29/92
18700      C*          *-----*                      03/29/92
18800      C*****                                03/29/92
18900      C* CHECKS WHAT IS CURRENT ORDER AND SHIPMENT STATUS FOR THE PART *          03/29/92
19000      C* IF ORDERED AND SHIPPED IS LESS THAN REORDER POINT OF PART, *          03/29/92
19100      C* PERFORMS A SUBROUTINE WHICH MAKES AN ORDER. *          03/29/92
19200      C*****                                03/29/92
19300      C*                                          03/29/92
19400      C          MOVE 0          RTCOD2              03/29/92
19500      C          MOVE 0          QTYREQ              03/29/92
19600      C          MOVE 0          QTYREC              03/29/92
19700      C*                                          03/29/92
19800      C/EXEC SQL                                03/29/92
19900      C+          CONNECT  TO :REMOB              03/29/92
20000      C/END-EXEC                                03/29/92
20100      C/EXEC SQL                                03/29/92
20200      C+          OPEN      NEXT_ORDER_LINE        03/29/92
20300      C/END-EXEC                                03/29/92
20400      C*                                          03/29/92
20500      C          RTCOD2    DOWNE100              03/29/92
20600      C*                                          03/29/92
20700      C/EXEC SQL                                03/29/92
20800      C+          FETCH    NEXT_ORDER_LINE        03/29/92
20900      C+          INTO     :CURORD,                03/29/92

```

그림 24. RPG 프로그램 예 (4/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    5
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
21000      C+                :CURORL,                03/29/92
21100      C+                :QUANTI                03/29/92
21200      C/END-EXEC                03/29/92
21300      C*                03/29/92
21400      C          SQLCOD  IFEQ 100                03/29/92
21500      C          MOVE 100      RTCOD2            03/29/92
21600      C          ELSE                03/29/92
21700      C          ADD  QUANTI  QTYREQ            03/29/92
21800      C*                03/29/92
21900      C/EXEC SQL                03/29/92
22000      C+          SELECT  SUM(QUANT_RECV)        03/29/92
22100      C+          INTO    :QUANTI:INDNUL
22200      C+          FROM    SHIPMENTLN            03/29/92
22300      C+          WHERE   ORDER_LOC = :LOC        03/29/92
22400      C+          AND    ORDER_NUM = :CURORD      03/29/92
22500      C+          AND    ORDER_LINE = :CURORL     03/29/92
22600      C/END-EXEC                03/29/92
22700      C*                03/29/92
22800      C          INDNUL  IFGE 0                03/29/92
22900      C          ADD  QUANTI  QTYREC            03/29/92
23000      C          ENDIF                03/29/92
23100      C*                03/29/92
23200      C          ENDIF                03/29/92
23300      C          ENDDO                03/29/92
23400      C*                03/29/92
23500      C/EXEC SQL                03/29/92
23600      C+          CLOSE NEXT_ORDER_LINE          03/29/92
23700      C/END-EXEC                03/29/92
23800      C*                03/29/92
23900      C          QTYSTC  ADD  QTYREQ  QUANTI    03/29/92
24000      C          SUB    QUANTI  QTYREC          03/29/92
24100      C*                03/29/92
24200      C          QTYROP  IFGT QUANTI            03/29/92
24300      C          EXSR ORDERP                    03/29/92
24400      C          ENDIF                03/29/92
24500      C*                03/29/92
24600      C/EXEC SQL                03/29/92
24700      C+          COMMIT                03/29/92
24800      C/END-EXEC                03/29/92
24900      C*                03/29/92
25000      C          ENDSR                CHECKO      03/29/92
25100      C*                03/29/92
25200      C*                03/29/92
25300      C          ORDERP  BEGSR                03/29/92
25400      C*          *-----*                03/29/92

```

그림 24. RPG 프로그램 예 (5/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    7
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
25500 C*****
25600 C* MAKES AN ORDER. IF FIRST TIME, PERFORMS THE SUBROUTINE, WHICH *          03/29/92
25700 C* SEARCHES FOR NEW ORDER NUMBER AND MAKES THE ORDER HEADER. *          03/29/92
25800 C* AFTER THAT MAKES ORDER LINES USING REORDER QUANTITY FOR THE *          03/29/92
25900 C* PART. FOR EVERY ORDERED PART WRITES A LINE ON REPORT. *          03/29/92
26000 C*****
26100 C*
26200 C          *IN89      IFEQ *OFF          FIRST ORDER ?          03/29/92
26300 C          EXSR STRORD          03/29/92
26400 C          MOVE *ON          *IN89      ORD.HEAD.DONE          03/29/92
26500 C          EXCPHEADER          WRITE HEADERS          03/29/92
26600 C          ENDIF          03/29/92
26700 C*
26800 C          ADD 1          NXTORL          NEXT ORD.LIN          03/29/92
26900 C/EXEC SQL          03/29/92
27000 C+          INSERT          03/29/92
27100 C+          INTO          PART_ORDLN          03/29/92
27200 C+          (ORDER_NUM,          03/29/92
27300 C+          ORDER_LINE,          03/29/92
27400 C+          PART_NUM,          03/29/92
27500 C+          QUANT_REQ,          03/29/92
27600 C+          LINE_STAT)          03/29/92
27700 C+          VALUES (:NXTORD,          03/29/92
27800 C+          :NXTORL,          03/29/92
27900 C+          :PRTTBL,          03/29/92
28000 C+          :QTYORD,          03/29/92
28100 C+          '0')          03/29/92
28200 C/END-EXEC          03/29/92
28300 C*
28400 C          *INOF      IFEQ *ON          03/29/92
28500 C          EXCPHEADER          03/29/92
28600 C          END          03/29/92
28700 C          EXCPTDETAIL          03/29/92
28800 C*
28900 C          ENDSR          ORDERP          03/29/92
29000 C*
29100 C*
29200 C          STRORD      BEGSR          03/29/92
29300 C*          *-----*          03/29/92
29400 C*****
29500 C* SEARCHES FOR NEXT ORDER NUMBER AND MAKES AN ORDER HEADER *          03/29/92
29600 C* USING THAT NUMBER. WRITES ALSO HEADERS ON REPORT. *          03/29/92
29700 C*****
29800 C*
29900 C/EXEC SQL          03/29/92
30000 C+          SELECT      (MAX(ORDER_NUM) + 1)          03/29/92
30100 C+          INTO          :NXTORD          03/29/92
30200 C+          FROM          PART_ORDER          03/29/92
30300 C/END-EXEC          03/29/92

```

그림 24. RPG 프로그램 예 (6/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE    8
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
30400      C/EXEC SQL
30500      C+          INSERT
30600      C+          INTO    PART_ORDER
30700      C+          (ORDER_NUM,
30800      C+          ORIGIN_LOC,
30900      C+          ORDER_TYPE,
31000      C+          ORDER_STAT,
31100      C+          CREAT_TIME)
31200      C+          VALUES (:NXTORD,
31300      C+          :LOC,
31400      C+          'R',
31500      C+          'O',
31600      C+          CURRENT_TIMESTAMP)
31700      C/END-EXEC
31800      C          ENDSR          STRORD
31900      C*
32000      C*
32100      C          CLEANU    BEGSR
32200      C*          *-----*
32300      C*****
32400      C* THIS SUBROUTINE IS ONLY REQUIRED IN A TEST ENVIRONMENT
32500      C* TO RESET THE DATA TO PERMIT RE-RUNNING OF THE TEST
32600      C*****
32700      C*
32800      C/EXEC SQL
32900      C+          CONNECT TO :REMOB
33000      C/END-EXEC
33100      C/EXEC SQL
33200      C+          DELETE
33300      C+          FROM    PART_ORDLN
33400      C+          WHERE   ORDER_NUM IN
33500      C+          (SELECT ORDER_NUM
33600      C+          FROM    PART_ORDER
33700      C+          WHERE   ORDER_TYPE = 'R')
33800      C/END-EXEC
33900      C/EXEC SQL
34000      C+          DELETE
34100      C+          FROM    PART_ORDER
34200      C+          WHERE   ORDER_TYPE = 'R'
34300      C/END-EXEC
34400      C/EXEC SQL
34500      C+          COMMIT
34600      C/END-EXEC
34700      C*
34800      C          ENDSR          CLEANU
34900      C*
35000      C*

```

그림 24. RPG 프로그램 예 (7/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:48          PAGE 9
SOURCE FILE . . . . . DRDA/QRPGSRC
MEMBER . . . . . DDBPT6RG
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
35100 C*****
35200 O* OUTPUTLINES FOR THE REPORT *
35300 O*****
35400 O*
35500 QQPRINT E 2          HEADER
35600 0                    + 0 '***** ROP PROCESSING'
35700 0                    + 1 'REPORT *****'
35800 O*
35900 QQPRINT E 2          HEADER
36000 0                    + 0 ' ORDER NUMBER = '
36100 0                    NXTORDZ + 0
36200 O*
36300 QQPRINT E 1          HEADER
36400 0                    + 0 '-----'
36500 0                    + 0 '-----'
36600 O*
36700 QQPRINT E 1          HEADER
36800 0                    + 0 ' LINE '
36900 0                    + 0 'PART '
37000 0                    + 0 'QTY '
37100 O*
37200 QQPRINT E 1          HEADER
37300 0                    + 0 ' NUMBER '
37400 0                    + 0 'NUMBER '
37500 0                    + 0 'REQUESTED '
37600 O*
37700 QQPRINT E 11         HEADER
37800 0                    + 0 '-----'
37900 0                    + 0 '-----'
38000 O*
38100 QQPRINT EF1          DETAIL
38200 0                    NXTORLZ + 4
38300 0                    PRITBL + 4
38400 0                    QTYORD1 + 4
38500 O*
38600 QQPRINT T 2          LRN99
38700 0                    + 0 '-----'
38800 0                    + 0 '-----'
38900 QQPRINT T 1          LRN99
39000 0                    + 0 'NUMBER OF LINES '
39100 0                    + 0 'CREATED = '
39200 0                    NXTORLZ + 0
39300 O*
39400 QQPRINT T 1          LRN99
39500 0                    + 0 '-----'
39600 0                    + 0 '-----'
39700 O*
39800 QQPRINT T 2          LRN99
39900 0                    + 0 '*****'
40000 0                    + 0 ' END OF PROGRAM '
40100 0                    + 0 '*****'
40200 O*
40300 QQPRINT E 2          ERRLIN
40400 0                    + 0 '** ERROR **'
40500 0                    + 0 '** ERROR **'
40600 0                    + 0 '** ERROR **'
40700 QQPRINT E 1          ERRLIN
40800 0                    + 0 '* SQLCOD:'
40900 0                    SQLCODM + 0
41000 0                    33 '* '
41100 QQPRINT E 1          ERRLIN
41200 0                    + 0 '* SQLSTATE:'
41300 0                    SQLSTT + 2
41400 0                    33 '* '
41500 QQPRINT E 1          ERRLIN
41600 0                    + 0 '** ERROR **'
41700 0                    + 0 '** ERROR **'
41800 0                    + 0 '** ERROR **'

```

그림 24. RPG 프로그램 예 (8/8)

COBOL 프로그램 예

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE 1
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100  IDENTIFICATION DIVISION.
200  *-----
300  PROGRAM-ID. DDBPT6CB.                                03/29/92
400  *****                                              03/29/92
500  *  MODULE NAME = DDBPT6CB                            03/29/92
600  *
700  *  DESCRIPTIVE NAME = D-DB SAMPLE APPLICATION
800  *  REORDER POINT PROCESSING
900  *  AS/400                                           03/29/92
1000 *  COBOL
1100 *
1200 *  FUNCTION = THIS MODULE PROCESS THE PART_STOCK TABLE AND
1300 *  FOR EACH PART BELOW THE ROP (REORDER POINT)
1400 *  CHECKS THE EXISTING ORDERS AND SHIPMENTS,        03/29/92
1500 *  CREATES A SUPPLY ORDER AND PRINTS A REPORT.      03/29/92
1600 *
1700 *  DEPENDENCIES = NONE                               03/29/92
1800 *
1900 *  INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:
2000 *
2100 *  LOCAL-DB      LOCAL DB NAME                       03/29/92
2200 *  REMOTE-DB     REMOTE DB NAME                       03/29/92
2300 *
2400 *  TABLES = PART-STOCK      - LOCAL                 03/29/92
2500 *  PART_ORDER           - REMOTE                    03/29/92
2600 *  PART_ORDLN           - REMOTE                    03/29/92
2700 *  SHIPMENTLN          - REMOTE                    03/29/92
2800 *
2900 *  CRTSQLCBL SPECIAL PARAMETERS                      03/29/92
3000 *  PGM(DDBPT6CB) RDB(remotedbname) OPTION(*APOST *APOSTSQL) 03/29/92
3100 *
3200 *  INVOKE BY : CALL DDBPT6CB PARM(localdbname remotedbname) 03/29/92
3300 *
3400 *****                                              03/29/92
3500 ENVIRONMENT DIVISION.
3600 *-----
3700 INPUT-OUTPUT SECTION.
3800 FILE-CONTROL.
3900 SELECT RELAT ASSIGN TO PRINTER-QPRINT.              03/29/92
4000 DATA DIVISION.
4100 *-----
4200 FILE SECTION.
4300 *-----                                              03/29/92
4400 FD RELAT
4500 RECORD CONTAINS 33 CHARACTERS
4600 LABEL RECORDS ARE OMITTED
4700 DATA RECORD IS REPREC.
4800 01 REPREC          PIC X(33).
4900 WORKING-STORAGE SECTION.
5000 *-----                                              03/29/92
5100 *  PRINT LINE DEFINITIONS                             03/29/92
5200 01 LINE0           PIC X(33) VALUE SPACES.
5300 01 LINE1           PIC X(33) VALUE

```

그림 25. COBOL 프로그램 예 (1/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    2
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
5400      '***** ROP PROCESSING REPORT *****'.
5500      01 LINE2.
5600      05 FILLER          PIC X(18) VALUE ' ORDER NUMBER = '.
5700      05 MASK0          PIC ZZZ9.
5800      05 FILLER          PIC X(11) VALUE SPACES.
5900      01 LINE3          PIC X(33) VALUE
6000      '-----'.
6100      01 LINE4          PIC X(33) VALUE
6200      ' LINE PART QTY '.
6300      01 LINE5          PIC X(33) VALUE
6400      ' NUMBER NUMBER REQUESTED '.
6500      01 LINE6.
6600      05 FILLER          PIC XXXX VALUE SPACES.
6700      05 MASK1          PIC ZZZ9.
6800      05 FILLER          PIC XXXX VALUE SPACES.
6900      05 PART-TABLE     PIC XXXXX.
7000      05 FILLER          PIC XXXX VALUE SPACES.
7100      05 MASK2          PIC Z,ZZZ,ZZZ.ZZ.
7200      01 LINE7.
7300      05 FILLER          PIC X(26) VALUE
7400      'NUMBER OF LINES CREATED = '.
7500      05 MASK3          PIC ZZZ9.
7600      05 FILLER          PIC XXX VALUE SPACES.
7700      01 LINE8          PIC X(33) VALUE
7800      '***** END OF PROGRAM *****'.
7900      * MISCELLANEOUS DEFINITIONS                                03/29/92
8000      01 WHAT-TIME      PIC X VALUE '1'.
8100      88 FIRST-TIME     VALUE '1'.
8200      01 CONTL          PIC S9999 COMP-4 VALUE ZEROS.                03/29/92
8300      01 CONTD          PIC S9999 COMP-4 VALUE ZEROS.                03/29/92
8400      01 RTCODE1       PIC S9999 COMP-4 VALUE ZEROS.                03/29/92
8500      01 RTCODE2       PIC S9999 COMP-4.                             03/29/92
8600      01 NEXT-NUM      PIC S9999 COMP-4.                             03/29/92
8700      01 IND-NULL      PIC S9999 COMP-4.                             03/29/92
8800      01 LOC-TABLE     PIC X(16).
8900      01 ORD-TABLE     PIC S9999 COMP-4.                             03/29/92
9000      01 ORL-TABLE     PIC S9999 COMP-4.                             03/29/92
9100      01 QUANT-TABLE   PIC S9(9) COMP-4.                             03/29/92
9200      01 QTY-TABLE     PIC S9(9) COMP-4.                             03/29/92
9300      01 ROP-TABLE     PIC S9(9) COMP-4.                             03/29/92
9400      01 EOQ-TABLE     PIC S9(9) COMP-4.                             03/29/92
9500      01 QTY-REQ       PIC S9(9) COMP-4.                             03/29/92
9600      01 QTY-REC       PIC S9(9) COMP-4.                             03/29/92
9700      * CONSTANT FOR LOCATION NUMBER                            03/29/92
9800      01 XPARAM.                                              03/29/92
9900      05 LOC          PIC X(4) VALUE 'SQLA'.                        03/29/92
10000     * DEFINITIONS FOR ERROR MESSAGE HANDLING                03/29/92
10100     01 ERROR-MESSAGE.                                        03/29/92
10200     05 MSG-ID.                                              03/29/92
10300     10 MSG-ID-1      PIC X(2)                                03/29/92
10400     VALUE 'SQ'.
10500     10 MSG-ID-2      PIC 99999.                                03/29/92

```

그림 25. COBOL 프로그램 예 (2/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    3
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
10600      *****
10700      *   SQLCA INCLUDE   *
10800      *****
10900      EXEC SQL INCLUDE SQLCA   END-EXEC.
11000
11100      LINKAGE SECTION.
11200      *-----
11300      01 LOCAL-DB          PIC X(18).
11400      01 REMOTE-DB       PIC X(18).
11500
11600      PROCEDURE DIVISION USING LOCAL-DB REMOTE-DB.
11700      *-----
11800      *****
11900      *   SQL CURSOR DECLARATION *
12000      *****
12100      * RE-POSITIONABLE CURSOR : POSITION AFTER LAST PART_NUM
12200      EXEC SQL DECLARE NEXT_PART CURSOR FOR
12300          SELECT PART_NUM,
12400                 PART_QUANT,
12500                 PART_ROP,
12600                 PART_EOQ
12700      FROM PART_STOCK
12800      WHERE PART_ROP > PART_QUANT
12900            AND PART_NUM > :PART-TABLE
13000      ORDER BY PART_NUM ASC
13100      END-EXEC.
13200      * CURSOR FOR ORDER LINES
13300      EXEC SQL DECLARE NEXT_ORDER_LINE CURSOR FOR
13400          SELECT A.ORDER_NUM,
13500                 ORDER_LINE,
13600                 QUANT_REQ
13700      FROM PART_ORDLN A,
13800            PART_ORDER B
13900      WHERE PART_NUM = :PART-TABLE
14000            AND LINE_STAT <> 'C'
14100            AND A.ORDER_NUM = B.ORDER_NUM
14200            AND ORDER_TYPE = 'R'
14300      END-EXEC.
14400      *****
14500      *   SQL RETURN CODE HANDLING*
14600      *****
14700      EXEC SQL WHENEVER SQLERROR GO TO DB-ERROR END-EXEC.
14800      EXEC SQL WHENEVER SQLWARNING CONTINUE END-EXEC.
14900
15000      MAIN-PROGRAM-PROC.
15100      *-----
15200      PERFORM START-UP THRU START-UP-EXIT.
15300      PERFORM MAIN-PROC THRU MAIN-EXIT UNTIL RTCODE1 = 100.
15400      END-OF-PROGRAM.

```

그림 25. COBOL 프로그램 예 (3/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    4
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
15500      *-----
15600      ****
15700      EXEC SQL CONNECT RESET END-EXEC.
15800      ****
15900      CLOSE RELAT.
16000      GOBACK.
16100      MAIN-PROGRAM-EXIT. EXIT.
16200      *-----
16300
16400      START-UP.
16500      *-----
16600      OPEN OUTPUT RELAT.
16700      ****
16800      EXEC SQL COMMIT END-EXEC.
16900      ****
17000      PERFORM CLEAN-UP THRU CLEAN-UP-EXIT.
17100      *****
17200      * CONNECT TO LOCAL DATABASE *
17300      *****
17400      ****
17500      EXEC SQL CONNECT TO :LOCAL-DB END-EXEC.
17600      ****
17700      START-UP-EXIT. EXIT.
17800      *-----
17900      EJECT
18000      MAIN-PROC.
18100      *-----
18200      EXEC SQL OPEN NEXT_PART END-EXEC.
18300      EXEC SQL
18400          FETCH NEXT_PART
18500          INTO :PART-TABLE,
18600              :QUANT-TABLE,
18700              :ROP-TABLE,
18800              :EOQ-TABLE
18900      END-EXEC.
19000      IF SQLCODE = 100
19100          MOVE 100 TO RTCODE1
19200          PERFORM TRAILER-PROC THRU TRAILER-EXIT
19300      ELSE
19400          MOVE 0 TO RTCODE2
19500          MOVE 0 TO QTY-REQ
19600          MOVE 0 TO QTY-REC
19700      * --- IMPLICIT "CLOSE" CAUSED BY COMMIT ---
19800      ****
19900      EXEC SQL COMMIT END-EXEC
20000      ****
20100      *****
20200      * CONNECT TO REMOTE DATABASE *
20300      *****

```

그림 25. COBOL 프로그램 예 (4/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    5
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
20400      ****                                03/29/92
20500      EXEC SQL CONNECT TO :REMOTE-DB END-EXEC          03/29/92
20600      ****                                03/29/92
20700      EXEC SQL OPEN NEXT_ORDER_LINE END-EXEC          03/29/92
20800      PERFORM UNTIL RTCODE2 = 100
20900          EXEC SQL                                03/29/92
21000              FETCH NEXT_ORDER_LINE
21100              INTO :ORD-TABLE,
21200                  :ORL-TABLE,
21300                  :QTY-TABLE
21400      END-EXEC
21500      IF SQLCODE = 100
21600          MOVE 100 TO RTCODE2
21700      EXEC SQL CLOSE NEXT_ORDER_LINE END-EXEC
21800      ELSE
21900          ADD QTY-TABLE TO QTY-REQ
22000      EXEC SQL
22100          SELECT SUM(QUANT_RECVD)                                03/29/92
22200          INTO :QTY-TABLE:IND-NULL
22300          FROM SHIPMENTLN                                03/29/92
22400          WHERE ORDER_LOC = :LOC
22500          AND ORDER_NUM = :ORD-TABLE
22600          AND ORDER_LINE = :ORL-TABLE
22700      END-EXEC
22800      IF IND-NULL NOT < 0
22900          ADD QTY-TABLE TO QTY-REC
23000      END-IF
23100      END-IF
23200      END-PERFORM
23300      IF ROP-TABLE > QUANT-TABLE + QTY-REQ - QTY-REC
23400          PERFORM ORDER-PROC THRU ORDER-EXIT
23500      END-IF
23600      END-IF.
23700      ****                                03/29/92
23800      EXEC SQL COMMIT END-EXEC.                            03/29/92
23900      ****                                03/29/92
24000      *****
24100      * RECONNECT TO LOCAL DATABASE *                        03/29/92
24200      *****
24300      ****                                03/29/92
24400      EXEC SQL CONNECT TO :LOCAL-DB END-EXEC.              03/29/92
24500      ****                                03/29/92
24600      MAIN-EXIT. EXIT.
24700      *-----
24800      ORDER-PROC.
24900      *-----
25000      IF FIRST-TIME
25100          MOVE '2' TO WHAT-TIME
25200          PERFORM CREATE-ORDER-PROC THRU CREATE-ORDER-EXIT. 03/29/92
25300      ADD 1 TO CONTL.

```

그림 25. COBOL 프로그램 예 (5/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    7
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
25400          EXEC SQL
25500              INSERT
25600                  INTO    PART_ORDLN                      03/29/92
25700                      (ORDER_NUM,
25800                        ORDER_LINE,
25900                        PART_NUM,
26000                        QUANT_REQ,
26100                        LINE_STAT)
26200              VALUES (:NEXT-NUM,
26300                        :CONTL,
26400                        :PART-TABLE,
26500                        :EQQ-TABLE,
26600                        'O')
26700          END-EXEC.
26800          PERFORM DETAIL-PROC THRU DETAIL-EXIT.
26900          ORDER-EXIT. EXIT.
27000          *-----
27100
27200          CREATE-ORDER-PROC.                                03/29/92
27300          *-----
27400          *GET NEXT ORDER NUMBER                            03/29/92
27500              EXEC SQL                                     03/29/92
27600                  SELECT (MAX(ORDER_NUM) + 1)             03/29/92
27700                  INTO  :NEXT-NUM:IND-NULL                 03/29/92
27800                  FROM  PART_ORDER                         03/29/92
27900          END-EXEC.                                        03/29/92
28000          IF IND-NULL < 0                                  03/29/92
28100              MOVE 1 TO NEXT-NUM.                          03/29/92
28200          EXEC SQL                                        03/29/92
28300              INSERT                                       03/29/92
28400                  INTO    PART_ORDER                       03/29/92
28500                      (ORDER_NUM,                          03/29/92
28600                        ORIGIN_LOC,                         03/29/92
28700                        ORDER_TYPE,                         03/29/92
28800                        ORDER_STAT,                         03/29/92
28900                        CREAT_TIME)                        03/29/92
29000              VALUES (:NEXT-NUM,                          03/29/92
29100                        :LOC, 'R', 'O',                     03/29/92
29200                        CURRENT_TIMESTAMP)                  03/29/92
29300          END-EXEC.                                        03/29/92
29400          MOVE NEXT-NUM TO MASK0.                           03/29/92
29500          PERFORM HEADER-PROC THRU HEADER-EXIT.            03/29/92
29600          CREATE-ORDER-EXIT. EXIT.                          03/29/92
29700          *-----
29800
29900          DB-ERROR.                                        03/29/92
30000          *-----
30100          PERFORM ERROR-MSG-PROC THRU ERROR-MSG-EXIT.     03/29/92
30200          *****
30300          *   ROLLBACK THE LUW *                            03/29/92

```

그림 25. COBOL 프로그램 예 (6/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    8
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
30400          *****
30500          EXEC SQL WHENEVER SQLERROR CONTINUE END-EXEC.          03/29/92
30600          ****          03/29/92
30700          EXEC SQL ROLLBACK WORK END-EXEC.          03/29/92
30800          ****          03/29/92
30900          PERFORM END-OF-PROGRAM THRU MAIN-PROGRAM-EXIT.          03/29/92
31000          * -- NEXT LINE INCLUDED TO RESET THE "GO TO" DEFAULT --          03/29/92
31100          EXEC SQL WHENEVER SQLERROR GO TO DB-ERROR END-EXEC.          03/29/92
31200
31300          ERROR-MSG-PROC.          03/29/92
31400          *-----          03/29/92
31500          MOVE SQLCODE TO MSG-ID-2.          03/29/92
31600          DISPLAY 'SQL STATE =' SQLSTATE ' SQLCODE =' MSG-ID-2.          03/29/92
31700          * -- ADD HERE ANY ADDITIONAL ERROR MESSAGE HANDLING --          03/29/92
31800          ERROR-MSG-EXIT. EXIT.          03/29/92
31900          *-----          03/29/92
32000
32100          *****          03/29/92
32200          * REPORT PRINTING *          03/29/92
32300          *****          03/29/92
32400          HEADER-PROC.          03/29/92
32500          *-----          03/29/92
32600          WRITE REPREC FROM LINE1 AFTER ADVANCING PAGE.
32700          WRITE REPREC FROM LINE2 AFTER ADVANCING 3 LINES.
32800          WRITE REPREC FROM LINE3 AFTER ADVANCING 2 LINES.
32900          WRITE REPREC FROM LINE4 AFTER ADVANCING 1 LINES.
33000          WRITE REPREC FROM LINE5 AFTER ADVANCING 1 LINES.
33100          WRITE REPREC FROM LINE3 AFTER ADVANCING 1 LINES.
33200          WRITE REPREC FROM LINE0 AFTER ADVANCING 1 LINES.
33300          HEADER-EXIT. EXIT.
33400          *-----
33500          DETAIL-PROC.
33600          *-----
33700          ADD 1 TO CONTD.
33800          IF CONTD > 50
33900             MOVE 1 TO CONTD
34000             PERFORM HEADER-PROC THRU HEADER-EXIT
34100          END-IF
34200          MOVE CONTL TO MASK1.
34300          MOVE EQ-TABLE TO MASK2.
34400          WRITE REPREC FROM LINE6 AFTER ADVANCING 1 LINES.
34500          DETAIL-EXIT. EXIT.
34600          *-----
34700          TRAILER-PROC.
34800          *-----
34900          MOVE CONTL TO MASK3.
35000          WRITE REPREC FROM LINE3 AFTER ADVANCING 2 LINES.
35100          WRITE REPREC FROM LINE7 AFTER ADVANCING 2 LINES.
35200          WRITE REPREC FROM LINE3 AFTER ADVANCING 2 LINES.
35300          WRITE REPREC FROM LINE8 AFTER ADVANCING 1 LINES.
35400          TRAILER-EXIT. EXIT.
35500          *-----

```

그림 25. COBOL 프로그램 예 (7/8)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:12:35          PAGE    8
SOURCE FILE . . . . . DRDA/QLBLSRC
MEMBER . . . . . DDBPT6CB
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
35600          *****
35700          * THIS PARAGRAPH IS ONLY REQUIRED IN A TEST ENVIRONMENT*
35800          * TO RESET THE DATA TO PERMIT RE-RUNNING OF THE TEST *
35900          *****
36000          CLEAN-UP.
36100          *-----
36200          *****
36300          * CONNECT TO REMOTE DATABASE *
36400          *****
36500          ****
36600          EXEC SQL CONNECT TO :REMOTE-DB END-EXEC.
36700          ****
36800          *-----DELETED ORDER ROWS FOR RERUNABILITY
36900          EXEC SQL
37000          DELETE
37100          FROM PART_ORDLN
37200          WHERE ORDER_NUM IN
37300          (SELECT ORDER_NUM
37400          FROM PART_ORDER
37500          WHERE ORDER_TYPE = 'R')
37600          END-EXEC.
37700          EXEC SQL
37800          DELETE
37900          FROM PART_ORDER
38000          WHERE ORDER_TYPE = 'R'
38100          END-EXEC.
38200          ****
38300          EXEC SQL COMMIT END-EXEC.
38400          ****
38500          CLEAN-UP-EXIT. EXIT.
38600          *-----
* * * * E N D O F S O U R C E * * * *

```

그림 25. COBOL 프로그램 예 (8/8)

C 프로그램 예

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.


```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:11:13          PAGE    1
SOURCE FILE . . . . . DRDA/QCSRC
MEMBER . . . . . DDBPT6C
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
100  /*****
200  /*  MODULE NAME = DDBPT6C
300  /*
400  /*  DESCRIPTIVE NAME:  D-DB SAMPLE APPLICATION
500  /*                                REORDER POINT PROCESSING
600  /*                                AS/400
700  /*                                C/400
800  /*
900  /*  FUNCTION:  THIS MODULE PROCESS THE PART_STOCK TABLE AND
1000 /*                                FOR EACH PART BELOW THE ROP (REORDER POINT)
1100 /*                                CREATES A SUPPLY ORDER.
1200 /*
1300 /*  OUTPUT:  BATCH : SPOOLFILE
1400 /*                                INTER : DISPLY
1500 /*
1600 /*  LOCAL TABLES:  PART_STOCK
1700 /*
1800 /*  REMOTE TABLES:  PART_ORDER, PART_ORDLN, SHIPMENTLN
1900 /*
2000 /*  COMPILER OPTIONS:
2100 /*  CRTSQLC  PGM(DDBPT6C) COMMIT(*CHG) RDB(rdbname)
2200 /*
2300 /*  INVOKED BY: CALL PGM(DDBPT6C) PARM('lcldbname' 'rmtdbname')
2400 /*****
2500
2600 #include <stdlib.h>
2700 #include <string.h>
2800 #include <stdio.h>
2900
3000          EXEC SQL BEGIN DECLARE SECTION;
3100
3200  char loc      [4] = "SQLA";          /* dealer's database name */
3300  char remote_db [18] = "              "; /* sample remote database */
3400  char local_db [18] = "              "; /* sample local database */
3500
3600  char part_table [5] = " ";          /* part number in table part_stock */
3700  long  quant_table;                  /* quantity in stock, tbl part_stock */
3800  long  rop_table;                    /* reorder point , tbl part_stock */
3900  long  eoq_table;                    /* reorder quantity , tbl part_stock */
4000
4100  short next_num;                    /* next order nbr,table part_order */
4200
4300  short ord_table;                    /* order nbr. , tbl order_line */
4400  short orl_table;                    /* order line , tbl order_line */
4500  long  qty_table;                    /* ordered quantity , tbl order_line */
4600  long  line_count = 0;                /* total number of order lines */
4700  short ind_null;                    /* null indicator for qty_table */
4800  short contl = 0;                    /* continuation line, tbl order_line */
4900
5000          EXEC SQL END  DECLARE SECTION;
5100          EXEC SQL INCLUDE SQLCA;
5200          EXEC SQL WHENEVER SQLERROR go to error_tag;
5300          EXEC SQL WHENEVER SQLWARNING CONTINUE;
5400

```

그림 26. C 프로그램 예 (1/5)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:11:13          PAGE    2
SOURCE FILE . . . . . DRDA/QCSRC
MEMBER . . . . . DDBPT6C
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
5500 /*****/ 03/29/92
5600 /* Other Variables */ 03/29/92
5700 /*****/ 03/29/92
5800 03/29/92
5900     char first_time, what_time; 03/29/92
6000     long qty_rec = 0, qty_req = 0; 03/29/92
6100 03/29/92
6200 /*****/ 03/29/92
6300 /* Function Declaration */ 03/29/92
6400 /*****/ 03/29/92
6500 03/29/92
6600 declare_cursor () { 03/29/92
6700 03/29/92
6800 /* SQL Cursor declaration and reposition for local UW */ 03/29/92
6900 03/29/92
7000     EXEC SQL DECLARE NEXT_PART CURSOR FOR 03/29/92
7100         SELECT PART_NUM, PART_QUANT, PART_ROP, PART_EOQ 03/29/92
7200         FROM PART_STOCK 03/29/92
7300         WHERE PART_ROP > PART_QUANT 03/29/92
7400         AND PART_NUM > :part_table 03/29/92
7500         ORDER BY PART_NUM; 03/29/92
7600 /* SQL Cursor declaration and connect for RUW */ 03/29/92
7700 03/29/92
7800     EXEC SQL DECLARE NEXT_OLINE CURSOR FOR 03/29/92
7900         SELECT A.ORDER_NUM, ORDER_LINE, QUANT_REQ 03/29/92
8000         FROM PART_ORDLN A, 03/29/92
8100             PART_ORDER B 03/29/92
8200         WHERE PART_NUM = :part_table 03/29/92
8300         AND LINE_STAT <> 'C' 03/29/92
8400         AND A.ORDER_NUM = B.ORDER_NUM 03/29/92
8500         AND ORDER_TYPE = 'R'; 03/29/92
8600 03/29/92
8700 /* upline exit function in connectable state */ 03/29/92
8800 03/29/92
8900     EXEC SQL COMMIT; 03/29/92
9000 goto function_exit; 03/29/92
9100 error_tag: 03/29/92
9200     error_function(); 03/29/92
9300 function_exit: ; 03/29/92
9400 } /* function declare_cursor */ 03/29/92
9500 03/29/92
9600 delete_for_rerun () { 03/29/92
9700 03/29/92

```

그림 26. C 프로그램 예 (2/5)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:11:13          PAGE    3
SOURCE FILE . . . . . DRDA/QCSRC
MEMBER . . . . . DDBPT6C
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
9800 /* Clean up for rerunability in test environment          */          03/29/92
9900          EXEC SQL CONNECT TO :remote_db;                  03/29/92
10000         EXEC SQL DELETE                                  03/29/92
10100         FROM PART_ORDLN                                  03/29/92
10200         WHERE ORDER_NUM IN                              03/29/92
10300         (SELECT ORDER_NUM                                03/29/92
10400         FROM PART_ORDER                                  03/29/92
10500         WHERE ORDER_TYPE = 'R');                          03/29/92
10600         EXEC SQL DELETE                                  03/29/92
10700         FROM PART_ORDER                                  03/29/92
10800         WHERE ORDER_TYPE = 'R';                          03/29/92
10900 /* upline exit function in connectable state              */          03/29/92
11000         EXEC SQL COMMIT;                                  03/29/92
11100         EXEC SQL CONNECT TO :local_db;                    03/29/92
11200 goto function_exit;                                      03/29/92
11300 error_tag:                                             03/29/92
11400         error_function();                                  03/29/92
11500 function_exit: ;                                         03/29/92
11600 } /* function delete_for_rerun                            */          03/29/92
11700                                                         03/29/92
11800 calculate_order_quantity () {                             03/29/92
11900                                                         03/29/92
12000 /* available qty = Stock qty + qty in order - qty received */          03/29/92
12100                                                         03/29/92
12200         EXEC SQL OPEN NEXT_PART;                            03/29/92
12300         EXEC SQL FETCH NEXT_PART                            03/29/92
12400         INTO :part_table, :quant_table, :rop_table, :eoq_table; 03/29/92
12500                                                         03/29/92
12600         if (sqlca.sqlcode == 100) {                         03/29/92
12700             printf("-----\n");                            03/29/92
12800             printf("NUMBER OF LINES CREATED = %d\n",line_count); 03/29/92
12900             printf("-----\n");                            03/29/92
13000             printf("***** END OF PROGRAM *****\n");    03/29/92
13100             rop_table = 0; /* no (more) orders to process */ 03/29/92
13200         }                                                   03/29/92
13300         else { qty_rec = 0;                                  03/29/92
13400             qty_req = 0;                                     03/29/92
13500 /*                                                         */          03/29/92
13600             EXEC SQL COMMIT;                                  03/29/92
13700             EXEC SQL CONNECT TO :remote_db;                 03/29/92
13800             EXEC SQL OPEN NEXT_OLINE;                       03/29/92
13900             do {                                             03/29/92
14000                 EXEC SQL FETCH NEXT_OLINE                  03/29/92
14100                 INTO :ord_table, :ori_table, :qty_table;   03/29/92
14200                 qty_rec = qty_rec + qty_table;             03/29/92
14300             } while(sqlca.sqlcode != 100);                  03/29/92
14400             EXEC SQL CLOSE NEXT_OLINE;                      03/29/92

```

그림 26. C 프로그램 예 (3/5)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:11:13          PAGE    4
SOURCE FILE . . . . . DRDA/QCSRC
MEMBER . . . . . DDBPT6C
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
14500          EXEC SQL SELECT SUM(QUANT_RECV)                                03/29/92
14600          INTO :qty_table:ind_null                                        03/29/92
14700          FROM SHIPMENTLN                                              03/29/92
14800          WHERE ORDER_LOC = :loc                                        03/29/92
14900          AND ORDER_NUM = :ord_table                                    03/29/92
15000          AND ORDER_LINE = :orl_table;                                  03/29/92
15100          if (ind_null != 0)                                            03/29/92
15200              qty_rec = qty_rec + qty_table;                              03/29/92
15300          } /* end of else branch                                        */ 03/29/92
15400 goto function_exit;                                                    03/29/92
15500 error_tag:                                                              03/29/92
15600     error_function();                                                    03/29/92
15700 function_exit: ;                                                         03/29/92
15800 } /* end of calculate_order_quantity                                    */ 03/29/92
15900                                                                           03/29/92
16000 process_order () {                                                     03/29/92
16100                                                                           03/29/92
16200 /* insert order and order_line in remote database                        */ 03/29/92
16300                                                                           03/29/92
16400     if (cont1 == 0) {                                                    03/29/92
16500                                                                           03/29/92
16600         EXEC SQL SELECT (MAX(ORDER_NUM) + 1)                              03/29/92
16700             INTO :next_num                                                03/29/92
16800             FROM PART_ORDER;                                              03/29/92
16900         EXEC SQL INSERT INTO PART_ORDER                                    03/29/92
17000             (ORDER_NUM, ORIGIN_LOC, ORDER_TYPE, ORDER_STAT, CREAT_TIME) 03/29/92
17100             VALUES (:next_num, :loc, 'R', 'O', CURRENT_TIMESTAMP);      03/29/92
17200         printf("***** ROP PROCESSING *****\n");                       03/29/92
17300         printf("ORDER NUMBER = %d \n\n",next_num);                        03/29/92
17400         printf("-----\n");                                              03/29/92
17500         printf("  LINE   PART      QTY   \n");                             03/29/92
17600         printf("  NBR     NBR      REQUESTED\n");                        03/29/92
17700         printf("-----\n");                                              03/29/92
17800         cont1 = cont1 + 1;                                                 03/29/92
17900     } /* if cont1 == 0                                                    */ 03/29/92
18000                                                                           03/29/92
18100     EXEC SQL INSERT INTO PART_ORDLN                                        03/29/92
18200         (ORDER_NUM, ORDER_LINE, PART_NUM, QUANT_REQ, LINE_STAT)          03/29/92
18300         VALUES (:next_num, :cont1, :part_table, :eqq_table, 'O');      03/29/92
18400     line_count = line_count + 1;                                          03/29/92
18500     printf("  %d      %.5s      %d\n",                                     03/29/92
18600         line_count,part_table,eqq_table);                                03/29/92
18700     cont1 = cont1 + 1;                                                    03/29/92
18800                                                                           03/29/92
18900 /* upline exit function in connectable state                            */ 03/29/92
19000     EXEC SQL COMMIT;                                                      03/29/92

```

그림 26. C 프로그램 예 (4/5)

```

5738PW1 V2R1M1 920327          SEU SOURCE LISTING          03/29/92 17:11:13          PAGE    5
SOURCE FILE . . . . . DRDA/QCSRC
MEMBER . . . . . DDBPT6C
SEQNBR*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+... 8 ...+... 9 ...+... 0
19100 /* RECONNECT TO LOCAL DATABASE */ 03/29/92
19200 EXEC SQL CONNECT TO :local_db; 03/29/92
19300 03/29/92
19400 goto function_exit; 03/29/92
19500 error_tag: 03/29/92
19600 error_function(); 03/29/92
19700 function_exit: ; 03/29/92
19800 } /* end of function process_order */ 03/29/92
19900 03/29/92
20000 error_function () { 03/29/92
20100 /* */ 03/29/92
20200 printf("*****\n"); 03/29/92
20300 printf("* SQL ERROR *\n"); 03/29/92
20400 printf("*****\n"); 03/29/92
20500 printf("SQLCODE = %d\n",sqlca.sqlcode); 03/29/92
20600 printf("SQLSTATE = %5s",sqlca.sqlstate); 03/29/92
20700 printf("\n*****\n"); 03/29/92
20800 EXEC SQL WHENEVER SQLERROR CONTINUE; 03/29/92
20900 EXEC SQL ROLLBACK; 03/29/92
21000 EXEC SQL CONNECT RESET; 03/29/92
21100 exit (999); 03/29/92
21200 } 03/29/92
21300 03/29/92
21400 main(int argc, char *argv[]) {
21500 memcpy(local_db,argv[1],strlen(argv[1]));
21600 memcpy(remote_db,argv[2],strlen(argv[2]));
21700 /* clean up */ */ 03/29/92
21800 declare_cursor(); 03/29/92
21900 delete_for_rerun(); 03/29/92
22000 03/29/92
22100 /* main-line, state is connectable */ 03/29/92
22200 03/29/92
22300 do { 03/29/92
22400 calculate_order_quantity (); 03/29/92
22500 if (rop_table > quant_table + qty_req - qty_rec) { 03/29/92
22600 process_order(); 03/29/92
22700 quant_table = qty_req = qty_rec = 0; 03/29/92
22800 } 03/29/92
22900 } while (sqlca.sqlcode == 0); 03/29/92
23000 /* RECONNECT TO APPLICATION SERVER */ 03/29/92
23100 EXEC SQL CONNECT RESET; 03/29/92
23200 exit(0); 03/29/92
23300 03/29/92
23400 03/29/92
23500 } /* end of main */ 03/29/92
* * * * E N D O F S O U R C E * * * *

```

그림 26. C 프로그램 예 (5/5)

프로그램 출력 예

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

```
***** ROP PROCESSING *****
ORDER NUMBER = 6
-----
LINE      PART      QTY
NBR      NBR      REQUESTED
-----
1         14020     100
2         14030     50
3         18020     50
4         21010     50
5         37020     40
-----
NUMBER OF LINES CREATED = 5
-----
***** END OF PROGRAM *****
```

그림 27. 프로그램 출력 예

부록 B. DRDA를 사용하는 공통 플랫폼 액세스

이 설명서는 iSeries 서버의 네트워크(유사 환경)에서 분산 관계형 데이터베이스를 지원하는 iSeries의 설명에 중점을 두고 있습니다. 많은 분산 관계형 데이터베이스 구현이 다른 DRDA 지원 플랫폼의 네트워크에 있습니다. 이 부록에서는 서로 다른 DRDA 환경에서 iSeries 서버를 사용할 때 고려해야 할 추가 정보와 기술 리스트를 제공합니다.

이 부록에서는 다른 특정 IBM 제품으로 작업할 때 고려해야 할 조건을 설명합니다. 포괄적으로 나열하지는 않습니다. 여기에서 설명하는 내용과 유사한 많은 문제나 조건은 사용하는 어플리케이션에 따라 많이 다릅니다. 다양한 IBM 플랫폼 간의 차이점에 대한 자세한 정보는 *IBM SQL Reference Volume 2*, SC26-8416 또는 *DRDA Application Programming Guide*, SC26-4773을 참조하십시오.

DRDA를 사용한 다른 플랫폼간 액세스에 대한 자세한 정보는 다음 주제를 참조하십시오.

- CCSID 고려사항
- 서로 다른 어플리케이션 서버에 대화식 SQL 및 조회 관리자 설정
- DB2 Connect 사용자 FAQ
- DB2 Connect 및 iSeries용 DB2 UDB를 사용하는 워크스테이션과의 상호 조작용을 위한 기타 추가 정보
- VM용 DB2 UDB Server에서 대화식 SQL 패키지 작성

DB2 UDB Server for VM에서 콜렉션 명은 사용자 ID와 같습니다. DB2 UDB Server for VM 어플리케이션 서버에서 대화식 SQL이나 iSeries 조회 관리자로 사용할 패키지를 작성하려면 OS/400 시스템에 QSQL400의 사용자 ID를 작성하십시오. 사용자 ID는 DB2 UDB Server for VM 어플리케이션 서버에서 필요한 모든 패키지를 작성하는 데 사용될 수 있습니다. 사용자는 OS/400의 대화식 SQL 또는 iSeries 조회 관리자를 통해 사용자 ID를 사용하여 DB2 UDB Server for VM에 액세스할 수 있습니다.

CCSID 고려사항

서로 다른 환경에서 분산 관계형 데이터베이스로 작업할 때 코드화 문자 세트 ID(CCSID)를 적절히 설정하여 사용해야 합니다. iSeries 서버는 디폴트 값으로 적재되므로 서로 다른 환경에서 작업하려면 변경해야 하는 경우도 있습니다. 또한 서버는 DB2 UDB for z/OS 및 DB2 UDB Server for VM 데이터베이스 관리자가 지원하지 않는 일부 DBCS용 CCSID를 지원합니다. 이 섹션에서는 이 두 가지 조건에 대해 설명하며 이에 대한 작업 방법을 사용자에게 제공합니다.

자세한 정보는 다음 섹션을 참조하십시오.

- iSeries 시스템 값 QCCSID
- VM 데이터베이스 관리자용 DB2 UDB 서버 및 DB2 UDB for z/OS에 대한 CCSID 변환 고려사항
- 비iSeries 어플리케이션 리퀘스터(AR)에 대한 CCSID 변환 고려사항

| 비iSeries 어플리케이션 리퀘스터(AR)에서 iSeries용 DB2 UDB 어플리케이션 서버(AS)로 연결할 때 CCSID
| 65535 태그 열은 변환되지 않습니다. 이 열이 있는 파일에 명백히 식별된 CCSID가 있는 열이 들어 있지
| 않으면 모든 문자 열의 CCSID는 다른 CCSID 값으로 변경될 수 있습니다. CCSID를 변경하려면 CHGPF(실
| 제 파일 변경) 명령을 사용하십시오. 실제 파일에 빌드된 논리 파일이 있을 경우, 오류 메시지(CPD322D)의
| 회복 섹션에 제공된 지시사항을 따르십시오.

iSeries 시스템 값 QCCSID

iSeries 서버는 QCCSID 시스템 값을 65535로 설정하여 출하됩니다. CCSID가 태그로 붙은 데이터는 수신 서버에 의해 변환되지 않습니다. 사용자의 iSeries 서버 어플리케이션 리퀘스터(AR)가 이 CCSID를 사용할 경우, 서로 다른 서버에 연결할 수 없습니다. 또한 서로 다른 서버에서 어플리케이션을 작성하는 데 이 CCSID 태그가 붙은 소스 파일을 사용할 수 없습니다.

209 페이지의 『코드화 문자 세트 ID(CCSID)』에서 설명한 것처럼 연결 시간에 사용된 CCSID는 작업 CCSID가 판별합니다. 작업이 시작되면 작업이 실행되고 있는 사용자 프로파일로 CCSID를 판별합니다. 사용자 프로파일은 디폴트로 시스템 값 QCCSID를 사용할 수 있습니다.

서버 디폴트 CCSID를 지원하지 않는 서버에 연결 중이라면 사용자의 작업 CCSID를 변경해야 합니다. 작업 변경(CHGJOB) 명령을 사용하여 작업 CCSID를 변경할 수 있습니다. 그러나 이 솔루션은 현재 작업 중인 작업에만 해당되기 때문에 다음에 작업할 때는 작업 CCSID를 다시 변경해야 합니다.

보다 영구적인 솔루션은 분산 관계형 데이터베이스에서 사용하는 사용자 프로파일이 지정한 CCSID를 변경하는 것입니다. 사용자 프로파일을 변경하면 변환된 데이터를 가져야 하는 사용자만 영향을 받습니다. iSeries용 DB2 Universal Database 어플리케이션 서버(AS)로 작업하는 경우, AS가 사용하는 사용자 프로파일을 변경해야 합니다.

| 다른 어플리케이션 리퀘스터(AR)가 작업 CCSID 65535를 사용하여 iSeries용 DB2 UDB 어플리케이션 서버
| (AS)에 연결되면 작업 디폴트 CCSID를 사용하도록 작업이 전환됩니다. 작업 디폴트 CCSID는 작업 언어
| ID(LANDID)에 의해 결정됩니다. 이 경우에는 보다 좋은 성능을 위해 작업의 CCSID를 65535이 아닌 값으
| 로 전환해야 합니다. 예를 들어 CCSID 값을 서버 작업이 실행되는 사용자 프로파일의 값으로 변경할 수 있습
| 니다.

사용자 프로파일의 디폴트 CCSID 값은 *SYSVAL 입니다. 이 값은 QCCSID 시스템값을 참조합니다. 시스템 값 변경(CHGSYSVAL) 명령으로 시스템 값과 모든 사용자 프로파일이 사용하는 디폴트 값을 변경할 수 있고 서버에 있는 대부분의 사용자(모든 사용자가 아님)를 나타내는 CCSID를 선택할 수 있습니다. 사용할 수 있는 CCSID 리스트와 CCSID가 표시하는 언어는 자국어 지원 주제를 참조하십시오.

| 작업 중인 서버가 작업이나 서버가 사용하는 CCSID를 지원하는지의 여부는 작업 기록부, SQLCA 또는 SQL
| 진단 영역에서 다음 인디케이터를 참조하십시오.

| 메시지 SQ30073

| **SQLCODE** 또는 **DB2_RETURNED_SQLCODE**

| -30073

| **SQLSTATE**

| 58017

| 텍스트 분산 자료 관리(DDM) 매개변수 X'0035'가 지원되지 않습니다.

| 메시지 SQL0332

| **SQLCODE** 또는 **DB2_RETURNED_SQLCODE**

| -332

| **SQLSTATE**

| 57017

| 텍스트 CCSID &1과 CCSID &2 사이의 전체 변환은 유효하지 않습니다.

VM 데이터베이스 관리자용 DB2 UDB 서버 및 DB2 UDB for z/OS에 대한 CCSID 변환 고려사항

iSeries용 DB2 Universal Database 및 다른 DB2* 데이터베이스 사이의 차이점 중 하나는 iSeries 시스템이 더 큰 CCSID 세트를 지원한다는 것입니다. 다른 데이터베이스 관리자가 자료(SQLCODE -332 및 SQLSTATE 57017)에서 문자 변환을 시도할 때 오류가 발생할 수 있습니다.

DB2 UDB SQL 카탈로그 표의 특정 필드는 DBCS 개방 자료 유형을 갖도록 정의될 수 있습니다. 이는 2바이트 문자 세트(DBCS)와 1바이트 문자 세트(SBCS) 문자를 모두 허용하는 자료 유형입니다. 이 필드 유형에 대한 CCSID는 서버와 함께 적재된 디폴트 CCSID를 기본으로 합니다.

DB2 UDB for z/OS 또는 DB2 UDB Server for VM 어플리케이션 리퀘스터(AR)에서 이 필드를 선택한 경우, DB2 UDB for z/OS 및 DB2 UDB Server for VM 데이터베이스가 CCSID로의 변환을 지원하지 못하기 때문에 SELECT 명령문이 실행되지 않을 수 있습니다.

이 오류를 방지하려면 DB2 UDB for z/OS 데이터베이스나 DB2 UDB Server for VM AR을 다음 중 하나와 실행되도록 변경해야 합니다.

- iSeries SQL 카탈로그 표의 DBCS-OPEN 필드와 동일한 혼합 바이트 CCSID.
- iSeries SQL 카탈로그 표에 있는 DBCS-OPEN 필드의 혼합 바이트 CCSID 자료일 경우, 서버가 자료 변환을 허용하는 CCSID. CCSID는 iSeries SQL 카탈로그 표 DBCS-OPEN 필드의 자료가 모두 1바이트 자료일 경우 1바이트 CCSID가 될 수 있습니다.

사용자 서버로 올라가 변경될 수 있도록 DB2 UDB for z/OS 또는 DB2 UDB Server for VM에서 지원하는 CCSID 변환을 일부 분석할 필요가 있습니다. 오류를 처리하는 방법에 대한 내용은 *DB2 UDB for z/OS Administration Guide*를 참조하십시오.

서로 다른 어플리케이션 서버에 대화식 SQL 및 조회 관리자 설정

대화식 SQL 및 iSeries 조회 관리자는 필요할 때 사용자의 실행 옵션(날짜 형식, 약속 제어 등)을 기초로 서로 다른 어플리케이션에 패키지를 작성합니다. 이 패키지는 어플리케이션 서버에서 QSQL400이라는 콜렉션에 작성됩니다. 패키지 이름은 QSQLabcd이며, 'abcd'는 해당 패키지에 사용되는 특정 옵션을 참조하는 번호와 일치합니다.

'abcd'에 대한 값은 다음 옵션과 부합합니다.

위치	옵션	값
a	날짜 형식	0 = ISO, JIS 날짜 형식 1 = USA 날짜 형식 2 = EUR 날짜 형식
b	시간 형식	0 = JIS 시간 형식 1 = USA 시간 형식 2 = EUR, ISO 시간 형식
c	약속 제어 십진 분리문자	0 = *CS 약속 제어 마침표 십진 분리문자 1 = *CS 약속 제어 쉼표 십진 분리문자 2 = *RR 약속 제어 마침표 십진 분리문자 3 = *RR 약속 제어 쉼표 십진 분리문자
d	STRING 분리문자 디폴트 문자 부속 유형	0 = 작은 따옴표 STRING 분리문자, 1바이트 문자 부속 유형 1 = 작은 따옴표 STRING 분리문자, 2바이트 문자 부속 유형 2 = 이중 인용부호 STRING 분리문자, 1바이트 문자 부속 유형 3 = 이중 인용부호 STRING 분리문자, 2바이트 문자 부속 유형

예를 들어, USA 날짜 형식, USA 시간 형식, *CS의 약속 제어 레벨, 십진 분리문자용 마침표, STRING 분리문자용 작은 따옴표 및 1바이트 디폴트 문자 부속 유형 등의 옵션을 사용하여 대화식 SQL로부터 서로 다른 어플리케이션 서버에 작성한 패키지의 이름은 'QSQL1100'이 될 것입니다. 특정 옵션 세트로 패키지를 작성하면 해당 어플리케이션 서버에 대해 같은 옵션을 사용하여 실행하는 모든 대화식 SQL 또는 iSeries 조회 관리자 사용자는 이 패키지를 사용합니다.

앞에서 지적했듯이 패키지를 작성할 때는 어플리케이션 서버(AS)에 대한 갱신 가능한 연결이 있어야 합니다. 패키지를 작성하기 위해 AS에 연결하기 전에 RELEASE ALL과 COMMIT을 실행해야 합니다.

1 비iSeries 어플리케이션 리퀘스터(AR) 사용자 FAQ

이 섹션에서는 비iSeries 어플리케이션 리퀘스터(AR)를 통해 iSeries 자료에 액세스하려는 워크스테이션 사용자로부터 자주 제기되는 질문에 대한 답을 제공합니다.

- 1 • 257 페이지의 『DB2 Connect에서 연결하려고 할 때 SQL5048N 메시지를 받는 이유는 무엇입니까?』
- iSeries 파일을 저널링해야 합니까?
- 조회 자료는 성능 향상을 위해 언제 블록됩니까?
- DBM SQL0969N 오류 메시지에 보고된 SQLCODE 및 연관된 토큰을 어떻게 해석합니까?

- WHERE절의 호스트 변수 유형이 성능에 어떤 영향을 미칩니까?
 - 규정되지 않은 표와 뷰 이름을 분석하기 위해 라이브러리 리스트를 사용할 수 있습니까?
 - CCSID에 대한 고려사항은 무엇입니까?
 - DB2 Connect 사용자가 일반적으로 사용되는 EBCDIC 순서 대신 iSeries 서버의 DRDA 작업 NLSS 정렬 순서표가 사용되도록 지정할 수 있습니까?
 - 조회를 실행할 때 행이 리턴되지 않는 이유는 무엇입니까?
- | • iSeries용 DB2 UDB V5R3과 대화하기 위해 필요한 DB2 UDB for Linux/Unix/Windows(LUW) 또는 DB2 Connect LUW의 레벨은 무엇입니까?
- | • DB2 Connect 버전 8에서 iSeries V5R3까지 사용 가능한 화면 이동 커서 지원을 얻는 방법은 무엇입니까?

| DB2 Connect에서 연결하려고 할 때 SQL5048N 메시지를 받는 이유는 무엇입니까?

| 메시지 SQL5048N은 데이터베이스 클라이언트의 릴리스 레벨이 데이터베이스 서버의 릴리스 레벨에 의해 지원되지 않는다고 정의합니다. 그러나 종종 메시지가 잘못될 수도 있습니다. 이 문제의 일반적인 원인은 다음과 같습니다.

- | 1. 클라이언트 어플리케이션 인에이블러만 설치한 경우 이 오류 메시지가 표시됩니다. 이러한 경우에는 게이트웨이 서버를 통해 클라이언트 시스템을 iSeries 서버에 연결해야 합니다. 직접 연결은 지원되지 않습니다.
- | 2. 또한 연결을 수동으로 구성하는 동안 다른 사람이 오류를 범한 경우에도 이 오류를 수신할 수 있습니다.

| SQL5048N 메시지를 피하려면 CCA(Client Configuration Assistant)를 사용하십시오.

| 다른 가능한 원인은 컬렉션 NULLID와 관련된 것입니다. DB2 Connect, IBM DB2 Universal Driver for SQLJ 및 JDBC 그리고 다른 어플리케이션 리퀘스터는 필요한 SQL 패키지를 빌드하는 데 컬렉션 NULLID를 사용합니다. 컬렉션과 패키지는 처음 연결될 때 작성됩니다. 사용자 프로파일에 컬렉션을 작성할 수 있는 권한이 없으면 상위 권한을 가진 다른 프로파일을 초기에 연결하여 이러한 오브젝트를 작성하도록 해야 합니다.

| 다른 원인이 있다고 생각되면 Authorized Problem Analysis Report 웹 사이트를 참조하십시오. 검색 필드에 APAR III2722를 입력하십시오.

iSeries 파일을 저널링해야 합니까?

이 질문에 대한 답은 258 페이지의 『조회 자료는 성능 향상을 위해 언제 블록됩니까?』의 질문과 밀접한 관련이 있습니다. 저널링은 클라이언트 어플리케이션이 NC(비확약, no-commit) 또는 UR(확약되지 않은 읽기, uncommitted read)의 분리 레벨을 사용 중이거나 DB2 UDB SQL 함수가 조회 자료를 블록할 수 있다고 판별한 경우에는 필요하지 않습니다. 확약 제어를 작동할 수 없으면 저널링이 필요하게 됩니다.

분리 레벨 변경 방법에 대한 예는 다음과 같습니다.

- DB2 Connect 사전검파일러는 ISOLATION UR 매개변수를 사용하여 미확약 읽기를 지정합니다.

- DB2 Connect 명령행 프로세서(CLP)는 DBM CHANGE SQLISL TO UR 명령을 사용하여 미확약 읽기를 지정합니다.
- DB2 Connect 명령행 프로세서(CLP)는 DBM CHANGE SQLISL TO NC 명령을 사용하여 미확약을 지정합니다.
- JDBC 클라이언트는 연결 등록 정보 분리 레벨을 TRANSACTION_READ_UNCOMMITTED로 설정하여 미확약 읽기를 지정합니다.

조회 자료는 성능 향상을 위해 언제 블록됩니까?

조회 자료는 다음 조건 중 어느 것도 참이 아니면 블록됩니다.

- 커서의 갱신이 가능합니다(주 1을 참조).
- 커서의 갱신이 임시적으로 가능합니다(주 2를 참조).
- SQLPREP 또는 SQLBIND에서 BLOCKING NO 사전컴파일이나 바인드 옵션이 사용된 경우.

BLOCKING NO 사전컴파일/바인드 옵션으로 단일 행 프로토콜을 강요하지 않으면 다음 두 경우 모두 블록됩니다.

- 커서가 읽기 전용인 경우(주 3 참조).
- 다음 중 모두가 참인 경우.
 - SELECT문 내에 FOR UPDATE OF절이 없습니다.
 - 프로그램 내의 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 명령문이 없습니다.
 - 프로그램에 동적 SQL문이 없거나 BLOCKING ALL이 사용되는 경우.

주:

1. 만일 읽기 전용이 아니고 다음 조건 중 하나에 해당할 경우 커서는 갱신이 가능합니다(주 3을 참조).
 - SELECT문에 FOR UPDATE OF절이 들어 있습니다.
 - 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 구가 프로그램 내에 있습니다.
2. 읽기 전용(주 3 참조)이 아니고 프로그램이 동적 명령문을 포함하고 SQLPREP 또는 SQLBIND에서 BLOCKING UNAMBIG 사전컴파일이나 바인드 옵션이 사용된 경우, 커서는 갱신가능합니다.
3. 다음 조건 중 하나 이상이 참이면 커서는 읽기 전용입니다.
 - DECLARE CURSOR 명령문이 ORDER BY절을 지정했으나 FOR UPDATE OF절은 지정하지 않았습니다.
 - DECLARE CURSOR 명령문이 FOR FETCH ONLY절을 지정했습니다.
 - 커서, 뷰 또는 커서 관련 외부 부속선택 내에 적용된 논리 파일 등에 대해서는 다음 조건 중 하나 이상이 해당됩니다.
 - 외부 부속 선택에는 DISTINCT 키워드, GROUP BY절, HAVING 절 또는 외부 부속 선택 내의 열 함수가 들어있습니다.
 - SELECT에 결합 함수가 있습니다.
 - SELECT에 UNION 오퍼레이터가 있습니다.

- SELECT에 외부 전용 부속 선택 표와 똑같은 표에 적용되는 부속 조회(subquery)가 있습니다.
- SELECT에 임시 파일로서 복사된 복잡한 논리 파일이 들어 있습니다.
- 모든 선택된 열은 표현식, 스칼라 함수 또는 상수 등입니다.
- 참조된 논리 파일의 모든 열이 입력 전용인 경우.

DBM SQL0969N 오류 메시지에 보고된 SQLCODE 및 연관된 토큰을 어떻게 해석합니까?

DB2 Connect에 사용된 클라이언트 지원은 동등하지 않은 코드를 갖는 호스트 SQLCODE와 토큰을 보고할 때 메시지 SQL0969N을 리턴합니다. 다음은 메시지 SQL0969N의 예입니다.

```
SQL0969N  There is no message text corresponding to SQL error
"-7008" in the Database Manager message file on this workstation.
The error was returned from module "QSQOPEN" with original
tokens "TABLE1  PRODLIB1  3".
```

Display Message Description(DSPMSGD) 명령을 사용하여 다음 코드와 토큰을 해석하십시오.

```
DSPMSGD SQL7008 MSGF(QSQLMSG)
```

선택 옵션 1(화면 메시지 텍스트)과 서버는 형식화된 메시지 텍스트 표시 화면을 제공합니다. 메시지의 세 개 토큰은 화면에 &1, &2 및 &3으로 표시됩니다. 메시지 예의 이유 코드는 3으로, 화면 맨 아래의 리스트에서 코드 3을 가리킵니다.

형식화된 메시지 텍스트 표시

```

시스템:  RCHASLAI
메세지 ID . . . . . :  SQL7008
메세지 파일 . . . . . :  QSQLMSG
라이브러리 . . . . . :  QSYS

메세지 . . . . . :  &2의 &1은 조작에 유효하지 않습니다.
원인 . . . . . :  이유 코드는 &3입니다.  이유 코드의 리스트는 다음과 같습니다.
-- 코드 1은 표에 멤버가 없음을 나타냅니다.
-- 코드 2는 표가 기억장치 없이 저장되었음을 나타냅니다.
-- 코드 3은 표가 저널되지 않고 표가 활약 제어 아래에 처리된
다른 표와 다른 저널로 저널되거나 저널에 대한
권한이 없음을 나타냅니다.
-- 코드 4는 표가 기간계 라이브러리에 있지만
사용자가 UPDPROD(*NO)가 있는 디버그 모드에 있어서 제품 표가
갱신되지 않음을 나타냅니다.
-- 코드 5는 표, 뷰 또는 색인이 기간계 라이브러리에 작성되지만
사용자가 UPDPROD(*NO)가 있는 디버그 모드에 있어서
표, 뷰 또는 색인이 작성되지 않음을 나타냅니다.
계속...
계속하려면 Enter를 누르십시오.

F3=종료  F11=형식화되지 않은 메시지 텍스트  F12=취소
```

WHERE절의 호스트 변수 유형이 성능에 어떤 영향을 미칩니까?

iSeries 서버의 성능 저하의 원인 중 하나는 클라이언트가 C 프로그램의 SELECT문의 WHERE절에서 비교 대상으로 부동 소수점 변수를 사용하는 것입니다. OS/400이 그 열의 자료 변환을 해야 한다면, 그 열에 대해

색인을 사용할 수 없도록 할 것입니다. 비교에 사용되는 열, 리터럴 및 호스트 변수에 대해 항상 같은 유형을 사용해야 합니다. 데이터베이스의 열이 팩(packed) 또는 존(zoned) 십진수로 정의되어 있고 호스트 변수가 다른 유형이라면 C에서 문제를 발생시킬 수 있습니다.

자세한 정보는 iSeries Information Center의 데이터베이스 성능을 위한 프로그래밍 기술 주제를 참조하십시오.

규정되지 않은 표와 뷰 이름을 분석하기 위해 라이브러리 리스트를 사용할 수 있습니까?

iSeries 서버는 DB2 Connect 제품을 사용하는 프로그램 등의 비iSeries DRDA 클라이언트 프로그램으로부터 iSeries 자료를 얻기 위해 DB2 UDB에 액세스할 때, OS/400 시스템 명명 옵션을 사용하는 제한된 기능을 지원합니다. 이전에는 상이한 DRDA 클라이언트로부터 연결할 때, SQL 명명 옵션만 사용할 수 있었습니다. 시스템 명명은 DB2 UDB for iSeries의 여러 특성을 변경합니다. 예를 들면, 다음과 같습니다.

1. 라이브러리 리스트가 규정되지 않은 표 및 뷰 이름을 분석하기 위해 검색됩니다.
2. CREATE SQL문을 실행할 때, 규정되지 않은 오브젝트는 현재 라이브러리에 작성될 것입니다.
3. 마침표(.) 대신 슬래시(/)가 규정된 오브젝트명을 이들이 상주하는 라이브러리 또는 컬렉션과 분리하기 위해 사용됩니다.
4. 어떤 권한부여 특성은 변경됩니다.

세부사항은 책 iSeries SQL 참조서의 서버 명명에 관해 읽어보십시오. 권한부여에 관련된 구현에 대한 자세한 정보는 분산 관계형 데이터베이스 계획 및 설계를 참조하십시오.

| DB2 Connect는 두 가지 프로그램 준비 명령인 사전컴파일(PREP) 명령과 (BIND) 명령에서 일반 바인드 옵션의 스펙을 지원합니다. OS/400 이름 지정은 Windows 일괄처리 파일에서 가져온 다음 예에서처럼 이들 중 하나에서 지정됩니다.

| DB2 Connect V8 이상의 경우:

```
| DB2 PREP %1.SQC BINDFILE OS400NAMING SYSTEM ...
| DB2 BIND %1.BND OS400NAMING SYSTEM ...
```

| DB2 Connect V8 이전 버전의 경우:

```
| DB2 PREP %1.SQC BINDFILE GENERIC 'OS400NAMING SYSTEM' ...
| DB2 BIND %1.BND GENERIC 'OS400NAMING SYSTEM' ...
```

| Windows 개발 플랫폼에서는 작은 따옴표(어포스트로피)가 일반 옵션인 이름/값 쌍을 둘러싸는데 사용된다는 점을 명심하십시오. AIX 또는 UNIX 플랫폼에서는 큰 따옴표가 사용됩니다.

주: iSeries V4R5 및 V5R1에서 옵션의 이름은 OS400NAMING가 아니라 AS400NAMING입니다.

SYSTEM 외에 OS400NAMING 옵션의 유일하게 유효한 값은 SQL이며 이는 디폴트 값이자 이 기능이 소개되기 전에 비iSeries 클라이언트에서 사용할 수 있는 유일한 옵션입니다.

사용자가 OS400NAMING 옵션을 (PREP) 명령이 아니라 (BIND) 명령에서 사용한다면, 서버 플랫폼이 SQL 오류를 감지하더라도 바인드 파일이 작성되어야 한다는 것을 나타내는 (PREP) 명령의 매개변수를 코딩해야 함

니다. DB2 Connect의 경우, SQLERROR CONTINUE 매개변수가 이 용도로 사용되어야 합니다. 이 기능은 '제한'으로 설명되는데 이는 어떤 경우에는 리모트 서버에서 실행하기로 되어 있는 SQL문을 클라이언트측 소프트웨어가 분석하기 때문입니다. 서버 명령에서 요구하는 대로 마침표 대신 슬래시가 스키마 ID를 표 ID와 구분하기 위해 사용될 경우, 그 명령문은 부적절한 구문을 가졌다는 이유로 거부됩니다.

DB2 Connect 사용자가 일반적으로 사용되는 EBCDIC 순서 대신 iSeries 서버의 DRDA 작업 NLSS 정렬 순서표가 사용되도록 지정할 수 있습니까?

iSeries는 일반 바인드 옵션을 인식하며 이는 프로그램을 준비하는 클라이언트가 DB2 Connect로부터 실행되도록 허용하거나 일반 바인드 옵션을 지원하는 임의의 다른 클라이언트가 iSeries 서버가 클라이언트의 요구가 실행되는 서버 작업과 연관된 NLSS 정렬 순서를 사용하는 것을 요구할 수 있도록 허용합니다. 이 기능은 V5R1의 경우 PTF SI00174를 통해 사용할 수 있습니다. 후속 릴리스의 기본 오퍼레이팅 시스템에 있습니다.

이 개선사항을 이용하기로 한다면, iSeries용 DB2 UDB에 임의의 SQL 패키지를 다시 작성해야 합니다. 왜냐하면 클라이언트 시스템으로부터 JOBRUN이라는 값을 가진 SORTSEQ 일반 바인드 옵션을 사용한 새로운 정렬 순서 옵션이 필요하기 때문입니다.

바인드 옵션은 사용자가 iSeries 서버에서 DRDA 작업의 NLSS 정렬 순서표가 흔히 사용되는 EBCDIC 순서 대신 사용되도록 지정할 수 있게 해 줍니다. 이전에는, 상이한 DRDA 클라이언트로부터 연결할 때, EBCDIC 순서가 사용되도록 하는 디폴트 *HEX 옵션만 사용할 수 있었습니다.

이 기능은 DRDA 일반 바인드 기능을 지원하는 DRDA 어플리케이션 리퀘스터에서 사용할 수 있습니다. 이는 개발 플랫폼 및 실행 환경으로 Windows에서 실행되는 DB2 Connect 6.1 FixPak 1의 제한된 테스트를 거쳤습니다. DB2 Connect는 두 가지 프로그램 준비 명령인 사전컴파일(PREP) 명령과 (BIND) 명령에서 일반 바인드 옵션의 스펙을 지원합니다. JOBRUN 정렬 순서는 Windows 일괄처리 파일에서 가져온 다음 예에 서처럼 이들 중 하나에서 지정됩니다.

```
| DB2 PREP %1.SQC BINDFILE SORTSEQ JOBRUN...
| DB2 BIND %1.BND SORTSEQ JOBRUN...
```

| 주: Windows 개발 플랫폼에서는 작은 따옴표(어포스트로피)가 일반 옵션인 이름/값 쌍을 둘러싸는데 사용되지만, AIX 또는 UNIX 플랫폼에서는 큰 따옴표가 사용됩니다.

SORTSEQ 옵션의 다른 유일한 유효한 값은 HEX이며 이는 디폴트 값이자 이 기능이 소개되기 전에 비iSeries 클라이언트에서 사용할 수 있는 유일한 옵션입니다.

조회를 실행할 때 행이 리턴되지 않는 이유는 무엇입니까?

이 문제의 한 가지 가능한 원인은 DB2 Connect 데이터베이스 통신 서비스 디렉토리에 iSeries 서버에 대한 항목을 추가하지 못했기 때문입니다.

| **iSeries용 DB2 UDB V5R3과 대화하기 위해 필요한 DB2 UDB for Linux/Unix/Windows(LUW) 또는 DB2 Connect LUW의 레벨은 무엇입니까?**

| 대화에는 다음과 같은 FixPaks가 필요합니다.

| • DB2 UDB LUW Version 7 FixPak 10

- | • DB2 UDB LUW Version 8 FixPak 4
- | • DB2 Connect LUW Version 7 FixPak 10
- | • DB2 Connect Version 8 FixPak 4

| 이러한 FixPaks는 DB2 Technical Support 웹 사이트  에서 얻을 수 있습니다.

| **DB2 Connect 버전 8에서 iSeries V5R3까지 사용 가능한 화면이동 커서 지원을 얻는 방법은 무엇입니까?**

| 클라이언트에 FixPak 4 이상을 사용해야 합니다. FixPak 4를 사용할 경우에는 다음 중 하나를 수행해야 합니다.

- | • 다음 명령을 발행하십시오.

```
| UPDATE CLI CFG FOR SECTION iSeries dbname USING CURSORTYPES 1
```

| 여기서 *iSeries dbname*은 iSeries 데이터베이스의 이름입니다.

- | • 다음 구문을 사용하여 db2cli.ini 파일을 편집하십시오.

```
| CURSORTYPES = 1
```

| **상이한 환경에서의 상호 조작에 대한 추가 정보**

다음 섹션에서는 DB2 Connect 및 DB2 UDB로 iSeries용 DB2 UDB를 사용하는 추가 정보를 제공합니다. 추가 내용은 OS/2 플랫폼에서 제품을 테스트한 경험으로 개발되었지만 제품을 포트한 모든 환경에 적용되는 것으로 알려져 있습니다.

DB2 Connect와 DB2 UDB

사용자는 DRDA 어플리케이션 서버 기능과 어플리케이션 리퀘스터(클라이언트) 기능을 실행하는 데 필요한 제품을 혼동하는 경우가 종종 있습니다. AR은 DB2 Connect로서 참조되고 AS는 DB2 Universal Database(UDB)로 참조됩니다. DB2 UDB는 다음 제품을 참조합니다.

- DB2 UDB for AIX
- DB2 UDB for HP-UX
- DB2 UDB for Linux
- DB2 UDB for Sun Solaris
- DB2 UDB for Windows

적절한 구성 및 유지보수 레벨

제품 매뉴얼에 제공된 설치 및 구성 지침을 그대로 따르십시오. 제품이 최신 레벨인지 확인하십시오. 최신 레벨이 아니면 해당 고정 팩을 적용하십시오.

포 및 콜렉션 명명

DRDA 어플리케이션이 액세스한 SQL 표의 이름은 세 부분으로 나누어집니다. 첫 번째 부분은 데이터베이스 이름이고 두 번째 부분은 컬렉션 ID이며 세 번째 부분은 기준 표 이름입니다. 처음 두 부분은 생략가능합니다. iSeries용 DB2 UDB는 컬렉션(또는 라이브러리) 이름으로 두 번째 레벨에서 표 이름을 규정합니다. 표는 iSeries용 DB2 UDB 데이터베이스에 있습니다.

V5R2 및 독립형 보조 기억장치 풀이 등장하기 전에는 각 iSeries 서버에 대해서 오직 하나의 데이터베이스만 존재했습니다. 그러나 DB2 UDB에서 표는 사용자 ID(표 작성자의 사용자 ID)로 규정되며 플랫폼의 여러 데이터베이스 중 하나에 있습니다. DB2 Connect에는 컬렉션 ID에 대해 사용자 ID를 사용하는 동일한 개념이 있습니다.

조회한 표의 이름이 컬렉션 이름없이 지정되었다면 DB2 Connect에서 iSeries용 DB2 UDB으로의 동적 조회는 디폴트 컬렉션 명에 대한(iSeries 서버에 있는) 목표 측 작업의 사용자 ID를 사용하게 됩니다. 이 방식은 사용자가 예상하지 못한 것이며 표를 찾지 못하는 원인이 될 수 있습니다.

iSeries용 DB2 UDB에서 DB2 UDB로의 동적 조회는 *qualifier.table-name* 형식으로 조회에 지정되지 않은 경우, 내포된 표 규정자를 갖고 있습니다. 두 번째 레벨 UDB 표 규정자는 조회하는 사용자의 ID를 디폴트로 합니다.

공통 사용자 ID로 DB2 UDB 데이터베이스와 표를 작성하는 경우도 있습니다. UDB의 경우에는 iSeries용 DB2 UDB에서처럼 실제 컬렉션은 없고 작성자의 사용자 ID인 표 규정자만 있음을 기억하십시오.

권한 부여

UDB 데이터베이스에 액세스하는 iSeries 서버에서 작성된 프로그램의 경우, 다음 UDB 명령을 실행해야 합니다(명령행 프로세서에서).

1. GRANT ALL PRIVILEGES ON TABLE table-name TO user(user는 'PUBLIC')
2. GRANT EXECUTE ON PACKAGE package-name(iSeries 프로그램명) TO user(user는 'PUBLIC')

APPC 통신 설정

AR로서 DB2 Connect 또는 AS로서 UDB를 사용할 때 워크스테이션에 대해 작성된 제어기와 장치로 OS/400 통신을 적절히 구성해야 합니다.

RDB 디렉토리 설정

iSeries 서버가 연결할 각 UDB 데이터베이스에 대한 RDB 디렉토리에 항목을 추가하십시오. 관계형 데이터베이스 디렉토리 항목 추가(ADDRDBDIRE) 명령을 사용하십시오. RDB 이름은 UDB 데이터베이스 이름입니다.

APPC 통신을 사용할 때 리모트 위치명은 워크스테이션 이름입니다.

TCP/IP를 사용할 때 리모트 위치명은 워크스테이션의 정의역명이거나 IP 주소입니다. UDB DRDA 서버가 사용하는 포트는 일반적으로 446이 아니라 iSeries 서버가 사용하는 DRDA 포트입니다(*DDM).

UDB 제품 설명서를 참조하여 포트 번호를 판별하십시오. 사용되는 공통 값은 50000입니다. UDB 서버에 알맞게 구성된 RDB 항목을 나타내는 DSPRDBDIRE 화면의 예는 다음과 같습니다.

```
관계형 데이터베이스 상세 표시
  관계형 데이터베이스 . . . . . : SAMPLE
리모트 위치          :
  리모트 위치 . . . . . : 9.5.36.17
  유형 . . . . . : *IP
  포트 수 또는 서비스명 . . . . . : 50000
  텍스트 . . . . . : 사용자 UDB 서버
```

DB2 Connect용 SQL 패키지 설정

iSeries용 DB2 UDB의 자료에 액세스하기 위해 DB2 Connect를 사용하기 전에 어플리케이션 프로그램과 DB2 Connect 유틸리티에 대해 iSeries 서버에 SQL 패키지를 작성해야 합니다.

삽입된 SQL이 있는 어플리케이션 프로그램 소스 파일을 처리하는 데 DB2(PREP) 명령을 사용할 수 있습니다. 이 처리는 SQL문에 대한 호스트 언어 호출이 들어 있는 수정된 소스 파일을 작성하며 디폴트로 현재 연결되어 있는 데이터베이스에 SQL 패키지를 작성합니다.

DB2 Connect를 iSeries용 DB2 UDB 서버에 바인드하려면 다음과 같이 하십시오.

1. CONNECT TO rdbname
2. Bind path@ddcs400.lst BLOCKING ALL SQLERROR CONTINUE MESSAGES DDCS400.MGS GRANT PUBLIC

위의 path@ddcs400.lst 매개변수에 있는 'path'를 디폴트 경로 C:\SQLLIB\BND\INTEL 플랫폼이 아닌 경우는 c:\sqllib/bin/) 또는 디폴트 디렉토리에 설치하지 않은 경우에는 사용자 값으로 대체하십시오.

3. CONNECT RESET

DB2 UDB에 대화식 SQL 사용

대화식 SQL을 사용하려면 OS/400에 DB2 UDB 조회 관리자 및 SQL 개발 킷 제품이 설치되어 있어야 합니다. UDB의 자료에 액세스하려면 다음과 같이 하십시오.

1. STRSQL로 세션을 시작할 때 NAMING(*SQL), DATFMT(*ISO) 및 TIMFMT(*ISO) 세션 속성을 사용하십시오. *ISO 작업 이외에 전부는 아니지만 기타 형식과 날짜 형식(DATFMT)에 사용되는 형식은 시간 형식(TIMFMT)에도 사용되어야 합니다.
2. iSeries 서버의 COLLECTION 간의 일치와 UDB에 대한 표 규정자(작성자의 사용자 ID)를 주의하십시오.
3. 첫 번째 대화식 세션에서 UDB에서 작성한 패키지를 얻으려면 SQL문을 다음 순서로 실행해야 합니다. (1) RELEASE ALL, (2) COMMIT 그리고 (3) CONNECT TO rdbname('rdbname'은 특정 데이터베이스로 대체됩니다).

대화식 SQL 사용을 위한 설정의 부분으로 다른 사용자가 대화식 SQL의 PC에 작성한 SQL PKG를 사용할 수 있도록 GRANT EXECUTE ON PACKAGE QSQL400.QSQLabcd TO PUBLIC(또는 특정 사용자)을 사용할 수도 있습니다. 위의 GRANT 명령문에서 'abcd'의 실제 값은 256 페이지의 『서로 다른 어플리케이션』

서버에 대화식 SQL 및 조회 관리자 설정』에 제시된 표에서 판별할 수 있는데, 패키지를 작성할 때 사실상 옵션의 다양한 세트에 패키지 이름을 제공합니다. 예를 들면, 패키지를 작성할 때 날짜는 *ISO, 시간은 *ISO, 화약 제어는 *CS, 스트링 분리문자는 작은 따옴표 그리고 문자 부속 유형은 1바이트의 옵션을 사용했을 경우, GRANT EXECUTE ON PACKAGE QSQL400.QSQL0200 TO some-user가 됩니다.

1 조회 닫기

DB2 Connect는 조회를 암시적 또는 명시적으로 닫을 때 읽기 잠금이 해제되도록 요청하는 옵션을 제공합니다. iSeries 서버의 경우처럼 서버가 요구를 받아들이지 않아도 오류로 간주되지 않습니다. DB2 Connect는 더 이상 읽을 행이 없는 경우 서버가 화면이동할 수 없는 커서에 대한 조회를 닫을지 여부를 지정하는 새 옵션을 제공합니다. 이전에는 서버가 이러한 결정을 했습니다. iSeries AS는 V5R3에서 이러한 새로운 피처를 지원합니다.

1 사용자 ID 및 암호 길이

어플리케이션 리퀘스터(AR)로 실행되는 iSeries용 DB2 UDB는 다른 어플리케이션 서버(AS)로 실행될 경우 10자 이상의 사용자 ID와 암호를 허용합니다. 정확한 제한치는 사용되는 특정 인터페이스에 대한 설명에서 지정됩니다. 예를 들어 SQL CONNECT문에 대한 제한치는 SQL 참조서 주제를 참조하십시오.

부록 C. 추적 작업 및 FFDC 자료 해석

이 부록은 추가 문제점 분석 정보를 제공합니다. 이 부록은 문제점을 판별할 책임이 있는 전문가에게 유용할 뿐만 아니라, iSeries 서버에 대한 연결성을 테스트하려는 분산 관계형 데이터베이스 구조를 따르도록 설계된 소프트웨어 제품의 공급자에게도 유용합니다.

이 부록에는 작업 추적으로 생긴 RW 구성요소 추적 자료에 대한 예가 추적 자료 출력 설명과 함께 수록되어 있습니다. 일부 정보는 통신 추적 자료를 해석하는 데 도움이 됩니다. 또한 이 부록에는 기억장치의 첫 번째 실패 자료 캡처 인쇄 출력의 실패에 대한 예도 설명과 함께 수록되어 있습니다.

추적 작업 및 FFDC 자료 해석의 예는 다음 주제를 참조하십시오.

- 추적 작업의 RW 구성요소에 대한 자료 항목 해석
- 첫 번째 실패 자료 캡처(FFDC)

추적 작업의 RW 구성요소에 대한 자료 항목 해석

이 부분은 OS/400 사용권 프로그램의 RW 구성요소로서 DRDA 지원사항 대부분이 포함되어 있습니다. 이 구성요소는 TRCJOB(작업 추적) 명령이 TRCTYPE(*ALL)이나 TRCTYPE(*DATA)으로 발행될 때 특정 유형의 진단 정보를 생성합니다. RW 추적점은 그림 28에 있는 유형입니다. RW 추적점은 탐색 인자로서 '>>' 스트링을 사용하여 찾기 작업을 실행하면 쉽게 찾을 수 있습니다. 자세한 정보는 RW 추적점 설명 주제를 참조하십시오. 각 추적점에 덤프된 자료의 끝을 '<<<...' 분리문자를 찾아서 판별할 수 있습니다. 자료의 끝에는 마지막 행을 채울 수 있는 하나 이상의 '<' 분리문자가 있습니다.

```
DATA FF 6E6ED9E6D8E840D9C37A0016D052000100102205000611490000    *>>RW0Q RC:   }           *
DATA FF 0006210224170025D0530001001F241A0C76D00500023100030A    *          }           }           *
DATA FF 00080971E0540001D000010671F0E00000002CD0530001002624    *    \   } 0\ }           *
DATA FF 1BFF000000100F1F1F1411000000000000FF0000000200F2F2    *      111           22 *
DATA FF F241200000000000000026D05200010020220B00061149000400    *2           }           *
DATA FF 162110C4C2F2C5E2E8E24040404040404040404040056D00300    * DB2ESYS           } *
DATA FF 01005024080000000064F0F2F0F0C4E2D5E7D9C6D54000C4C2    * &          02000DSNXRFN DB *
DATA FF F2C5E2E8E240404040404040404040FFFFF92000000000000    *2ESYS           k *
DATA FF 0000FFFFFFF0000000000000004040404040404040404000    *           *
DATA FF 0000004C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C4C    * <<<<<<<<<<<<<<<<<<<<< *

```

그림 28. 작업 추적 RW 구성요소 정보에 대한 예

다음 주제 예를 참조하십시오: 데이터 스트림을 구성하는 요소에 대한 RW 추적 데이터 정보 분석

주: 자료의 끝을 판별하기 위해 '<' 분리문자를 사용하는 예외적인 경우가 있습니다. 흔히 있는 일은 아니지만, 수신된 자료 스트림이 덤프 중인 상황에서는 추적 자료를 기록하는 모듈이 자료 스트림의 끝을 판별할 수 없습니다. 이 경우, 프로그램이 전체 수신 버퍼를 덤프하며 덤프된 자료의 길이가 자료 스트림의 길이 보다 길다고 경고하여 '<<<' 분리문자를 '(' 문자로 바꿉니다.

다음 '>>' 접두부는 7자로 된 스트링으로서 추적점을 식별합니다. 처음 두 글자 'RW'는 구성요소를 식별하며, 두 번째 두 글자는 실행 중인 RW 기능을 식별합니다. 'QY'는 DDM 명령 PNQRY, CNTQRY, CLSQRY에 해당하는 조회 기능을 나타내고 'EX'는 DDM 명령 EXCSQLSTT, EXCSQLIMM, PRPSQLSTT에 해당하는 EXECUTE 기능을 나타냅니다.

이 각각의 기능에 해당하는 프로그램 모듈은 작업 추적이 분산 SQL 액세스 작업의 어플리케이션 리퀘스터(AR) 끝에서 이루어졌는지 아니면 어플리케이션 서버(AS) 끝에서 이루어졌는지에 따라 달라집니다. 이 프로세스와 조회 기능을 실행하는 AR 모듈은 QRWSEXEC와 QRWSQRY이고 AS 모듈은 QRWTEXEC와 QRWTQRY입니다.

7바이트인 추적점의 마지막 두 글자 ID는 덤프된 자료의 성질이나 덤프가 실행된 지점을 나타냅니다. 예를 들면, SN은 AR이나 AS에서 송신된 자료 스트림에 해당되고 RC는 AR이 수신한 자료 스트림에 해당됩니다.

예: RW 추적 자료 분석

267 페이지의 그림 28에 있는 예는 분산 SQL 조회 기능이 실행되는 동안 수신된 자료 스트림입니다. 이 특정 추적은 연결의 어플리케이션 리퀘스터(AR) 끝에서 실행됩니다. 그러므로 자료를 생성하는 관련 프로그램 모듈은 QRWSQRY입니다.

다음 내용은 이 예에서 자료 스트림을 구성하고 있는 요소를 검사합니다. DRDA 자료 스트림 해석에 대한 자세한 정보는 *Distributed Relational Database Architecture Reference* 및 *Distributed Data Management Level 4.0 Architecture Reference* 서적을 참조하십시오. 이 문서는 웹 사이트 (<http://www.opengroup.org/dbiop/index.htm>)에서 구할 수 있습니다.

추적 자료는 추적점 ID의 끝을 표시해 주는 ':'의 뒤에 옵니다. 이 예에서, 자료 스트림의 처음 6바이트에는 DDM 자료 스트림 구조 헤더가 있습니다. 이 DSS 헤더의 처음 2바이트는 길이 필드이고 세 번째 바이트, X'D0'는 모든 DDM 자료에 대한 등록된 SNA 구조 ID입니다. 네 번째 바이트는 형식 ID입니다(자세한 정보는 뒤에 설명되어 있습니다). 5번째, 6번째 바이트에는 DDM 요구 상관 ID가 있습니다.

다음 2바이트, X'0010'(소수 16자리)에는 다음 번 DDM 오브젝트의 길이가 나타나 있으며, 이 경우, DDM 오브젝트는 이 오브젝트 뒤에 오면서 OPNQRYRM 응답 메시지에 대한 코드점인 X'2205'에 의해 식별됩니다.

다음 16바이트 응답 메시지는 응답 메시지 뒤에 오는 응답 오브젝트에 대한 6바이트 DSS 헤더입니다. 첫 번째 응답 오브젝트는 X'241A' 코드점으로 식별되며 QRYDSC 오브젝트입니다. 이 예에 나오는 두 번째 응답 오브젝트는 X'241B' 코드점으로 식별되는 QRYDTA 구조입니다. (추적 출력에서 두 행으로 분할됩니다.) OPNQRYRM 코드점의 경우, 앞의 2바이트에는 오브젝트의 길이가 나타나 있습니다.

- | QRYDTA 오브젝트에 대한 자세한 정보는 X'241B' 코드점 뒤에 오는 X'FF'를 참조하십시오. 이것은 널
- | SQLCAGRP를 의미합니다(전선을 따라 흐르는 SQLCA 또는 SQL 진단 영역 양식). SQLCAGRP의 널 양
- | 식은 관련 자료에 관한 오류나 경고 정보가 없음을 나타냅니다. 이 경우, 관련 자료는 SQL SELECT 작업으
- | 로 생긴 자료 행이며 널 SQLCAGRP 뒤에 옵니다. 그러나 자료 행과 SQLCAGRP가 모두 널일 수 있으므로
- | 널 SQLCAGRP 뒤에 오는 첫 번째 바이트는 자료 행이 널이 아님을 나타내는 X'00'이 있는 인디케이터입니

- 1 다. 널 인디케이터 바이트의 의미는 첫 번째 비트로 결정됩니다. 이 위치에 있는 '1'은 '널'을 의미합니다. 그러나 모든 8비트는 보통 인디케이터가 널 오브젝트를 나타낼 때 설정됩니다.

자료 행의 포맷은 선행 QRYDSC 오브젝트에 표시됩니다. 이 경우, QRYDSC는 널이 가능한 SMALLINT 값, CHAR(3) 값 그리고 널이 불가능한 배정밀도 부동 소수점 값이 행에 있음을 나타냅니다. 널 SQLCAGRP의 두 번째 바이트는 SMALLINT 필드와 관련있는 널 인디케이터로서, 이 필드가 널이 아니라는 것과 그 뒤에 오는 X'0001'이 필드 자료임을 표시합니다. 그 뒤에는 널이 가능한 CHAR(3)이 오며 '111'이 들어 있습니다. 그 뒤에 오는 부동 소수점 값은 널이 불가능하도록 정의되기 때문에 X'00'바이트가 올 수 없습니다.

널 SQLCAGRP로 된 두 번째 자료 행은 첫 번째 행 뒤에 오고, 그 뒤에는 다른 6바이트 DSS 헤더가 옵니다. 헤더에 있는 형식 바이트의 두 번째 반(X'2')은 해당 DSS가 REPLY임을 나타냅니다. 이전 DSS의 형식 바이트(X'53')는 이전 DSS가 OBJECT DSS이었음을 나타냅니다. 세 번째 DSS가 전달해 준 ENDQRYRM 응답 메시지는 REPLY DSS가 있어야 합니다. ENDQRYRM 코드점은 X'220B'입니다. 이 응답 메시지는 심각도 코드 X'0004'와 조회 자료('DB2ESYS')를 리턴하는 RDB의 이름이 들어 있습니다.

이 예에 나오는 다음 세 번째 DSS는 네 번째이자 마지막 응답 메시지입니다. 이에 대한 형식 바이트는 X'03'입니다. 3은 OBJECT DSS를 나타내며, 그 앞에 오는 0은 체인의 마지막 DSS를 나타냅니다(체인 비트는 꺼져 있습니다).

DSS에 있는 오브젝트는 SQLCAGRP로서 널이 아닌 SQLCAGRP가 들어 있습니다. X'2408' SQLCARD 코드점 뒤에 오는 첫 번째 바이트는 SQLCAGRP가 널이 아님을 알려 주는 인디케이터입니다. 다음 4바이트, X'00000064'는 +100 SQLCODE를 나타내는데 '행을 찾을 수 없음' 상황이 발생하여 조회 작업이 종료됨을 의미합니다. 나머지 필드는 SQLCA에 있는 다른 필드에 해당됩니다. 나머지 필드는 SQLCA 또는 SQL 진단 영역에 있는 다른 필드에 해당됩니다. SQLCAGRP 필드를 SQLCA 및 SQL 진단 영역 필드에 맵핑하는 방법은 *Distributed Relational Database Architecture Reference* 서적을 참조하십시오. 본 문서는 웹 사이트 (<http://www.opengroup.org/dbiop/index.htm>)에서 구할 수 있습니다.

RW 추적점 설명

RWff RC--수신 자료 스트림 추적점

이 자료 스트림에는 어플리케이션 서버(AS) 프로그램으로 생긴 DDM 응답이 있으며 DSS 헤더도 이 자료 스트림에 있습니다. 이는 추적점이며 267 페이지의 그림 28에서 설명됩니다.

실행될(ff) DRDA 기능의 ID는 다음과 같습니다.

- ff** DRDA 기능
- AC** RDB 액세스.
- OQ** 조회 열기.
- CQ** 조회 계속.
- EQ** 조회 닫기.
- PS** SQL문 준비.

- XS SQL문 실행.
- XI SQL문 즉시 실행.
- DT 표 명령문 설명.
- DS 명령문 설명.
- SY TCP/IP 동기화 위치 조작

RWff SN--송신 자료 스트림 추적점

전선을 통한 헤더와 전송의 추가에 대한 낮은 레벨의 CN 구성요소가 제공되기 전에 어플리케이션 리퀘스터 (AR) 프로그램의 DDM 요구나 어플리케이션 서버(AS) 프로그램의 DDM 응답이 존재하므로 이 자료 스트림에는 AR 프로그램의 DDM 요구나 AS 프로그램의 DDM 응답이 들어 있습니다. 내용 외에도, 수신 자료 스트림과 송신 자료 스트림의 추적 정보 사이의 주된 차이는 송신 자료 스트림의 추적 정보에 6바이트 DSS 헤더 정보가 없다는 점입니다. 송신 자료 스트림 추적 영역에 있는 첫 번째 DSS의 경우에는 헤더가 전체적으로 생략되어 있으며 그 다음 DSS의 경우는 6바이트가 0으로 되어 있어서 CN 구성요소 모듈이 차후에 이를 구성할 때 헤더가 덮어쓰게 됩니다.

실행 중인 DRDA 기능의 ID는 269 페이지의 『RWff RC--수신 자료 스트림 추적점』에 나열된 ID와 같습니다.

RWQY S1--부분 송신 자료 스트림 추적점 1

비슷한 환경에서 QRYDTA를 빌드하기 위해 새로운 조회 블록이 있어야 하는 경우, 이 추적점은 QRWTQRY 모듈의 NEWBLOCK 루틴에서 발생합니다. 비슷한 환경에서는 조회 블록을 전송 전에 채울 필요가 없으며, 항상 이 지점의 전선에 있어야 버퍼 공간을 재사용할 수 있습니다. 다른 송신 자료 스트림에서와 같이 DSS 헤더는 비어 있습니다.

RWQY S2--부분 송신 자료 스트림 추적점 2

비슷하지 않은 환경에서 QRYDTA를 빌드하기 위해 새로운 조회 블록이 있어야 하는 경우, 이 추적점은 QRWTQRY 모듈의 NEWBLOCK 루틴에서 발생합니다. 비슷하지 않은 환경에서는 마지막 하나만 제외하고 모든 조회 블록이 새 블록을 구성하기 전에 채워져야 하고 모든 조회 블록이 빌드되어야만 전송됩니다.

RWQY BP--성공적인 페치 추적점

SQFCHCRS 매크로에 대한 호출로 인해 BPCA 구조에 널이 아닌 포인터가 생기면 이 추적점은 하나 이상의 레코드가 BPCA 버퍼에 리턴될 것을 나타내는 QRWTQRY 모듈의 FETCH 루틴에서 발생합니다. 덤프된 자료는 BPCA 구조(관련 버퍼 없음)로서 다른 버퍼 사이에서 리턴된 레코드의 수를 나타냅니다.

RWQY NB--성공하지 못한 페치 추적점

SQFCHCRS 매크로에 대한 호출로 인해 BPCA 구조에 널 포인터가 생기면 이 추적점은 BPCA 버퍼에 리턴되는 레코드가 없음을 나타내는 QRWTQRY 모듈의 FETCH 루틴에서 발생합니다. 덤프된 자료는 SQLSTATE입니다.

| RWQY P0--결과 세트 의사 열기

| 연관된 정보는 패키지 리스트 항목입니다.

| RWQY AR--배열 결과 세트가 처리됨

| 연관된 정보는 배열 결과 세트 제어 블록입니다.

| RWQY DA--배열 결과 세트 SQLDA

| 연관된 정보는 배열 결과 세트 SQLDA입니다.

| RWQY DO--디버그 옵션

| 연관된 정보는 QRWOPTIONS 스트링의 수정 버전입니다.

아웃바운드 LOB 표 추적점에 저장된 RWQY L1 및 RWEX L1

이 추적점은 나중에 어플리케이션 리퀘스터로 전송하기 위해 QRWTQRY 또는 QRWTEEXEC에 의해 저장되는 큰 오브젝트(LOB) 열에 대한 주소 및 기타 정보를 기록합니다.

LOB 표 추적점의 자료 스트림에서 빌드되는 RWQY L2 및 RWEX L2

이 추적점은 QRWTQRY 또는 QRWTEEXEC에 의해 통신 버퍼로 복사되는 큰 오브젝트(LOB)에 대한 주소 및 기타 정보를 기록합니다.

인바운드 LOB 표 추적점에 저장된 RWQY L0 및 RWEX L0

이 추적점은 이후의 데이터베이스 관리 시스템(DBMS) 입력용 SQL 설명자 영역(SQLDA) 구성을 위해 QRWTQRY 또는 QRWTEEXEC에 의해 저장되는 큰 오브젝트(LOB) 열에 대한 주소 및 기타 정보를 기록합니다.

RWAC RQ--액세스 RDB 요구 추적점

이 추적점은 DRDA 어플리케이션 리퀘스터(AR)에 있는 QRWSARDB 모듈이나 어플리케이션 서버(AS)에 있는 QRWTARDB 모듈에 대한 항목에서 발생합니다. 따라서 내용은 모듈에 따라 다양합니다. 추적이 AS에서 발생하는 경우, 자료 내용은 QRWTARDB에 의해 실행되도록 DDM 명령을 식별해 주는 2바이트 DDM 코드입니다. QRWTARDB 뒤에는 단절시에는 SXXDSCT이고 클린업 시에는 SXXCLNUP이며, 연결 시에는 ACCRDB가 되는 명령의 영어 이름이 옵니다. 추적이 AR에서 발생하는 경우, 자료 내용은 다음과 같습니다.

주: 중요 범률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

OFFSET	TYPE	CONTENT
0	BIN(8)	FUNCTION CODE
1	CHAR(8)	INTERPRETATION OF FUNCTION CODE
9	BIT(8)	BIT FLAGS
10	CHAR(1)	COMMIT SCOPE
11	CHAR(1)	SQLHOLD value
12	CHAR(1)	CMTFAIL value
13	BIN(15)	Index of last AFT entry processed by RWRDBCMT

The function codes are:

0	'CONNECT '	==>	CONNECT
1	'DISCONN'	==>	DISCONNECT
2	'CLEANUP '	==>	CLEANUP
3	'RELEASE '	==>	RELEASE
4	'EXIT '	==>	EXIT
5	'PRECOMT '	==>	PRE-COMMIT
6	'POSTCMT '	==>	POST-COMMIT
7	'PREROLLB'	==>	PRE-ROLLBACK
8	'POSTROLL'	==>	POST-ROLLBACK
9	'FORCED D'	==>	FORCED DISCONNECT

RWAC cb--액세스 RDB 제어 블록 추적점

다음 추적점은 QRWSARDB 모듈이 제공하는 기능과 관련된 제어 블록을 식별해 줍니다.

cb 제어 블록의 이름.
LV 로컬 변수.
DD 화약 정의 디렉토리.
CD 화약 정의 제어 블록.
RI TSSCNAFT '리모트 정보' 구조.
CB 액세스 RDB 제어 블록.
DE RDB 디렉토리 항목.
TE 활동 파일 표 항목.

RWSY FN: SYNCxxx [TYPE:x] -- 소스 TCP SYNC/RESYNC 추적점

이 소스 측 추적점은 TCP/IP 이단계 화약 조작 실행에서 나온 다양한 명령과 응답을 기록합니다. 위의 'xxx' 로 표시되는 자료의 세그먼트는 다음과 같습니다.

- CTL, 제어 명령 표시
- RSY, 재동기화 명령 표시
- CRD, 제어 명령의 응답 자료 표시
- RRD, 재동기화 명령의 응답 자료 표시

CTL 및 RSY 레코드에 대해, 해당 명령과 연관된 TYPE 코드도 있습니다. 이는 인쇄 가능한 문자가 아니므로 레코드의 16진 자료 부분에서만 볼 수 있습니다. 이는 스트링 'TYPE:'을 따릅니다.

RWSY xx: yyyyyyy... -- 목표 TCP SYNC/RESYNC 추적점

이 목표 측 추적점은 다양한 정보를 기록합니다. 정보 유형은 위의 xx로 표시된 두 문자로 식별합니다. 세부사항은 가변 길이 yyyyyyy 스트링에 있습니다.

- RC 유형은 수신된 명령을 기록합니다: SYNCCTL 또는 SYNCRSY.
- RW 유형은 매개변수 구조 WrwSYData를 기록합니다.
- LG 유형은 수신된 synclog(여러 번 발생할 수 있음)를 기록합니다.
- SN 유형은 오류가 발생하지 않았을 때, 송신 버퍼를 기록합니다.

- GE 유형은 일반적인 예외의 경우 로컬 변수를 기록합니다.
- TE 유형은 TN 구성요소에 대한 요구가 실패했을 때, 송신 버퍼와 로컬 변수를 기록합니다(기록이 두 번 발생).
- CP 유형은 대화 프로토콜 오류가 감지되었을 때, 송신 버퍼와 로컬 변수를 기록합니다(기록이 두 번 발생 함).

RW_ff_m--어플리케이션 리퀘스터 드라이버(ARD) 제어 블록 추적점

이 추적점은 서로 다른 유형으로 된 ARD 호출의 ARD 제어 블록 내용을 표시합니다. 이 추적점이 표시하는 세 가지 제어 블록 유형은 입력 형식, 출력 형식, SQLCA입니다. 표시되는 호출 유형과 제어 블록 유형은 추적점 ID로 코드화됩니다. ID 형식은 RW_ff_m인데 ff는 호출 유형 ID, m은 제어 블록 유형 코드를 의미합니다. 호출 유형 ID(ff)와 제어 블록 유형 코드(m)는 다음과 같습니다.

주: 중요 범률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

ff	Call Type	m	Ctl Blk Type
--	-----	-	-----
CN	Connect	I	Input Format
DI	Disconnect	O	Output Format
BB	Begin bind	C	SQLCA
BS	Bind Statement		
EB	End bind		
PS	Prepare Statement		
PD	Prepare and Describe Statement		
XD	Execute Bound Statement with Data		
XB	Execute Bound Statement without Data		
XP	Execute Prepared Statement		
XI	Execute Immediate		
OC	Open Cursor		
FC	Fetch from Cursor		
CC	Close Cursor		
DS	Describe a Statement		
DT	Describe an Object		

첫 번째 실패 자료 캡처(FFDC)

iSeries 서버는 분산 관계형 데이터베이스(RDB)에 대한 오류 정보를 캡처하여 보고하는 방법을 제공합니다. 이 기능을 첫 번째 실패 자료 캡처(FFDC)라고 합니다. FFDC 기능을 지원하는 주 목적은 공식 프로그램 결합 수정 의뢰서(APAR)를 작성하는 OS/400 시스템의 DDM 구성요소에서 감지된 광범위한 오류 정보를 제공하는 것입니다.

또한 일부 시스템 관련 어플리케이션 문제를 진단하는 데에도 도움이 됩니다. 이 기능을 통해, 주요 구조와 DDM 자료 스트림은 스푼 파일에 자동 덤프됩니다. 첫 번째로 발생한 오류 정보가 자동으로 덤프되는 목표는 사용자는 실패 보고서를 작성하여 서비스 센터에 보고할 필요를 최소화하는 것입니다. FFDC는 어플리케이션 리퀘스터와 어플리케이션 서버 모두에서 사용됩니다.

유념해야 할 한 가지 사항은 모든 음수 SQLCODE가 덤프되는 것이 아니라 APAR 상태가 덤프인 경우만 SQLCODE가 덤프된다는 점입니다.

First-Failure Data Capture(FFDC) 및 덤프에 대한 자세한 정보는 다음 주제를 참조하십시오.

- FFDC 덤프
- FFDC 덤프 출력 설명
- DDM 코드
- 첫 번째 실패 자료 캡처(FFDC) 자료 찾기

FFDC 덤프

- 1 시스템 감지 내부 오류가 덤프할 FFDC 자료를 트리거합니다. QSFWERRLOG 시스템 값을 *NOLOG로 설정하게 되면 FFDC 출력이 불가능해질 수 있지만 FFDC 덤프 프로세스는 항상 작동 가능하게 하는 것이 좋습니다. FFDC 덤프가 발생하면, Qxxxxxxx에서 *소프트웨어 문제가 감지됨(여기서, Qxxxxxxx는 OS/400 모듈 ID임)이라는 정보용 메시지가 QSYSOPR 메시지 대기행렬에 기록됩니다.

FFDC 덤프 작업으로부터의 출력을 보려면 WRKSPLF(스플 파일에 대한 작업) 명령을 사용하여 QPSRVDMF를 보십시오. 덤프 출력에 들어 있는 내용은 다음과 같습니다.

- DDM 기능
- 실패한 DDM 모듈에 대한 특정 정보
- DDM 소스 또는 목표 기본 제어 블록
- DDM 내부 제어 구조
- DDM 통신 제어 블록
- 어플리케이션 리퀘스터에 있는 경우, 실패한 DDM 모듈에 대한 입출력 매개변수 리스트
- 요구 및 응답 자료 스트림

자료의 첫 1KB는 오류 기록부에 기록됩니다. 그러나 스플 파일에 기록된 자료는 항상 완벽해야 하고 작업하기 편해야 합니다. 여러 가지 DDM 대화가 빌드된 경우에는 각 스플 파일당 제한이 32개 항목이므로 덤프 출력에 스플 파일이 두 개 이상 있을 수 있습니다. 이 경우, 여러 개의 『소프트웨어 문제』 메시지가 앞에 별표(*)가 붙는 QSYSOPR 메시지 대기행렬에 있습니다.

주: 중요 법률 정보는 xv 페이지의 『코드 면책사항 정보』를 읽으십시오.

A 5738SS1 V2R1M1 AS/400 DUMP **B** 090454/SRR/SRRS1 02/27/91 15:12:52 PAGE 1
DUMP TAKEN FOR DETECTED ERROR

C
.SUSPECTED- QRWSQRY LIBRARY- S
..LICENSED PROGRAM- 5738SS1 V2R1M1
..FUNCTION- 5001
..LOAD- 0000
..PTF-

D
.DETECTOR- QRWSQRY LIBRARY- S
..LICENSED PROGRAM- 5738SS1 V2R1M1
..FUNCTION- 5001
..LOAD- 0000
..PTF-
.SYMPTOM STRING-

E 5738 MSGCPF3E86 F/QRWSQRY RC10000002
F
G

H
.SPACE- 01 **I**
000000 F0F17EC9 D5C4E740 F0F27EC6 C3E34E40 F0F37EC5 D4E2C740 F0F47ED7 D9D4E240 *01=INDX 02=FACT+ 03=EMSG 04=PRMS *
000020 F0F57EE2 D5C4C240 F0F67ED9 C3E5C240 F0F77EC1 D9C4C240 F0F87ED8 C4E3C140 *05=SNDB 06=RCVB 07=ARDB 08=QDTA *
000040 F0F97EC9 D5C4C140 F1F07EE2 D8C3C140 F1F17EE6 D9C3C140 F1F27ED9 C6D4E340 *09=INDA 10=SQCA 11=WRCA 12=RFMT *
000060 F1F37EC1 C6E34040 F1F47EE2 D4C3C240 F1F57EE3 E2D3D240 F1F67EE5 C1D9E240 *13=AFT 14=SMCB 15=TSLK 16=VAR5 *
000080 4DD9C5E2 E340C9E2 40C3C3C2 6BD7C3C2 E26BE2C1 E36BD7D4 C1D76BD9 C3E5C240 *(REST IS CCB,PCBS,SAT,PMAP,RCVB *
0000A0 D7C5D940 C3C3C25D *PER CCB) *
.SPACE- **J** 02
000000 200C1254 0102F5F8 F0F0F9 * 58009 *
.SPACE- 04
000000 D8D7C1D9 D4E20000 D67FC01D A60065A0 00000000 F0F10000 00000434 00000000 *QPARMS 0" 01 *
000020 D9C3C8C1 E2F2F6F6 40404040 40404040 4040E2D9 D9404040 40404040 40404040 *RCHAS266 SRR *
000040 40404040 D7E3F140 40404040 40404040 40404040 4040700F 70DB33C0 00BB0005 * PT1 *
.SPACE- 05
000000 00000000 0056D051 00010050 200C0044 2113D9C3 C8C1E2F2 F6F64040 40404040 * & RCHAS266 *
000020 40404040 E2D9D940 40404040 40404040 40404040 4040D7E3 F1404040 40404040 * SRR PT1 *
000040 40404040 40404040 700F70DB 33C000BB 00050008 21140000 7FFF0021 D0030001 * *
000060 001B2412 00100010 0676D004 00000671 E4D00001 0007147A 000002 * *
.SPACE- 06
000000 0016D052 00010010 22050006 11490000 00062102 24170052 D0530001 0022241A * *
000020 0F76D004 00002600 03020000 0A000009 71E05400 01D00001 0671F0E0 0000002A * *
000040 241BFF00 0001F0F0 F1000000 013FF000 00000000 00FF0000 02F0F0F2 00000002 * 001 0 002 *
000060 40000000 00000000 0010D052 0001000A 220B0006 11490004 0069D003 00010063 * *
0000E0 FF * *
.SPACE- **K** 07
000000 D9C3C8C1 E2F2F6F6 40404040 40404040 4040D9C3 C8C1E2F2 F6F64040 40404040 *RCHAS266 RCHAS266 *
000020 40404040 E2D9D940 40404040 40404040 40404040 4040D7E3 F1404040 40404040 * SRR PT1 *
000040 40404040 40404040 700F70DB 33C000BB D8E3C4E2 D8D3F4F0 F0D8E2D8 F0F2F0F1 * QTDSQL400QS0201*
000060 F1002500 00000000 25000000 000010F0 F4F5F1F7 F461E2D9 D961C4E2 F3F7F840 *1 045174/SRR/DS378 *
000080 40404040 40404040 40404040 40404040 40404040 40404040 40404040 * *
LINES 0000A0 TO 00015F SAME AS ABOVE
000160 40404040 40404040 40404040 4040A000 2434E2D9 D9404040 40404040 00000000 * SRR *
000180 C1D7D7D5 4BD9C3C8 C1E2F3F7 F8A7CCA7 54137200 40404000 00000000 00000000 *APPN.RCHAS378x x *
0001A0 00000000 00000000 * *

```

.SPACE-          09
000000 E2D8D3C4 C1404040 00000060 00010001 01F40002 00000400 00000040 40404040 *SQLDA          4          *
000020 80000000 00000000 007FC01E 11000334 00000000 00000000 00000000 00000000 *
000040 00080000 00250000 00000000 00000000 00000000 00000000 00000000 *
.SPACE-          10
000000 E2D8D3C3 C1404040 00000088 FFF8ABC 00041254 01020000 00000000 00000000 *SQLCA          *
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000060 00000000 00000000 00000000 00000000 00000000 00000000 40404040 *
000080 404040F5 F8F0F0F9          *          58009          *
.SPACE-          11
000000 E2D8D3C3 C1404048 00000088 00000000 00000000 00000000 00000000 *SQLCA          *
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000040 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000060 00000000 00000000 00000000 00000000 00000000 40404040 40404040 *
000080 404040F0 F0F0F0F0          *          00000          *
.SPACE-          13
000000 00001BB0 00310001 F0F0F0F0 F0F0F0F0 00000000 00000000 00000000 00000000 *          00000000          *
000020 00000470 000002C0 7023C382 57000048 80000000 00000000 007FA083 A3000820 *
000040 80000000 00000000 007FA083 E7000100 D9C3C8C1 E2F2F6F6 40404040 40404040 *
000060 40405CD3 D6C34040 40404040 5CD5C5E3 C1E3D940 D9C3C8C1 E2F2F6F6 5CD3D6C3 * *LOC          *NETATR RCHAS266*LOC*
LINES 0000A0 TO 001B9F SAME AS ABOVE
001BA0 00000000 00000000 00000000 00000000          *
.SPACE-          14
000000 E2D4C3C2 20000100 00000010 F0F9F0F4 F5F461E2 D9D961E2 D9D9E2F1 00000000 *SMCB          090454/SRR/SRRS1          *
000020 00000000 00000000 E5F0F2D9 F0F1D4F0 F1D9C3C8 C1E2F3F7 F8000000 00800000 *          V02R01M01RCHAS378          *
000040 0302C3D5 E2E2D5D9 C3E5D8D3 F7F9F7F1 80000000 00000000 007FA083 E9000106 *          CNSSNRCVQL7971          *
000060 F1000000 00710000 00000000 00000000 00000470 000002C0 7023C382 57000048 *1
.SPACE-          15
000000 00000000 00000000 007FA083 E60019FF 00000000 00000000 00000000 00000000 *
000020 00000000 00400000          *
.SPACE-          16
000000 00000000 00000000 00000000 00000002 00000017 000000E1 00000000 00000071 *
000020 00000000 00007FFF 00000003 00170000 001B0000 FF000000 00002410 00F0F060 *
000040 E70400          *X
.SPACE-          17
000000 E2C3C3C2 5CD3D6C3 40404040 40405CD5 C5E3C1E3 D9405CD3 D6C34040 4040D9C3 *SCCB*LOC          *NETATR *LOC          RC*
000020 C8C1E2F2 F6F65CD3 D6C34040 404007F6 C4C24040 40405CC4 D9C4C140 40404040 *HAS266*LOC          6DB          *DRDA          *
000040 40404040 40404040 4000001E 00110000 00000000 00000000 00000000 00000000 *
000060 00000000 00000000 00000000 00000000          *
.SPACE-          18
000000 E2D7C3C2 00000000 007FA083 A3000810 00000470 000002C0 7023C382 57000048 *SPCB          *
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000040 00000000 00000000 00000000 00000000          *
.SPACE-          19
000000 C5E7C3C2 00000076 00000003 00000079 00000009 00000082 00000010 00000092 *EXCB          *
000020 00000008 00000000 00000018 00200003 00030003 00030003 00030001 00030003 *
000040 00000000 00000000 00000000 00000000 00000000 0000C4C4 D4E5F0F2 D9F0F1D4 *
DMV02R01M*000060 F0F1F0F4 F5F1F7F4 61E2D9D9 61C4E2F3 F7F8D9C3 C8C1E2F2 F6F6 *
01045174/SRR/DS378RCHAS266          *
.SPACE-          20
000000 00000030 000002B6 00000430 0000043E 00010000 00000000 00000000 00000000 *
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000040 80000000 00000000 007FA083 D2000100 00000000 0000029A 0000005C 22050000 *
000060 00060000 02B60000 00B00000 00000000 00000000 00000000 00000000 *
LINES 0000E0 TO 00017F SAME AS ABOVE

```

```

.SPACE-                21
000000 0016D052 00010010 22050006 11490000 00062102 24170052 D0530001 0022241A *
000020 0F76D004 00002600 03020000 0A000009 71E05400 01D00001 0671F0E0 0000002A *
000040 2418FF00 0001F0F0 F1000000 013FF000 00000000 00FF0000 02F0F0F2 00000002 * 001 0 002 *
000060 40000000 00000000 0010D052 0001000A 220B0006 11490004 0069D003 00010063 *
000080 24080000 000064F0 F2F0F0F0 D8E2D8C6 C5E3C3C8 00D9C3C8 C1E2F2F6 F6404040 * 02000QSQFETCH RCHAS266 *
.SPACE-                22
000000 E2C3C3C2 5CD3D6C3 40404040 40405CD5 C5E3C1E3 D9405CD3 D6C34040 4040D9C3 *SCCB*LOC *NETATR *LOC RC*
000020 C8C1E2F2 F6F65CD3 D6C34040 404007F0 F0F14040 4040E77D F0F7C6F0 C6F0C6F1 *HAS266*LOC 001 X'07F0F0F1*
000040 7D404040 40404040 40000014 00110000 00000000 00000000 00000000 00000000 *
000060 00000000 00000000 00000000 00000000 00008F00 00000700 F0F0F100 00000000 * 001 *
.SPACE-                23
000000 C5E7C3C2 00000076 00000003 00000079 00000009 00000082 00000010 00000092 *EXCB b k*
000020 00000008 00000000 00000018 00200003 00030003 00030003 00030001 00030003 *
000040 00000000 00000000 00000000 00000000 00000000 0000C4C4 D4E5F0F2 D9F0F1D4 * DDMV02R01M*
000060 F0F1F0F4 F5F1F7F2 61E2D9D9 61C4E2F3 F7F8D9C3 C8C1E2F2 F6F6 *01045172/SRR/DS378RCHAS266 *
.SPACE-                24
000000 00000030 0000005C 00000000 000000CC 00010000 00000000 00000000 00000000 * *
000020 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000 *
000040 80000000 00000000 007FA083 A4000100 00000000 00000000 0000005C D2010000 * *K *
.SPACE-                25
000000 0010D002 0001000A D2010006 11490000 E2000D11 5AE5F0F2 D9F0F1D4 F0F1000C * V02R01M01 *
000020 116DD9C3 C8C1E2F2 F6F60014 115EF0F4 F5F1F7F2 61E2D9D9 61C4E2F3 F7F80064 * RCHAS266 045172/SRR/DS378 *
000040 14041403 00031423 00031405 00031406 00031407 00031444 00031458 00011457 *
000060 0003140C 00031419 0003141E 00031422 0003240F 000314A0 00041432 00031433 *
END OF DUMP
***** END OF LISTING *****

```

FFDC 덤프 출력 설명

다음은 앞의 그림에서와 같이 FFDC 덤프 출력에서 사용할 수 있는 자료 영역과 정보 유형에 대한 설명입니다.

주:

1. 각 FFDC 덤프 출력은 내용면에서는 서로 다르지만, 형식은 일반적으로 같습니다. 각 자료 섹션의 내용과 위치에 대한 사용자의 이해를 돕기 위해 색인(**I**)이 제공됩니다.
2. 자료 섹션은 『SPACE-』와 번호로 구분됩니다(예: SPACE- ... 01). 덤프 출력에 있는 자료 섹션은 실패 시의 작업과 진행 정도에 따라 달라집니다.
3. 각 자료 섹션에는 SQCA와 같은 이름이 주어집니다. SQCA는 DB2 UDB Query Manager 및 SQL Development Kit SQL 진단 영역으로부터의 자료에 대한 색션 이름입니다. SQL 진단 영역 자료를 찾으려면, 색인(**I**)에서 SQCA를 찾아 보십시오. 덤프 색인 샘플에서 SQCA는 자료 섹션 10(10=SQCA)으로 표시됩니다. SQL 진단 영역 자료를 보려면 SPACE- 10으로 가십시오.
4. 덤프될 수 있는 모듈의 기본 종류는 다음 두 가지입니다.
 - 어플리케이션 리퀘스터(AR) 모듈
 - 어플리케이션 서버(AS) 모듈

덤프 출력 샘플은 AR 모듈에서 사용하는 일반 덤프입니다. 보통 AR 덤프 출력에는 자료 섹션에 대한 고정된 번호가 있어서 색인에서 식별될 수 있습니다. (예를 들면, 덤프 출력 샘플에서는 SPACE- 01에서 16까지 나열되어 있습니다.) 또한 기타 자료 섹션의 변수 번호가 있으나, 이 기타 자료 섹션은 색인에 포함되지 않습니다. (예를 들면, 덤프 출력 샘플에서 SPACE- 17에서 25까지는 나열되어 있지 않습니다.)

어플리케이션 서버 덤프 출력은 자료 섹션의 고정 번호로만 구성되어 있고 이 모두가 색인에서 식별되기 때문에 간단합니다.

5. 자료 섹션이 현재 덤프 출력에 실제 있는지의 여부에 관계없이, 모든 자료 섹션에 대한 색인 항목이 있습니다. 예를 들면, 덤프 출력 샘플에는 SPACE- 08이 없지만, 색인에서 08은 QDTA(조회 자료)입니다. SPACE- 08이 없다는 것은 리턴될 조회 자료가 없으므로 아무것도 덤프할 수 없다는 의미입니다.
6. 덤프 출력 샘플에서, 색인의 마지막 항목은 『(REST IS CCB, PCBS, SAT, PMAP, RCVB, PER CCB)』입니다. 이 항목은 SPACE- 17 이상에는 통신 제어 블록(CCB)이 하나 이상 있음을 의미하며, 각 통신 제어 블록에 있는 내용은 다음과 같습니다.
 - 0, 한 개 또는 그 이상의 경로 제어 블록(SPCB); 보통 단 한 개 있음.
 - Exchange 서버 속성 제어 블록(EXCB)
 - 구문 분석기 맵 공간
 - 통신 제어 블록에 대한 수신 버퍼

자료 섹션 번호는 각 제어 블록이 덤프될 때 17에서부터 하나씩 커집니다. 예를 들면, 덤프 출력 샘플에서 SPACE-17에서 SPACE- 21까지의 자료 섹션은 덤프된 첫 번째 자료 제어 블록(CCB 1)이고, SPACE-22에서 SPACE- 25까지의 자료 섹션은 덤프된 두 번째 자료 제어 블록(CCB 2)을 나타냅니다.

17 CCB(아이캐처는 ‘SCCB:’입니다. 어플리케이션 서버 모듈에서 아이캐처는 ‘TCCB:’입니다.)

18 CCB 1에 대한 PCB(아이캐처는 ‘SPBC:’입니다.)

19 CCB 1에 대한 SAT(아이캐처는 ‘EXCB:’입니다.)

20 CCB 1에 대한 PMAP(아이캐처 없음.)

21 CCB 1에 대한 RCVB(아이캐처 없음.)

22 CCB 2(아이캐처는 ‘SCCB:’입니다.)

-- (대화를 사용할 수 없으므로 CCB 2에 대한 PCB가 없습니다.)

23 CCB 2에 대한 SAT(아이캐처는 ‘EXCB:’입니다.)

24 CCB 2에 대한 PMAP(아이캐처 없음.)

25 CCB 2에 대한 RCVB(아이캐처 없음.)

- A** 덤프가 발생한 서버의 이름과 릴리스 정보.
- B** 덤프 출력이 작성된 작업의 이름.
- C** 실패 우려가 있는 오퍼레이팅 시스템의 모듈 이름.
- D** 실패가 감지된 모듈의 이름.

징후 스트링-의 내용은 다음과 같습니다.

- E** 메시지 ID.
- F** FFDC 덤프를 초래할 우려가 있는 모듈의 이름.

G 실패 지점을 식별해주는 리턴 코드(RC).

RC 뒤에 있는 첫 번째 숫자는 이번 실패와 관련된 덤프 파일의 수입니다. 할당된 대화 수에 따라 여러 개의 덤프 파일이 있을 수 있습니다. 덤프 출력 샘플에서, 『1』이라는 숫자는 이번 실패와 관련된 첫 번째 덤프 파일 (단 하나일 수도 있음)을 나타냅니다.

오류 유형을 나타내는 리턴 코드의 제일 오른쪽에는 0이 아닌 4개의 숫자가 있습니다.

- AR이 감지한 오류 코드는 다음과 같습니다.

- 0001** 리모트 데이터베이스에 연결하는 동안에 발생한 실패.
- 0002** 있어서는 안되는 추가 수신 인디케이터.
- 0003** AR이 AS로부터 수신한 자료 스트림에서 인식할 수 없는 오브젝트를 감지.
- 0097** AR DDM 통신 관리자가 감지한 오류.
- 0098** AR의 DDM 구성요소가 감지한 대화 프로토콜 오류.
- 0099** 기능 검사.

- AS가 감지한 오류 코드는 다음과 같습니다.

- 0099** 기능 검사.
- 4415** 대화 프로토콜 오류.
- 4458** 에이전트 영구 오류.
- 4459** 자원 한계 도달.
- 4684** 자료 스트림 구문이 맞지 않음.
- 4688** 지원되지 않는 명령.
- 4689** 지원되지 않는 매개변수.
- 4690** 지원되지 않는 값.
- 4691** 지원되지 않는 오브젝트.
- 4692** 명령 검사.
- 8706** 조회가 열리지 않았습니다.
- 8708** 액세스할 수 없는 리모트 데이터베이스.
- 8711** 이전에 액세스했던 리모트 데이터베이스.
- 8713** 패키지 바인드 프로세스 활동 중.
- 8714** FDO:CA 설명자가 맞지 않음.
- 8717** 작업 단위의 이상 종료.
- 8718** 자료 및/또는 설명자가 일치하지 않음.
- 8719** 이전에 열었던 조회.

8722 조회 열기 실패

8730 사용할 수 없는 리모트 데이터베이스.

H 자료 섹션을 식별하는 SPACE- 번호. 이 번호는 색인별 자료 섹션 이름과 관계가 있습니다. 자료 섹션명은 다음 **I**에서 정의됩니다.

I 자료 섹션의 내용과 위치에 대한 사용자의 이해를 돕기 위한, SPACE- 번호(**H**에 정의되어 있음)의 색인 및 정의. 서로 다른 자료 섹션의 순서는 다른 모듈의 덤프 출력에 따라 다양합니다. 자료 섹션명의 의미는 다음과 같습니다.

- AFT: 모든 대화 정보가 들어 있는 DDM 활동 파일 표
- ARDB: AR과 AS 연결 정보가 들어 있는 액세스 리모트 데이터베이스 제어 블록
- ARDP: 사용자 공간 시작 부분에서의 ARD 프로그램 매개변수
- BDTA: 버퍼 처리 통신 영역(BPCA) 및 SELECT INTO 명령문의 관련 자료 레코드
- Bind: SQL 바인드 템플릿
- BPCA: BPCA 구조(자료 레코드 없음)
- DATA: BPCA와 관련된 자료 레코드. 이 섹션에 있는 레코드는 전체 BPCA 버퍼 내용에 영향을 주지 않습니다. 이미 처리된 레코드는 포함되지 않습니다.
- DOFF: 오류가 감지된 조회 자료 스트림(QRYDTA) 내의 오프셋
- EICB: 오류 정보 제어 블록
- EMSG: 기능 검사나 DDM 통신 관리자 오류와 관련된 오류 메시지
- FCT: DDM 기능 코드점(2바이트)
- FDOB: 실행 조작에 구문 분석기에 대한 FDO:CA 설명자 입력
- FDTA: FDO:CA 자료 구조는 다음 사항으로 구성됩니다.
 - FDO:CA 자료 스트림(FDODTA)의 길이를 정의하는 4바이트 필드
 - FDODTA
- HDRS: 통신 관리자 명령 헤더 스택
- IFMT: ARD 프로그램 입력 형식
- INDA: 작업의 삽입, 선택, 삭제, 갱신, 열기 및 실행을 위한 사용자 정의 SQLDA가 들어 있는 입력 SQLDA
- INDX: 자료 섹션명을 자료 섹션 SPACE- 코드에 맵하는 색인. 색인에 있는 모든 항목에 해당 자료 섹션이 있는 것은 아닙니다. 덤프 자료는 발생된 오류와 오류 시간에서의 진행 정도를 토대로 합니다. 최대 32개 항목까지 하나의 스푼 파일에 덤프될 수 있습니다.
- INST: SQL문.
- ITKN: 인터럽트 토큰.
- OFMT: ARD 프로그램 출력 형식.
- PKGN: 입력 패키지명, 일관성 토큰 및 섹션 번호.

- PMAP: AS 덤프 출력에 있는 구문 분석기 맵.
- PRMS: DDM 모듈 입출력 매개변수 구조.
- PSOP: 입력 구문 분석기 옵션.
- QDTA: 조회 자료 구조는 다음 사항으로 구성됩니다.
 - 조회 자료 스트림(QRYDTA)의 길이를 정의하는 4바이트 필드.
 - QRYDTA
- RCVB: 수신된 자료 스트림. 내용은 다음에 따라 달라집니다.
 - 덤프가 어플리케이션 서버에서 발생하는 경우, 섹션에는 어플리케이션 리퀘스터로부터 수신된 DDM 요구 자료가 들어 있습니다.
 - 덤프가 어플리케이션 리퀘스터에서 발생하는 경우, 섹션에는 어플리케이션 서버로부터 수신된 DDM 응답 자료가 들어 있습니다. 이 섹션이 없는 경우, 수신된 자료를 덤프의 변수 부분에 있는 수신 버퍼에서 찾을 수 있습니다.
- RDBD: 관계형 데이터베이스 디렉토리.
- RFMT: 레코드 형식 구조.
- RMTI: 약속 제어 블록에 있는 리모트 위치 정보.
- RTDA: ARD 프로그램으로부터 리턴된 SQLDA.
- SMCB: 다른 DDM 연결 제어 블록과 내부 DDM 제어 블록에 대한 포인터가 있는 DDM 소스 마스터 제어 블록.
- SNDB: 송신 자료 스트림. 내용은 다음에 따라 달라집니다.
 - 덤프가 어플리케이션 리퀘스터에서 발생하는 경우, 버퍼에는 어플리케이션 서버로 송신되었거나 송신 준비 중인 DDM 요구가 들어 있습니다.

이 예에서는 SPACE- 05의 시작 부분에 있는 4바이트로 된 0에 주목하십시오. 0이 있는 경우, 이는 자료 스트림의 일부분이 아니라 DDM 대형 오브젝트가 DDM 요구로 송신되어야 하는 경우에만 사용되는 버퍼 공간을 나타냅니다. 이 경우, DDM 요구 스트림은 왼쪽 4바이트로 바뀌게 됩니다.

 - 덤프가 어플리케이션 서버에서 발생하는 경우, 버퍼에는 어플리케이션 리퀘스터로 송신 준비 중인 DDM 응답 자료가 들어 있습니다.
- SQCA: 사용자에게 리턴될 출력 SQL 진단 영역
- SQDA: FDO:CA 구문 분석기로 빌드된 SQLDA.
- TBNM: 입력 리모트 데이터베이스 표 이름.
- TMCB: 목표 기본 제어 블록.
- TSLK: DDM 활동 파일 표와 다른 내부 DDM 제어 블록에 대한 포인터가 들어 있는 목표 또는 소스 연결 제어 블록.
- VARS: 덤프될 모듈의 로컬 변수.
- WRCA: 열기 작업(OPNQRYM)의 경우에만 리턴되는 경고 SQLCA.

- XSAT: 교환 서버 속성 제어 블록.
- 기타: 오류 시 작업의 경우 모든 DDM 대화에 대한 여러 개의 대화 제어 블록. 대화 제어 블록의 내용은 다음과 같습니다.
 - 설정된 대화에 대한 정보가 들어 있는 경로 제어 블록. 한 개의 대화 제어 블록에 여러 개의 경로 제어 블록이 있을 수 있습니다.
 - 어플리케이션 리퀘스터와 어플리케이션 서버 정보가 들어 있는 한 개의 교환 서버 정보 제어 블록.
 - 모든 DDM 명령, 오브젝트 및 응답에 대한 위치와 값이 들어 있는 DDM 구문 분석기 맵 영역.
 - 어플리케이션 서버가 수신하는 요구된 자료 스트림에 들어 있는 한 개의 수신 버퍼. 278 페이지의 6도 참조하십시오.

자료 섹션 번호는 각 제어 블록이 덤프될 때 하나씩 커집니다.

J 아이캐처 영역. 덤프되는 일부 영역의 자료 유형을 식별하는 정보.

K 액세스 RDB 제어 블록에서 실패가 발견될 때 진행 중인 대화에 대한 논리적 작업 단위 ID(LUWID). 이 자료 영역은 FFDC 색인에서 'ARDB' 스트링으로 식별할 수 있습니다. 이 예에서 자료 영역은 SPACE- 07입니다. LUWID는 오프셋 180에서 시작됩니다. 네트워크 ID(NETID)는 APPC입니다. 마침표로 네트워크 ID와 그 뒤에 오는 논리 장치(LU) 이름 RCHAS378을 구분합니다. 다음 LU 이름은 6바이트로 된 LUW 인스턴스 번호 X'A7CCA7541372'입니다.

DDM 코드

오류 코드는 DDM 오류 조건을 식별하는 샘플(덤프 출력)에 나오는 FFDC 덤프(**L**)에 포함됩니다. 이 조건은 DDM 구조로 정의할 수도 있고 정의하지 않을 수도 있습니다.

명령 검사 코드

FCT+(SPACE- 02)에 3, 4바이트인 1254가 있으면 6바이트로 된 다음 코드 중 하나를 찾아 보십시오.

- 01 관계형 데이터베이스(RDB)에 연결 실패.
- 02 DDM 자료 스트림의 상태가 부정확.
- 03 자료 스트림에 있는 인식할 수 없는 오브젝트.
- 04 SQL로부터 수신된 명령문 CCSID를 인식할 수 없음.
- 05 EXCSQLSTT OUTEXP 값이 실행될 SQL문과 일치하지 않음.
- 06 어플리케이션 서버(AS)로 송신하는 DDM 명령이나 오브젝트가 DRDA2 구조에 대한 OS/400 확장자를 위반.
- 07 AS로부터 수신된 DDM 응답이나 오브젝트가 DRDA2 구조를 위반.
- 08 SQLDA 자료 포인터가 사용해서는 안되는 NULL임.
- 09 제품 자료 구조가 맞지 않음.
- 0A XLATECC 실패.

- 0B EXTJOBDI 실패.
- 0C 이름 실패로부터 ASP 가져오기.
- 0D ASP 실패로부터 RDB 이름 가져오기.
- | 0E 예기치 않은 오류 자료.
- | 0F DDM/DRDA 요구를 인식할 수 없음.
- 10 예상 LOB가 수신되지 않음.
- 11 변수와 수신된 자료 사이의 LOB 길이가 맞지 않음.
- 12 LOB 사용법이 맞지 않음.
- 13 LOB에 맞지 않는 XMIT 모드
- 14 버퍼 확장 실패.
- 15 열기가 성공한 후 폐치된 음의 SQLCODE.
- 16 공간 할당 오류.
- 17 결과 세트 응답(SQRY)의 불일치.
- 18 결과 세트 응답(SQRY)의 예상치 못한 RM.
- | 19 응답을 빌드하는 중 오류 발생.
- 1A SQ 구성요소가 SQL 코드 -30020을 리턴함.
- | 1B SQL 진단 영역을 갱신하는 중 오류 발생.
- | 1C 응답을 빌드하는 중 오류 발생.
- 88 BPCA에 레코드가 없음.
- 89 예상치 못한 BGNBND 오브젝트.
- 8A 지원되지 않는 대형 DDM 오브젝트 헤더 크기.
- 8B LOB 표 오류.
- 8C LOB 요구 및 아무것도 사용할 수 없음.
- | 8D SET_LELAST 오류 1.
- | 8E SET_LELAST 오류 2.
- | 8F 0이 아닌 예상치 못한 QRYINSID.
- | 90 0이 아닌 QRYINSID.
- | 91 P 열기 시 OPNQFL.
- | 92 정상 열기 시 OPNQFL.
- 97 오류를 감지한 DDM 통신 관리자.
- 98 DDM 모듈이 감지한 대화 프로토콜 오류.

99 기능 검사. SPACE- 03에 있는 EMSG 섹션을 찾으십시오.

FF SQ 열기(TQRY)의 오류.

대화 프로토콜 오류 코드 설명

FCT+(SPACE- 02)에 3, 4바이트인 1245가 있으면 6바이트로 된 다음 코드 중 하나를 찾아 보십시오.

01 목표 통신 관리자가 수신한 RPYDSS.

02 체인없는 여러 DSS 송신 또는 여러 DSS 체인 송신.

03 허용되지 않는 OBJDSS 송신.

04 RQSDSS의 요구 상관 ID가 체인의 이전 RQSDSS 요구 상관 ID보다 작거나 같음.

두 RQSDSS에 똑같은 요구 상관 ID가 있는 경우, PRECCNVRM는 요구 상관 ID -1과 함께 RPYDSS에서 송신되어야 합니다.

05 OBJDSS의 요구 상관 ID가 이전 RQSDSS 요구 상관 ID와 같지 않음.

06 EXCSAT가 연결이 설정된 후의 첫 번째 명령이 아님.

DA SQLDA가 레이블을 조정하는 데 배가 되지 않음.

DF FDODSC가 수신되었지만 FDODTA를 수반하지 않음.

E0 OPNQRY(조회 열기) 응답 메시지가 없음.

E1 ENDQRYRM의 RDBNAM(조회 종료 응답 메시지)이 유효하지 않음.

E2 QRYDSC(조회 응답 세트 설명) 없이 OPEN이 QRYDTA(조회 응답 세트 자료)를 받음.

E3 예상치 못한 OPNQRY 응답 오브젝트.

E4 예상치 못한 CXXQRY 응답 오브젝트.

E5 하나의 OPEN 행에 있는 QRYDTA.

E6 OPNQRYRM 뒤의 RM이 유효하지 않음.

E7 인터럽트 응답 메시지 없음.

E8 어플리케이션 서버(AS)가 지원하지 않는 LOB 요구

I E9 표준 버전이 예상될 때 Light SQLDA를 수신함.

FD 오류 RM 뒤에 오는 널 SQLCARD(SQLCA 응답 자료).

FE 널 QRYDTA 행이 널 SQLCA 뒤에 있음.

FF 예상했던 SQLCARD가 빠짐.

DDM 구문 오류 코드 설명

FCT+(SPACE- 02)에 3, 4바이트인 124C가 있으면 6바이트로 된 다음 코드 중 하나를 찾아 보십시오.

01 6보다 작은 DSS 헤더 길이.

02 DSS 헤더 길이가 발견된 자료의 바이트 수와 일치하지 않음.

- 03 X'D0'가 아닌 DSS 헤드 C 바이트.
- 04 인식하지 못하거나 지원되지 않는 DSS 헤더 F 바이트.
- 05 DSS 연속이 지정되었지만 발견되지 않음. 예를 들면, DSS 연속이 마지막 DSS에 지정되고 SEND 인디케이터가 SNA LU 6.2 통신 프로그램에 의해 리턴되는 경우.
- 06 DSS 체인이 지정되었지만 발견된 DSS가 없음. 예를 들면, DSS 체인이 마지막 DSS에 지정되고 SEND 인디케이터가 SNA LU 6.2 통신 프로그램에 의해 리턴되는 경우.
- 07 4보다 작은 오브젝트 길이. 예를 들면, 명령 매개변수 길이는 2로, 명령 길이는 3으로 지정된 경우.
- 08 오브젝트 길이가 발견된 자료의 바이트 수와 일치하지 않음. 예를 들면, 길이가 150인 RQSDSS에 길이가 125인 명령어가 있거나 SRVDGN(서버 진단 정보) 매개변수의 길이가 200으로 지정되어 있지만 DDS에 50바이트만 남아 있는 경우.
- 09 허용된 최대값을 초과한 오브젝트 길이. 예를 들면, RECCNT 매개변수의 길이가 5로 지정되어 있지만, 전체 시간 필드 대신 시간 필드의 절반만 있는 경우.
- 0A 필요한 최소값 이하의 오브젝트 길이. 예를 들면, SVRCOD 매개변수의 길이가 5로 지정되었지만, 매개변수 길이가 6으로 고정된 경우.
- 0B 허용되지 않는 오브젝트 길이. 예를 들면, FILEXPDT 매개변수의 길이가 11로 지정되어 있지만, 전체 시간 필드 대신 시간 필드 절반만 있는 경우.
- 0C 잘못된 대형 오브젝트 확장 길이 필드(DSS 설명 참조). 예를 들면, 확장 길이 필드가 있지만 길이가 단지 3바이트인 경우. 확장 길이 필드는 2바이트의 배수가 되어야 합니다.
- 0D 오브젝트 코드점 색인이 지원되지 않음. 예를 들면 코드점 X'8032'가 있지만 X'8'은 예약된 코드점 색인인 경우.
- 0E 필요한 오브젝트를 찾을 수 없음. 예를 들면, CLRFIL 명령에 FILNAM 매개변수가 없거나, MODREC 명령 뒤에 RECORD 명령 자료 오브젝트가 오지 않는 경우.
- 0F 송신된 명령 자료 오브젝트가 너무 많음. 예를 들면, MODREC 명령 뒤에 두 개의 RECORD 명령 자료 오브젝트가 오거나, DELREC 명령 뒤에 RECORD 오브젝트가 오는 경우.
- 10 상호 배타적인 오브젝트가 있음. 예를 들면, CRTDIRF 명령에 DCLNAM과 FILNAM 매개변수가 모두 지정되어 있는 경우.
- 11 송신된 명령 자료 오브젝트가 너무 적음. 예를 들면, RECCNT(5)를 지정하는 INSRECEF 명령 뒤에 RECORD 명령 자료 오브젝트가 네 개만 오는 경우.
- 12 중복 오브젝트가 있음. 예를 들면, LSTFAT 명령에 두 개의 FILNAM 매개변수가 지정된 경우.
- 13 지정된 요구 상관 ID가 유효하지 않음. 이 오류 코드 대신 X'04' 또는 X'05' PRCCNVCD를 PRCCNVRM와 함께 사용하십시오. 이 오류 코드는 레벨 1 구조와 호환되도록 유지관리됩니다.
- 14 필요한 값을 찾을 수 없음.
- 15 예약된 값이 허용되지 않음. 예를 들면, INSRECEF 명령에 RECCNT(0) 매개변수가 지정된 경우.
- 16 2보다 작거나 같은 DSS 연속. 예를 들면, DDS 연속 바이트 길이가 1인 경우.

- 17 필요한 순서가 아닌 오브젝트. 예를 들면, RECAL 오브젝트에 지정된 순서가 아닌 RECNR 오브젝트 앞에 오는 RECORD 오브젝트가 있는 경우.
- 18 바이너리 1이 아니라 바이너리 1로 지정된 DSSFMT 비트 3인 DSS 체인 비트가 요구됨.
- 19 현재 DSS를 나타내는 이전 DSS에 똑같은 요구 상관이 있지만 요구 상관 ID는 같지 않음.
- 1A 바이너리 1이 아니라 오류 연속인 DSS 체인 비트가 요구됨.
- 1B 상호 배타적인 매개변수 값이 지정됨. 예를 들면, OPEN 명령에 PRPSHD(TRUE)와 FILSHR(READER)이 지정되어 있는 경우.
- 1D 코드점이 유효하지 않은 명령임. 예를 들면, RQSDSS에 있는 첫 번째 코드점이 디렉토리에 있지 않거나 명령에 대한 코드점이 아닌 경우.

부록 D. 용어집

고성능 라우팅(HPR)

Advanced Peer-to-Peer Networking(APPN) 구조에 추가된 것으로 특히 고속 링크를 사용할 때 데이터 라우팅 성능과 신뢰도를 향상.

관계형 데이터베이스(Relational Database)

표의 집합으로 인식되고 관계형 자료 모델에 따라 조작되는 데이터베이스입니다. 사용자가 iSeries 서버에서 액세스할 수 있는 관계형 데이터베이스는 세 가지 유형이 있습니다. 시스템 관계형 데이터베이스(또는 시스템 데이터베이스), 사용자 관계형 데이터베이스(또는 사용자 데이터베이스) 및 리모트 관계형 데이터베이스(또는 리모트 데이터베이스)가 그것입니다.

구조화 조회 언어(SQL: Structured Query Language)

호스트 프로그래밍 언어 내에서 사용되는 언어 또는 대화식으로 정보를 데이터베이스에 넣고 선택된 정보를 데이터베이스에서 가져와 구성하는 작업을 하는 언어입니다. SQL 또한 데이터베이스 자원에 대한 액세스를 제어할 수 있습니다. SQL은 다른 서버에서 분산 자료 처리를 가능하게 하는 데 필요한 일관성을 제공합니다.

논리 장치(LU: Logical unit)

SNA에서 세 가지 유형의 네트워크 주소지정 가능 단위 중 하나로 사용자가 통신 네트워크에 액세스하는 데 필요한 수단인 포트 역할을 합니다. 다른 두 가지는 물리 장치(PU)와 시스템 서비스 제어점(SSCP)입니다.

대화식 구조화 조회 언어(ISQL: Interactive Structured Query Language)

DB2 UDB 조회 관리자 및 SQL 개발 킷 사용권 프로그램에 있는 SQL문이 일괄처리 모드 대신 동적으로 실행될 수 있도록 하는 기능. 모든 대화식 SQL문은 워크스테이션에서 읽혀지고 준비되고 동적으로 실행됩니다.

대화식 처리(Interactive processing)

각 오퍼레이터의 조치가 프로그램이나 시스템으로부터 응답을 발생시키는 처리 방법. 일괄처리와 대조를 이룹니다.

독립형 보조 기억장치 풀 또는 독립형 디스크 풀(Independent auxiliary storage pool, or independent disk pool)

디스크 장치 또는 주소지정 가능한 디스크 기억장치를 구성하는 디스크 장치 서브시스템으로부터 정의되는 하나 또는 그 이상의 저장 장치를 말합니다. 독립형 디스크 풀은 오브젝트, 오브젝트를 포함하는 디렉토리 및 인증 소유권 속성과 같은 다른 오브젝트 속성을 포함합니다. 독립형 디스크 풀은 시스템을 재시동하지 않고 사용할 수 있게(연결변환됨) 또는 사용할 수 없게(단절변환됨) 할 수 있습니다. 독립형 디스크 풀은 a) 클러스터링 환경에서 다중 시스템간에 연결변환이 가능하거나 또는 b) 단일 시스템에 개별적으로 연결될 수 있습니다.

동기 자료 링크 제어(SDLC)

(1) 통신 회선 상의 자료 전송을 제어하는 명령을 사용하는 통신 회선 제어의 한 형태 (2) 통신 회선에서

동기화되고 코드가 투명하여(code-transparent) 비트 단위의 직렬(serial-by-bit) 정보를 전송하기 위해 미국 표준 협회(ANSI)의 확장된 자료 통신 제어 프로시듀어(ADCCP)와 국제표준화 기구(ISO)의 고급 자료 링크 제어(HDLC)의 서브세트를 따르는 통신 통제. 전송 교환은 교환 또는 비교환 회선 상에 양방향 전송 또는 반이중 전송이 될 수 있습니다. 연결 구성은 지점 간, 분기 또는 루프가 있습니다.

롤백(Roll back)

작업 단위 지원으로 어플리케이션 프로그램은 작업 단위로 변경사항을 롤백할 수 있습니다. 작업 단위가 롤백되면 마지막 예약이나 롤백 연산이 적용되지 않았기 때문에 변경이 이루어집니다. 따라서 어플리케이션 프로그램은 데이터베이스에 대한 요구 집합을 단위로 처리합니다.

리모트 관계형 데이터베이스 또는 리모트 데이터베이스

리모트로 액세스할 수 있는 iSeries 서버 또는 다른 서버에 상주하는 데이터베이스

리모트 작업 단위(RUW)

리모트 작업 단위(RUW)는 어플리케이션 프로그램이 작업 단위 내의 리모트 데이터베이스에 있는 자료에 액세스할 수 있는 분산 관계형 데이터베이스 처리의 한 모드입니다. 리모트 작업 단위에 하나 이상의 관계형 데이터베이스 요구가 있지만 모든 요구는 동일한 리모트 데이터베이스에 대한 것이어야 합니다.

문자 자료 표시 구조(CDRA: Character Data Representation Architecture)

식별자, 서비스, 지원하는 자원 및 iSeries 서버간 CDRA를 지원하는 다른 유형의 서버간에 일관성 있는 표시, 처리, 문자(자료) 교환을 하기 위한 규약을 정의하는 IBM 구조입니다.

물리 장치(PU: Physical unit)

SNA에서 세 가지 유형의 네트워크 주소지정 가능 단위 중 하나 물리 장치는 SNA 네트워크의 각 노드에 존재하며, 시스템 서비스 제어점 논리 장치(SSCP-LU) 세션에서 요구하는 대로 노드의 자원(접속 링크와 인접 링크 스테이션과 같은)을 관리하고 모니터링합니다.

바인딩(Binding)

(1) 통합된 언어 환경®(ILE: Integrated Language Environment) 모듈을 패키지와하고 이 모듈들 간에 전달되는 기호들을 분석함으로써 프로그램을 작성하는 프로세스입니다. 예를 들어, 어플리케이션 프로그램을 실행하기 전에 프로그램, 참조하는 표 및 뷰 사이의 관계를 설정해야 합니다. (2) DRDA의 문맥에서, 어플리케이션 서버(AL)에서 SQL 패키지를 작성하는 프로세스입니다.

보안 소켓층(SSL: Secure Sockets Layer)

Netscape Communications Corp.와 RSA Data Security, Inc.가 개발한 널리 사용되는 보안 체계입니다. SSL은 클라이언트가 서버를 인증할 수 있도록 하고 모든 자료와 요구를 암호화합니다. SSL의 보호를 받는 보안 서버의 URL은 http가 아니라 https로 시작합니다.

보조 기억장치 풀(ASP: Auxiliary storage pool)

기억장치 또는 보조 기억장치를 구성하는 기억장치 서브시스템으로부터 정의되는 하나 또는 그 이상의 저장 단위를 말합니다. ASP는 기억장치 실패의 효과를 제한하고 회복 시간을 줄일 수 있도록 자료를 구성하는 방법을 제공합니다.

분산 관계형 데이터베이스(DRDB)

분산 관계형 데이터베이스는 자료와 자료 자체를 사용하는 어플리케이션 프로그램이 다른 시스템에 위치

할 때 또는 프로그램이 한 서버의 다중 데이터베이스에 있는 자료를 사용할 때 존재합니다. 후자의 경우, DRDA 프로토콜이 하나의 서버에서 하나 또는 그 이상의 데이터베이스를 액세스하는데 사용된다는 점에서 데이터베이스가 분산되어 있는 것입니다.

분산 자료 관리(DDM)

한 시스템의 어플리케이션 프로그램이나 사용자가 리모트 시스템에 저장된 데이터베이스 파일을 사용할 수 있도록 해주는 오퍼레이팅 시스템의 기능. 시스템은 통신 네트워크에 연결되어 있어야 하며 리모트 시스템은 DDM을 사용해야 합니다. 이 용어는 기초 통신 구조에도 적용됩니다.

분산 작업 단위(DUW)

분산 작업 단위(DUW)를 사용하면 사용자나 어플리케이션 프로그램이 작업 단위 내의 여러 위치에서 자료를 읽거나 갱신할 수 있습니다.

분산관계형 데이터베이스 구조 (DRDA)

DRDA는 다음 작업을 위해 형식 및 프로토콜 세트를 지정합니다.

- 어플리케이션에서 하나 또는 그 이상의 리모트 데이터베이스 관리 시스템으로의 연결
- 리모트 데이터베이스 관리 시스템 간의 자료 교환
- 무결성 및 일관성을 보장하기 위한 트랜잭션 관리

DRDA 프로토콜은 분산 자료 관리 구조 상에서 빌드됩니다.

사용자 관계형 데이터베이스 또는 사용자 데이터베이스(User Relational Database, or User Database)

하나의 독립형 보조 기억장치 풀 그룹에 존재하는 모든 데이터베이스 오브젝트와 독립형 보조 기억장치 풀에 저장되어 있지 않은 데이터베이스 오브젝트. 주: V5R2에서 iSeries 서버는 독립형 보조 기억장치 풀이 서버에 구성되어 있을 경우, 다중 관계형 데이터베이스의 호스트가 됩니다. 시스템 관계형 데이터베이스는 항상 하나가 존재하고 사용자 관계형 데이터베이스는 하나 이상 존재할 수 있습니다. 각 사용자 데이터베이스는 시스템 데이터베이스의 모든 오브젝트를 포함합니다. 주: 그러나 사용자는 SQL에서는 시스템 데이터베이스가 사용자 데이터베이스 내에 포함되는 것으로 볼 수 있는 반면, 확약 제어에서는 시스템 데이터베이스가 별개의 데이터베이스로 다루어진다는 것을 명심해야 합니다.

서브시스템 설명(Subsystem description)

시스템이 제어하는 운영 환경의 특성을 정의하는 정보를 담고 있는 시스템 오브젝트. 이 오브젝트 유형의 시스템 인식 식별자는 *SBSD입니다.

서브시스템(Subsystem)

시스템이 처리와 자원을 조정하는 운영 환경으로 서브시스템 설명에 의해 정의되어 있습니다.

스키마(Schema)

라이브러리, 저널, 저널 리시버, SQL 카탈로그 및 선택형 자료 사전으로 구성됩니다. 스키마는 관련 오브젝트를 그룹화하며 이름으로 오브젝트를 찾을 수 있습니다. 주: 또 스키마를 컬렉션으로서 공통으로 참조할 수 있습니다.

스풀(Spool)

(1) 파일 또는 작업을 나중에 처리 또는 인쇄하기 위해 디스크 기억장치에 넣는 시스템 기능입니다. (2) 주변 장치와 컴퓨터 프로세서 간에 자료를 전송할 때, 보조 기억장치를 버퍼 기억장치로 사용함으로써 처리 지연을 줄이는 것을 말합니다.

시스템 관계형 데이터베이스 또는 시스템 데이터베이스(System Relational Database or System Database)

iSeries 서버에 접속된 디스크 상에 존재하는 모든 데이터베이스 오브젝트로 독립형 보조 기억장치 풀에 저장되어 있지 않은 오브젝트.

시스템 네트워크 구조(SNA: Systems Network Architecture)

IBM 네트워크에서, 네트워크의 구성과 조작 제어뿐 아니라, 네트워크 상에 정보 단위를 전송하는데 사용되는 계층적 논리 구조, 포맷, 프로토콜 및 조작 순서의 설명.

시스템 서비스 제어점(SSCP: System services control point)

다른 시스템과 장치를 관리하고 네트워크 오퍼레이터의 요구와 문제점 분석 요구를 조정하고 네트워크 사용자에게 디렉토리 라우팅과 기타 세션 서비스를 제공하기 위한 SNA 네트워크 내에 있는 중재점.

암호화(Encryption)

컴퓨터 보안에서 원래 자료를 얻을 수 없거나 오직 암호 해독 프로세스를 사용해서 원래 자료를 얻을 수 있도록 자료를 알 수 없는 양식으로 변형시키는 프로세스.

액세스 계획(Access plan)

iSeries용 DB2 UDB에서 프로그램 실행 중에 나타나는 SQL문을 처리하는데 사용되는 컴파일 시간에 생성된 제어구조입니다.

어플리케이션 리퀘스터 드라이버(ARD: Application requester driver)

OS/400의 SQL Client Integration 기능에서 사용되는 종료 프로그램입니다. 이는 SQL 어플리케이션이 OS/400 관계형 데이터베이스 이외의 데이터베이스 관리 시스템이 관리하는 자료에 액세스할 수 있도록 해 줍니다.

어플리케이션 리퀘스터(AR)

분산 관계형 데이터베이스를 사용할 때, 어플리케이션 프로그램이 실행되는 시스템을 말합니다.

어플리케이션 서버(AS: Application server)

분산 관계형 데이터베이스를 사용할 때, 리모트 자료가 상주하는 시스템을 가리킵니다.

위치(Location)

위치는 분산 관계형 데이터베이스에 참여하고 있는 관계형 데이터베이스 관리 시스템의 상호 연결된 네트워크에 있는 특정 관계형 데이터베이스 관리 시스템입니다. 여기에서 '위치'는 독립형 ASP 그룹으로 구성된 시스템의 사용자 데이터베이스가 될 수도 있습니다.

일괄처리(Batch processing)

프로그램 또는 일련의 프로그램을 실행하는 방법으로 하나 또는 그 이상의 레코드(batch)가 사용자나 오퍼레이터의 조치없이 처리됩니다. 이는 대화식 처리와 대조를 이룹니다.

작업 단위(Unit of work)

작업 단위는 하나 이상의 데이터베이스 요구이며 완료된 작업을 구성하는 관련 처리입니다. 트랜잭션에 해당됩니다.

저널(Journal)

저널되고 있는 오브젝트, 현재 저널 리시버 및 저널을 위한 시스템 상의 모든 저널 리시버를 식별하는 시스템 오브젝트 이 오브젝트 유형의 시스템 인식 식별자는 *JRN입니다.

정의역명 시스템(DNS)

인터넷 프로토콜 집합에서 정의역명을 IP 주소로 맵핑시키는데 사용되는 분산 데이터베이스 시스템.

제어 서브시스템(Controlling subsystem)

시스템이 시작될 때 자동으로 처음 시작되는 대화식 서브시스템을 말하며 이를 통해 시스템 오퍼레이터가 시스템을 제어합니다.

첫번째 실패 자료 캡처(FFDC)

문제점 인식, 진단 자료의 선택적인 덤프, 증상 스트링 생성 및 문제점 기록부 항목을 제공하는 FFST™ 구조의 OS/400 구현.

카탈로그(Catalog)

표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가 들어 있는 표와 뷰의 세트. QSYS2의 카탈로그 뷰에는 iSeries 서버의 모든 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가 들어 있습니다. 또한 SQL 스키마에는 스키마의 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보만 들어 있는 뷰의 집합이 있습니다.

코드 페이지(Code page)

문자와 그 내부 코드를 지정한 것으로 이루어진 특정 세트

코드화 문자 세트 ID(CCSID: Coded character set identifier)

코드화 체계 ID, 문자 세트 ID, 코드 페이지 ID 및 사용된 코드화 그래픽 문자 표시를 고유하게 식별하는 코드 관련 정보의 특정 세트를 식별하는 16비트 숫자입니다.

트랜잭션 프로그램명(TPN: Transaction program name)

LU 6.2 대화에 참여하고 있는 각 프로그램을 인식할 수 있는 이름. 보통, 연결 개시자가 상대방 LU에서 연결하는 프로그램의 이름을 식별합니다. LU 이름과 함께 사용될 때, TPN은 네트워크의 특정 트랜잭션 프로그램을 식별합니다.

티켓 허가 티켓(TGT)

서비스 티켓이 요구될 때 프린시펄이 티켓 허가 서버에 전달하는 티켓. 티켓 허가 서비스는 티켓 허가 티켓을 사용하여 프린시펄이 서비스 티켓의 요구를 허가하기 전에 인증 서버로 인증했음을 확인합니다.

패키지(Package)

SQL 패키지 참조

페이싱(Pacing)

SNA에서 수신 시스템이 과수행을 방지하기 위해 시스템을 송신하는 전송 비율을 제어하는 기술

표시장치 pass-through(Display station pass-through)

사용자가 다른 iSeries 서버에서 한 iSeries 서버를 사인 온하고 서버의 프로그램과 자료를 사용할 수 있는 통신 기능.

호스트 변수(Host variable)

iSeries SQL 어플리케이션 프로그램용 DB2 UDB에서 삽입된 SQL문에 의해 참조되는 변수. RPG에서는 이 변수를 필드명이라 하고, C에서는 변수, COBOL에서는 자료 항목이라고 합니다.

호스트 언어(Host language)

iSeries SQL용 DB2 UDB에서 C, COBOL 및 RPG와 같은 프로그래밍 언어로 SQL문을 내장시킬 수 있습니다.

호스트 프로그램(Host program)

iSeries용 DB2 UDB에서 삽입된 SQL문이 포함된 호스트 언어로 쓰여진 프로그램

확약 제어(Commitment control)

확약 가능한 자원 연산을 그룹화하여 확약 가능한 자원 변경 그룹을 Commit 명령으로 하나의 단위로 처리하거나 확약 가능한 자원 변경 그룹을 Rollback 명령으로 하나의 단위로 제거하도록 허용하는 수단

확약(Commit)

요구가 완료되면 어플리케이션 프로그램은 작업 단위를 확약할 수 있습니다. 이는 작업 단위와 관련된 데이터베이스 변경이 영구적으로 이루어졌다는 의미입니다.

APPC(Advanced program-to-program communications)

iSeries 서버의 프로그램이 호환 가능한 통신 지원을 가지고 있는 다른 시스템의 프로그램과 통신할 수 있도록 하는 자료 통신 지원을 말합니다. iSeries 서버의 APPC는 SNA LU 유형 6.2 및 노드 유형 2.1 구조와의 어플리케이션 프로그래밍 인터페이스를 제공합니다. 이는 IBM 시스템 네트워크 구조(SNA)의 일부입니다.

APPN(Advanced Peer-to-Peer Networking(R))

직접 연결할 필요가 없는 두 개 또는 그 이상의 APPC 시스템 네트워크에서 자료를 라우트하는 자료 통신 지원에 에 관한 것입니다. 이는 IBM 시스템 네트워크 구조(SNA)의 일부입니다.

DB2 조회 관리자

iSeries 데이터베이스로부터 정보를 얻기 위해 사용되는 톨의 컬렉션인 DB2 조회 관리자 및 SQL 개발 킷 사용권 프로그램의 일부. 또한 DB2 조회 관리자는 새로운 또는 기존의 조회 정의를 실행하거나 조회 정보를 형식화하기 위해 조회 정의를 작성하는데도 사용됩니다.

DB2 조회 관리자 및 SQL 개발 킷

DB2 UDB 제품군의 하나인 IBM 사용권 프로그램. 조회 관리자는 사용자가 SQL 조회 및 보고서를 개발하도록 허용합니다. SQL 개발 킷은 프로그래머가 SQL 어플리케이션을 개발하도록 해 줍니다.

Enterprise Identity Mapping (EIM)

EIM은 사람이나 엔티티를 기업망을 통한 다양한 레지스트리에서 적절한 사용자 신원으로 맵핑(연관) 시키는 메카니즘입니다. EIM은 이 정보를 조회하기 위해 어플리케이션에 의해 사용되는 API뿐만 아니라 이 identity 맵핑 관계를 작성하고 관리하기 위한 API를 제공합니다.

IP 보안 구조(IPSec)

다양한 보안 서비스를 사용하여 IP 통신을 보호하기 위해 인터넷 프로토콜(IP) 계층에서 구조를 정의하는 Internet Engineering Task Force(IETF) 표준의 집합입니다.

iSeries용DB2 Universal Database(iSeries용 DB2 UDB)

iSeries의 통합된 관계형 데이터베이스 관리자. 자료에 대한 액세스와 자료 보호를 제공합니다. 이는 참조 무결성 및 병렬 데이터베이스 처리와 같은 확장 기능도 제공합니다.

Kerberos

메사추세츠 공과 대학(MIT's) Athena 프로젝트의 보안 시스템에 관한 것. 이는 대칭 키 암호를 사용하여 네트워크의 사용자에게 보안 서비스를 제공합니다.

Kerberos 구성 파일(krb5.conf)

Kerberos 영역 이름이 DNS 접미사 이름과 다른 경우에, 이는 올바른 영역으로 맵핑되어야만 합니다. 그러기 위해서, Kerberos 구성 파일(krb5.conf)에 항목이 있어야 각 리모트 호스트명을 올바른 영역 이름에 맵핑할 수 있습니다.

Kerberos 영역(Kerberos Realm)

Kerberos 영역은 Kerberos 서버내에서 등록된 Kerberos 프린시펄의 집합입니다.

SNA 분배 서비스(SNADS: SNA distribution services)

시스템 네트워크에서 전자 우편을 수신, 라우트 및 송신하는 규칙의 집합을 정의하는 IBM 비동기 분배 서비스입니다.

SNA 업라인 기능(SNUF)

iSeries 서버가 호스트 시스템의 CICS/VS 및 IMS/VS 어플리케이션 프로그램과 통신할 수 있도록 해 주는 통신 지원입니다. 예를 들어, DHCF는 HCF와 통신하고 DSNX는 NetView 분배 관리자 프로그램과 통신합니다.

SQL 패키지(SQL package)

SQL 패키지는 분산 관계형 데이터베이스에만 사용되는 iSeries 오브젝트입니다. SQL의 사전컴파일 프로세스의 결과로 작성되거나 컴파일된 프로그램 오브젝트로부터 작성될 수 있습니다. SQL 패키지는 어플리케이션 서버에 있습니다. 어플리케이션 서버에는 SQL문, 호스트 변수 속성 및 어플리케이션 서버가 어플리케이션 리퀘스터의 요구를 처리하는 데 사용하는 액세스 계획이 들어 있습니다.

Transmission Control Protocol/Internet Protocol (TCP/IP)

(1) 로컬 및 광역 네트워크의 일대일(peer-to-peer) 연결 기능을 지원하는 통신 프로토콜의 집합. (2) 인터넷에서 사용되는 1차 통신 프로토콜. TCP/IP도 내부 네트워크에서 사용될 수 있습니다.




부록 E. 관련 정보








다음 도서 리스트는 IBM에서 제공하는 이 설명서와 관련된 4가지 책 분류에 대한 것입니다. 분류는 다음과 같습니다.

- iSeries 서버 라이브러리
- 분산 관계형 데이터베이스 라이브러리
- 기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리
- 구조 설명서
- IBM 레드북™

iSeries 서버 정보

다음 iSeries 설명서에 사용자가 필요로 하는 정보가 들어 있습니다. 주문 번호는 주문과 참조 목적으로 제공 됩니다.

- *DSNX Support*  는 리모트 관리 지원(분산 호스트 명령 기능), 변경 관리 지원(분산 시스템 노드 실행) 및 문제점 관리 지원(경고)을 사용하는 데 iSeries 서버를 구성하는 정보를 제공합니다.
- iSeries Information Center의 백업 및 회복 주제에서는 데이터베이스 파일의 변경된 내용을 기록하고, 그 정보를 시스템 회복 및 활동 보고 정보에 사용할 수 있는 방법에 대한 설명 및 시스템 자료를 저장 및 복원하는 데 사용할 수 있는 여러가지 매체에 대한 정보를 시스템 프로그래머에게 제공합니다.
- iSeries Information Center의 CL 프로그래밍 주제에서는 오브젝트와 라이브러리, 제어 언어(CL) 프로그래밍, 프로그램 간의 제어 흐름과 통신, CL 프로그램에서의 오브젝트에 대한 작업 및 CL 프로그램 작성을 비롯하여 프로그래밍에 대한 광범위한 논의를 제공합니다. 다른 주제에는 사전 정의된 메시지와 메시지 처리, 사용자 정의 명령과 메뉴 정의 및 작성 그리고 디버그 모드, 중단점, 추적 및 표시장치 기능을 비롯한 어플리케이션 테스트에 대한 사항이 수록되어 있습니다.
- *Communications Management*  에는 통신 상태, 통신 관련 작업 관리 주제, 통신 오류, 성능, 회선 속도 및 서브시스템 기억장치에 대한 작업 정보가 들어 있습니다.
- iSeries Information Center의 분산 데이터베이스 관리 주제는 어플리케이션 프로그래머 또는 시스템 프로그래머에게 리모트 파일 처리에 대한 정보를 제공합니다. OS/400 분배 자료 관리(DDM)에 대해 리모트 파일을 정의하는 방법, DDM 파일을 작성하는 방법, DDM을 통해 지원되는 파일 유틸리티 및 다른 시스템과 관련된 OS/400 DDM의 요구 조건을 설명합니다.
- *Local Device Configuration*  은 시스템 오퍼레이터나 시스템 관리자에게 초기 로컬 하드웨어 구성 방법과 이 구성을 변경하는 방법에 대해 설명합니다. 또한 장치 구성에 관한 개념적 정보와 9406, 9404, 9402 시스템 장치의 장치 구성에 관한 계획 정보도 들어 있습니다.

- V5R1 Supplemental Manuals 웹 사이트의 *ADTS/400: Data File Utility*  는 어플리케이션 프로그래머, 프로그래머 또는 헬프 데스크 담당자에게 자료를 파일에 입력, 파일을 갱신, 파일을 조회하고 DFU 프로그램을 실행하는 프로그램을 작성하는 어플리케이션 개발 툴 자료 파일 유틸리티(DFU)에 대한 정보를 제공합니다. 이 안내서는 DFU를 배우는 데 필요한 활동과 자료를 워크스테이션 오퍼레이터에게 제공합니다.
- V5R1 Supplemental Manuals 웹 사이트의 *SNA Distribution Services*  는 시스템 프로그래머나 네트워크 관리자에게 분배 서비스(SNADS)와 가상 기계/복수 가상 기억장치(VM/MVS) 브릿지에 대한 통신 네트워크 구성 정보를 제공합니다. 또한 오브젝트 분배 기능, 문서 라이브러리 서비스 및 시스템 분배 디렉토리 서비스도 설명됩니다.
- *ICF Programming*  은 어플리케이션 프로그래머에게 iSeries 통신 및 ICF 파일을 사용하는 어플리케이션 프로그램을 작성하는 데 필요한 정보를 제공합니다. 자료 서술 스펙(DDS) 키워드, 시스템 제공 형식, 리턴 코드, 파일 전송 지원 및 프로그래밍 예제 등의 정보가 있습니다.
- *LAN, Frame-Relay and ATM Support*  에는 토큰링 네트워크, 이더넷 네트워크 또는 브릿지 네트워크 환경에서 iSeries 서버를 사용하는 데 대한 정보가 있습니다.
- *Remote Work Station Support*  는 어플리케이션 프로그래머나 시스템 프로그래머에게 명령을 구성하고 회선, 제어기 및 장치를 정의하는 데 대한 정보를 제공합니다.
- iSeries Information Center의 조회 관리 프로그래밍 주제는 어플리케이션 프로그래머에게 보고서를 조회할 데이터베이스 파일을 판별하고 SQL 조회 정의를 정의하고 조회 관리 명령을 사용하는 프로시저어를 사용하고 작성하는 방법에 대한 정보를 제공합니다. 또한 조회 글로벌 변수 지원을 사용하고 OS/400 조회 관리와 iSeries 조회 사용권 프로그램 사이의 관계를 이해하는 방법을 설명합니다.
- *Remote Work Station Support*  는 표시장치 pass-through, 분산 호스트 명령 기능 및 3270 리모트 접속과 같은 리모트 작업 장치 지원을 설정하고 사용하는 방법에 대한 정보를 제공합니다.
- iSeries Information Center의 보안 주제는 시스템 프로그래머(또는 보안 관리자의 권한이 할당된 사용자)에게 시스템 보안 개념, 보안 계획 및 시스템의 보안 설정에 대한 정보를 제공합니다.
- iSeries Information Center의 SQL 프로그래밍 개념 주제는 어플리케이션 프로그래머, 프로그래머 또는 데이터베이스 관리자에게 SQL문을 디자인, 기록, 테스트 및 실행하는 방법에 대한 개요를 제공합니다. 또한 대화식 SQL을 설명합니다.
- iSeries Information Center의 SQL 참조서 주제는 어플리케이션 프로그래머, 프로그래머 또는 데이터베이스 관리자에게 SQL문 및 그 매개변수에 대한 상세한 정보를 제공합니다.
- *X.25 Network Support*  에는 X.25 네트워크에서 iSeries 서버를 사용하는 데 대한 정보가 들어 있습니다.

분산 관계형 데이터베이스 라이브러리

다음 설명서는 IBM 분산 관계형 데이터베이스 구조 구현에 대한 백그라운드와 일반 지원 정보를 제공합니다.

- *DRDA: Every Manager's Guide*, GC26-3195는 분산 관계형 데이터베이스와 분배 파일에 대한 간략하고 상위 수준의 교육 내용을 제공합니다. 이 설명서는 IBM이 분산 자료 시스템의 개발을 지원하는 방법을 설명하고 최신 IBM 제품과 분산 자료에 대해 발표된 지원을 논의합니다. 이 설명서의 정보는 경영자, 관리자 및 기술 직원이 분산 자료의 개념을 이해하는 데 도움을 주기 위한 것입니다.
- *DRDA: Planning for Distributed Relational Database*, SC26-4650은 분산 관계 자료를 계획하는 데 도움을 줍니다. 이 설명서는 실행 단계, 의사 결정 및 의사 결정 시 선택할 옵션을 설명합니다. 또한 현재 사용 중이거나 발표된 분산 관계형데이터베이스 제품과 기능을 다루며 IBM이 언급한 분산 관계 자료의 향후 지원 방향을 논의합니다. 이 설명서의 정보는 계획 입안자를 위한 것입니다.
- *DRDA: 연결성 안내서* SC26-4783은 분산 관계형 데이터베이스 구조를 지원하는 IBM 제품을 상호연결하는 방법을 설명합니다. 이 설명서는 분산 관계형 데이터베이스와 네트워크 시스템이 관련된 개념과 용어를 설명합니다. 또한 분산 환경에서 상이한 시스템을 연결하는 방법을 알려 줍니다. 연결성 안내서의 정보는 다른 제품의 설명서에는 들어 있지 않습니다. 이 설명서의 정보는 시스템 관리자, 데이터베이스 관리자, 통신 관리자 및 시스템 프로그래머를 위한 것입니다.
- *DRDA: 어플리케이션 프로그래밍 안내서*, SC26-4773은 IBM의 관계형 데이터베이스 관리 시스템에 액세스하는 어플리케이션 프로그램을 설계, 빌드 및 수정하는 방법을 설명합니다. 이 매뉴얼은 상이한 환경에 대한 분산 관계형 데이터베이스 어플리케이션을 기록할 때 프로그래머가 달리 해야 할 일에 중점을 둡니다. 주제에는 성능 고려사항은 물론 프로그램 설계, 준비 및 실행이 포함됩니다. IBM C로 작성한 프로그램 예가 포함되어 있습니다. 이 매뉴얼의 정보는 IBM의 상위 수준 언어를 적어도 하나 사용하고 SQL을 사용하여 작업하는 프로그래머를 위해 설계되었습니다.
- *DRDA: 문제점 판별 안내서*, SC26-4782는 분산 관계형 데이터베이스 환경에서의 문제점 소스를 정의하는 데 도움을 줍니다. 이 매뉴얼에는 이들 제품에 익숙하지 않은 사용자를 위해 각 제품에 대해 소개하며 제품의 문제점을 진단하고 보고하는 방법에 대한 자세한 정보를 제공합니다. 안내서는 각 호스트 시스템에 고유하고 여러 시스템에 공통된 프로시저어와 툴을 설명합니다. 이 설명서의 정보는 IBM 지원 센터에 분산 관계형 데이터베이스 문제점을 보고하는 사용자를 위한 것입니다.
- *IBM SQL 참조서*, 볼륨 2, SC26-8416은 DRDA를 참조하고 다음 기능을 비교합니다.
 - IBM SQL 관계형 데이터베이스 제품
 - IBM SQL
 - ISO-ANSI SQL(SQL92E)
 - X/Open SQL(XPG4-SQL)
 - ISO-ANSI SQL 호출 레벨 인터페이스(CLI)
 - X/Open CLI
 - Microsoft ODBC(개방 데이터베이스 연결성) 버전 2.0

Other IBM 분산 관계형 데이터베이스 플랫폼 라이브러리

DB2 Connect 및 Universal Database

DB2 Connect 및 Universal Database에 작업하고 있고 이에 관한 자세한 정보를 원하시면 Knowledge Base: Windows, OS/2, UNIX용 DB2 Universal Database 및 DB2 Connect 웹 페이지를 참조하십시오. 다음의 책과 그 외 다른 버전에 관한 책도 구할 수 있습니다. (그러나 이 매뉴얼에 설명된 모든 기능이 모든 버전에서 지원되지 않는다는 점을 주의하십시오.)

- *DB2 Connect Enterprise Edition Quick Beginning*
- *DB2 Connect Personal Edition*용 빠른 시작
- *DB2 Connect* 사용자 안내서
- *DB2 UDB for OS/2 Quick Beginnings*
- *DB2 UDB for UNIX Quick Beginnings*
- *DB2 UDB for Windows NT[®] Quick Beginnings*
- *DB2 UDB Personal Edition*용 빠른 시작
- *DB2 UDB SQL Getting Started*
- *DB2 UDB* 관리 안내서
- *DB2 UDB SQL* 참조
- *DB2 UDB Command Reference*
- *DB2 UDB Messages Reference*
- *DB2 UDB* 문제점 해결 안내서

OS/390[®] 및 z/OS용 DB2

OS/390 및 z/OS용 DB2에 대해 작업하고 있고 이에 관한 자세한 정보를 원한다면 OS/390 및 z/OS용 DB2 웹 페이지를 참조하십시오. 다음의 책과 그 외 다른 버전에 관한 책도 구할 수 있습니다. (그러나 이 매뉴얼에 설명된 모든 기능이 모든 버전에서 지원되지 않는다는 점을 주의하십시오.)

- *DB2 for OS/390 Command Reference*
- *DB2 for OS/390 Reference for Remote DRDA*
- *DB2 for OS/390 SQL Reference*
- *DB2 for OS/390 Utility Guide and Reference*
- *DB2 for OS/390 Messages and Codes*

VSE & VM용 DB2 서버

VSE & VM용 DB2 서버에 대해 작업하고 있고 이에 관한 자세한 정보를 원한다면 VSE & VM용 DB2 웹 페이지를 참조하십시오. 다음의 책과 그 외 다른 버전에 관한 책도 구할 수 있습니다. (그러나 이 매뉴얼에 설명된 모든 기능이 모든 버전에서 지원되지 않는다는 점을 주의하십시오.)

- *SBOF for DB2 Server for VM*

- *DB2 and Data Tools for VSE and VM*
- *DB2 Server for VM Database Administration*
- *DB2 Server for VM Application Programming*
- *DB2 Server for VM Database Services Utilities*
- *DB2 Server for VM Messages and Codes*
- *DB2 Server for VM Master Index and Glossary*
- *DB2 Server for VM Operation*
- *DB2 Server for VSE & VM Quick Reference*
- *DB2 Server for VSE & VM SQL Reference*
- *DB2 Server for VM System Administration*
- *DB2 Server for VM Diagnosis Guide*
- *DB2 Server for VM Interactive SQL Guide*
- *DB2 Server Data Spaces Support for VM/ESA®*
- *DB2 Server for VSE & VM LPS*
- *DB2 Server for VSE & VM Data Restore*
- *DB2 for VM Control Center Installation*
- *DB2 Server for VM/VSE Training Brochure*

구조 설명서

- *Character Data Representative Architecture: Overview, GC09-2207*
- *Character Data Representative Architecture: Details, SC09-2190*

이 설명서에는 온라인 BOOK 형식으로 된 두 가지 CDRA 책, 2진 양식의 변환 표, 많은 2진 변환의 맵핑 소스, 코드 페이지 및 문자 세트 자원의 콜렉션과 IBM에서 사용되는 문자 명명 정보가 들어 있는 CD-ROM 이 있습니다. CD는 또한 제공된 자료로 사용될 보기 유틸리티를 포함합니다. 뷰는 OS/2, Windows 3.1 및 Windows 95로 작업합니다.

- *DRDA Vol. 1: Distributed Relational Database Architecture(DRDA)*

이러한 기술 표준은 분산 관계형 데이터베이스 구조 스펙을 문서화한 세 권 중 한 권입니다. 제 1권은 어플리케이션 프로그램으로 하여금 분산 관계형 자료에 액세스할 수 있게 하는 관계형 데이터베이스 관리자 간의 연결성에 대해 설명합니다. 제1권은 분산 환경에서 어플리케이션과 관계형 데이터베이스 관리 시스템 간의 필요한 연결, 참여자의 책임 및 흐름이 발생되어야 하는 시기, 분산 데이터베이스 관리 시스템 처리에 필요한 형식 및 프로토콜에 대해 설명합니다. 분산 데이터베이스 관리 시스템 처리를 위한 API에 대해서는 설명하지 않습니다. 본 문서는 웹 사이트(<http://www.opengroup.org/dbiop/index.htm>)에서 구할 수 있습니다.

- *DRDA Vol. 2: Formatted Data Object Content Architecture(FD:OCA)*

이러한 기술 표준은 분산 관계형 데이터베이스 구조 스펙을 문서화한 세 권 중 한 권입니다. 제 2권은 형식화된 자료 오브젝트 내용 구조(FD:OCA)를 구성하는 기능 및 서비스에 대해 설명합니다. 이 구조는 서로 다른 자료 유형 및 자료 표시 방법을 가진 환경 간의 연결성 간격을 브릿지할 수 있게 합니다. FD:OCA는 DRDA에 삽입됩니다. 본 문서는 웹 사이트(<http://www.opengroup.org/dbiop/index.htm>)에서 구할 수 있습니다.

- *DRDA Vol. 3: Distributed Data Management (DDM) Architecture*

이러한 기술 표준은 분산 관계형 데이터베이스 구조 스펙을 문서화한 세 권 중 한 권입니다. 제3권은 DDM 자료 스트림의 구조화된 명령, 매개변수, 오브젝트 및 메시지에 대해 설명합니다. 이 자료 스트림은 DDM 모델의 다양한 부분 간의 자료 교환을 실행합니다. 본 문서는 웹 사이트(<http://www.opengroup.org/dbiop/index.htm>)에서 사용 가능합니다.

레드북

- *Distributed Relational Database: Using DDCS/6000 DRDA Support with DB2 and DB2/400*, GG24-4155
- *Setup and Usage of SQL/DS in a DRDA Environment*, GG24-3733
- *DRDA Client/Server Application Scenarios*, GG24-4193
- *DRDA Client/Server for VM & VSE Setup*, GG24-4275
- *DATABASE 2/400 Advanced Database Functions*, GG24-4249
- *Distributed Relational Database Cross Platform Connectivity and Application*, GG24-4311
- *Getting Started with DB2 Stored Procedures: Give Them a Call through the Network*, GG24-4693
- *WOW! DRDA Supports TCP/IP: DB2 Server for OS/390*, SG24-2212, SG24-2212-00

부록 F. 주의사항

이 정보는 미국에서 제공되는 제품 및 서비스용으로 작성된 것입니다. IBM은 다른 국가에서는 이 자료에 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 언급했다는 것이 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 수 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.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터
전화번호: 080-023-8080

2바이트(DBCS) 정보에 관한 라이선스 문의는 한국 IBM 고객만족센터에 문의하거나 다음 주소로 서면 문의하시기 바랍니다.

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, 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 일체의 보증없이 이 책을 『현상태대로』 제공합니다. 일부 국가에서는 특정 거래에서 명시적 또는 묵시적 보증의 면책사항을 허용하지 않으므로, 이 사항이 적용되지 않을 수도 있습니다.

이 정보에는 기술적으로 부정확한 내용이나 인쇄상의 오류가 있을 수 있습니다. 이 정보는 주기적으로 변경되며, 이 변경사항은 최신판에 통합됩니다. IBM은 이 책에 설명한 제품 및(또는) 프로그램을 사전 통지없이 언제든지 개선 및(또는) 변경할 수 있습니다.

(1) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (2) 교환된 정보의 상호 이용을 목적으로 정보를 원하는 프로그램 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

서울특별시 강남구 도곡동 467-12, 군인공제회관빌딩
한국 아이.비.엠 주식회사
고객만족센터

이러한 정보는 해당 조건(예를 들어, 사용료 지불 등)에 따라 사용될 수 있습니다.

이 정보에 기술된 라이선스가 있는 프로그램 및 이 프로그램에 대해 라이선스가 있는 자료는 IBM이 IBM 기본 계약 또는 이와 동등한 계약에 따라 제공한 것입니다.

이 정보에는 일상의 비즈니스 운영에서 사용되는 자료 및 보고서에 대한 예제가 들어 있습니다. 이 예제에는 가능한 완벽하게 개념을 설명하기 위해 개인, 회사, 상표 및 제품의 이름이 사용될 수 있습니다. 이들 이름은 모두 가공의 것이며 실제 기업의 이름 및 주소와 유사하더라도 이는 전적으로 우연입니다.

라이선스:

이 정보에는 여러 운영 플랫폼에서의 프로그래밍 기법을 보여주는 원시 언어로 된 샘플 응용프로그램이 들어 있습니다. 귀하는 샘플 프로그램의 작성 기준이 된 운영 플랫폼의 응용프로그램 프로그래밍 인터페이스에 부합하는 응용프로그램을 개발, 사용, 마케팅 및 배포하기 위한 목적으로 IBM에 추가 비용없이 어떤 형태로든 이러한 샘플 프로그램을 복사, 수정 및 배포할 수 있습니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다. 귀하는 IBM의 응용프로그램 프로그래밍 인터페이스(API)에 부합하는 응용프로그램을 개발, 사용, 판매 또는 배포할 목적으로 추가 비용없이 이러한 샘플 응용프로그램을 어떠한 형태로든 복사, 수정 및 배포할 수 있습니다.

상표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 등록상표입니다.

Advanced Peer-to-Peer Networking

AIX

AnyNet

C/400

CICS

COBOL/400

DataJoiner

DataPropagator

DB2

DB2 Connect

DB2 Universal Database Distributed Relational Database Architecture

DRDA

DXT

Extended Services

FFST
IBM
IMS Informix
iSeries
Language Environment
MVSNetView
Operating System/400
OS/2
OS/390
OS/400
PS/2
Redbooks
RPG/400
RS/6000
SQL/DS
System/390
VM/ESA
z/OS

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc의 상표입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Intel, Intel Inside(로고), MMX 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표입니다.

UNIX는 미국 또는 기타 국가에서 사용되는 The Open Group의 등록상표입니다.

Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 상표입니다.

기타 회사, 제품 및 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

코드 면책사항 정보

이 문서에는 프로그래밍 예제가 들어 있습니다.

IBM은 사용자의 특정 요구에 맞게 유사한 기능을 생성할 수 있도록 모든 프로그래밍 코드 예제를 사용할 수 있는 비독점적인 저작권 라이선스를 부여합니다.

모든 샘플 예제는 IBM에 의해 예시 목적으로만 제공됩니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 샘플 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다.

여기에 포함된 모든 프로그램은 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여 일체의 보증 없이 "현상태대로" 제공됩니다. 타인의 권리 침해, 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증은 명시적으로 거부됩니다.

서적의 다운로드 및 인쇄 조건

귀하가 다운로드하려는 서적을 사용하는 데에는 다음의 조건이 적용되며 귀하가 이를 승인하는 경우에 해당 서적을 사용할 수 있습니다.

개인적인 사용: 일체의 소유권 표시를 하는 경우에 한하여 귀하는 이들 서적을 개인적이며 비상업적인 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적인 동의없이 해당 서적에 대한 2차적 저작물 또는 그 일부를 배포, 전시 또는 작성할 수 없습니다.

상업적 사용: 일체의 소유권 표시를 하는 경우에 한하여 귀하는 이들 서적을 귀하 사업장 내에서만 복제, 배포 및 전시할 수 있습니다. 귀하는 IBM의 명시적인 동의없이 귀하의 사업장 이외에서 해당 서적의 2차적 저작물을 작성할 수 없으며 이들 서적 또는 그 일부를 복제, 배포 또는 전시할 수 없습니다.

본 계약에서 명시하지 않는 한, 본 서적 또는 본 서적에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대하여 다른 허가나 라이선스 또는 권리가 부여되지 않습니다.

해당 서적의 사용이 IBM에게 손해를 가져오거나, 상기 지시사항이 적절하게 준수되지 않은 것으로 IBM이 판단한 경우, IBM은 본 계약에서 부여한 서적에 대해 허가를 취소할 권리가 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하지 않는 경우 본 정보를 다운로드, 송신 또는 재송신할 수 없습니다. IBM은 이들 서적의 내용과 관련하여 어떠한 보증도 하지 않습니다. 본 서적은 상품성 및 특정 목적에의 적합성에 대한 묵시적 보증을 포함하여(단, 이에 한하지 않음) 묵시적이든 명시적이든 일체의 보증없이 "현상태대로" 제공됩니다.

All material copyrighted by IBM Corporation.

귀하는 본 사이트로부터 서적을 다운로드하거나 인쇄함으로써 본 조건에 동의한 것으로 간주됩니다.

색인

[가]

감사

- 관계형 데이터베이스 디렉토리 124
- 관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 124
- 관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 124
- ADDRDBDIRE(Add Relational Database Directory Entry) 124
- CHGRDBDIRE(Change Relational Database Directory Entry) 124
- DSRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 124
- RMVRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 124
- WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 124

개념 및 용어 5

개발

- 관리 전략 20

결과 세트

- 정의 206

계정

- 계획 23

계획

- 백업 24
- 보안 22
- 일반 작업 20
- 회복 24

고려사항

- 어플리케이션 프로그래밍 194
- CCSID 253

공동-플랫폼 DRDB 주 253

관계형 데이터베이스

- 정의 1

관계형 데이터베이스 디렉토리

- 감사 124
- 로컬 항목 75
- 명령 76
- 별명
 - RDB 76
- 복원 138
- 선택형 매개변수 76

관계형 데이터베이스 디렉토리 (계속)

- 설정 75
- 설정 예 81
- 저장 138
- 정의 69
- 출력 파일 작성 138
- 항목 변경 80
- 항목 제거 79
- 항목 표시 79
- 항목에 대한 작업 79
- CL 프로그램 사용 83
- RDB(관계형 데이터베이스) 매개변수 76
- RMTLOCNAME 매개변수 76

관계형 데이터베이스 디렉토리 항목

- 작업 대상 79

관계형 데이터베이스 활동

- 모니터 97

관계형 데이터베이스명

- 관계형 데이터베이스 디렉토리에서 76

- 내재 CONNECT 201

관계형 데이터베이스(RDB) 디렉토리 항목

- 변경 80, 188
- 제거 79
- 추가 76, 138, 188
- 표시 79, 138

관계형 데이터베이스(RDB) 디렉토리 항목 변경

- (CHGRDBDIRE) 명령 80, 124, 188

관계형 데이터베이스(RDB) 디렉토리 항목 추가

- (ADDRDBDIRE) 명령 76, 124, 138, 188

관리 및 조작 97

관리 전략

- 개발 20

관리 타스크

- 리모트 명령 제출 105
- 리모트 서버 시작 및 중단 105
- 리모트로 서버 조작 105
- 분산 관계형 데이터베이스 작업 찾기 103
- 사용자 작업(jobs)에 대한 작업 99
- 작업 기록부 표시 102
- 작업 사용 계정 112
- 작업(jobs)에 대한 작업 98
- 확약 정의에 대한 작업 101
- 활동 작업(job)에 대한 작업 100

구성

- 변환 29, 142
- 복원 137
- 통신
 - 네트워크 인터페이스 설명 28
 - 단계 28
 - 제어기 설명 29
 - 회선 설명 28
- APPN 네트워크 노드 30

구성 상태

- 작업 대상 29, 142

권한

- 복원 137
- 저장 137

기능

- 분산 관계형 데이터베이스(DRDB) 16

[나]

내재적 연결 201

네트워크

- 고려사항 19
- 설계 18
- 성능 개선 145
- 요구사항 19

네트워크 고려사항

- DRDA 지원 26

네트워크 구성 예 30

네트워크 속성

- 변경 28
- 네트워크 중복 141

[다]

대기 문제점

- 어플리케이션 리퀘스터 155
- 어플리케이션 서버 156

대화

- SNA 대 TCP/IP 107

대화식 작업 70, 71

대화식 SQL

- 서버간 자료 이동 89
- 확약 제어 시작 133
- 대화식 SQL 및 조회 관리 설정 256

덤프, FFDC 274

데이터베이스

- 보안 41

데이터베이스 관리

- 리포트 명령 제출 105
- 리포트 서버 시작 및 중단 105
- 리포트로 서버 조작 105
- 분산 관계형 데이터베이스 작업 찾기 103
- 작업 기록부 표시 102
- 작업 대상
 - 사용자 작업 99
 - 작업 98
 - 확약 정의 101
 - 활동 작업 100
- 작업 사용 계정 112

데이터베이스 회복

- 디스크 장애 128
- 메소드 127
- 무정전 전원 장치(uninterruptible power supply) 128
- 보조 기억장치 풀 129
- 색인 재설정 131
- 색인 재설정 시간 줄이기 132
- 이중복사 보호 129
- 장애 유형 127
- 저널 관리 130
- 저널 리시버 입력 변환 131
- 체크섬 보호 129

데이터베이스, 성능 개선 147

독립 ASP

- 구성 19
- 연결변환/단절변환 130

디버그

- 시작 187

디버그 시작(STRDBG) 명령 187

디버그와 테스트

- 어플리케이션 프로그램 218

디스크 상태

- 작업 대상 146

디스크 상태에 대한 작업(WRKDSKSTS) 명령 146

디스크 장애

- 보조 기억장치 풀 129
- 이중복사 보호 129
- 체크섬 보호 129

디폴트 콜렉션 명 196

[라]

라이브러리

- 복원 137
- 저장 132, 136

라이브러리 복원(RSTLIB) 명령 137

라이브러리 저장(SAVLIB) 명령 132, 136

롤백

- 정의 3

루프 문제점

- 어플리케이션 리퀘스터 155
- 어플리케이션 서버 156

리포트 데이터베이스

- 정의 1

리포트 명령

- 제출 105, 106, 111

리포트 서버 조작

- 리포트 명령 제출 105
- 시작 및 중단 105

리포트 작업 단위(RUW) 196

- 정의 4

리포트 프로시유어 호출 206

리스너 프로그램 115

[마]

메세지

- 문제점 처리 160
- 분류 설명 160
- 분산 관계형 데이터베이스(RDB) 163
- 시작된 목표 DDM 작업 105
- 심각도 코드 162
- 엑세스된 데이터베이스 105
- 유형 162
- 정보용 160
- 조회 160
- 추가 메세지 정보 화면 161
- 프로그램 요구 실패 165
- DDM 작업 시작 103

메세지 분류 160

메세지 설명

- 표시 161

메세지 설명 표시(DSPMSGD) 명령 161

명령, CL

- 관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 80, 124, 188

명령, CL (계속)

- 관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 76, 124, 138, 188
- 디버그 시작(STRDBG) 187
- 디스크 상태에 대한 작업 (WRKDSKSTS) 146
- 라이브러리 복원(RSTLIB) 137
- 라이브러리 저장(SAVLIB) 132, 136
- 메세지 설명 표시(DSPMSGD) 161
- 변경된 오브젝트 저장 (SAVCHGOBJ) 137
- 복사 시작 화면(STRCPYSCRN) 159
- 서비스 작업 시작(STRSRVJOB) 187
- 시스템 상태에 대한 작업 (WRKSYSSTS) 146
- 시스템 저장(SAVSYS) 137
- 프로그램 참조 표시(DSPPGMREF) 108, 219
- 활동 작업에 대한 작업 (WRKACTJOB) 100, 146
- ADDRDBDIRE(Add Relational Database Directory Entry) 76, 124, 138
- ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 188
- CHGJOB(작업 변경) 108
- CHGNETA(네트워크 속성 변경) 28
- CHGOBJAUD(오브젝트 감사 값 변경) 127
- CHGRDBDIRE(Change Relational Database Directory Entry) 80, 124
- CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 188
- CHGSBSD(서브시스템 설명 변경) 71
- CHGSBSD(서브시스템 설명 변경) 명령 71
- CRTSQLCBLI(SQL COBOL ILE 작성) 216
- CRTSQLCBL(SQL COBOL 작성) 216
- CRTSQLCI(SQL C ILE 작성) 216
- CRTSQLC(SQL C 작성) 216
- CRTSQLFTN(SQL FORTRAN 작성) 216
- CRTSQLPKG(SQL 패키지 작성) 220
- CRTSQLPLI(SQL PL/I 작성) 216
- CRTSQLRPGI(SQL RPG ILE 작성) 216
- CRTSQLRPG(SQL RPG 작성) 216

명령, CL (계속)

DSPJOBLOC(작업 기록부 표시) 102
 DSPJOBLOG(작업 기록부 표시) 102
 DSPJRN(저널 표시) 23, 131
 DSPMSGD(Display Message Descriptions) 161
 DSPPGMREF(프로그램 참조 표시) 108, 219
 DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 79, 124, 138
 Pass-Through 시작(STRPASTHR) 159
 RCLDDMCNV(분산 자료 관리 대화 생성) 108
 RCLRSC(자원 재생) 108
 RMRVDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 79, 124
 RSTAUT(권한 복원) 137
 RSTCFG(구성 복원) 137
 RSTLIB(라이브러리 복원) 137
 RSTOBJ(오브젝트 복원) 137, 138, 140
 RSTUSRPRF(사용자 프로파일 복원) 137
 SAVCHGOBJ(변경된 오브젝트 저장) 137
 SAVLIB(라이브러리 저장) 132, 136
 SAVOBJ(오브젝트 저장) 132, 137, 138
 SAVSAVFDTA(저장 파일 자료 저장) 137
 SAVSECDTA(보안 자료 저장) 137
 SAVSYS(시스템 저장) 137
 SBMRMTCMD(리모트 명령 제출) 105, 111
 권한 제한사항 106
 SQL C ILE 작성(CRTSQLCI) 216
 SQL C 작성(CRTSQLC) 216
 SQL COBOL ILE 작성(CRTSQLCBLI) 216
 SQL COBOL 작성(CRTSQLCBL) 216
 SQL FORTRAN 작성(CRTSQLFTN) 216
 SQL PL/I 작성(CRTSQLPLI) 216
 SQL RPG ILE 작성(CRTSQLRPGI) 216
 SQL RPG 작성(CRTSQLRPG) 216
 SQL 패키지 작성(CRTSQLPKG) 220
 STRCMTCTL(확약 제어 시작) 133
 STRCPYSCRN (Start Copy Screen) 159
 STRDBG(디버그 시작) 187

명령, CL (계속)

STRJRNAP(저널 액세스 경로 시작) 131
 STRPASTHR(Start Pass-Through) 159
 STRSRVJOB(서비스 작업 시작) 187
 VRYCFG (구성 변환) 29, 142
 VRYCFG(구성 변환) 29, 142
 WRKACTJOB(Work with Active Jobs) 100, 146
 WRKCFGSGS(구성 상태에 대한 작업) 29, 142
 WRKCFGSTS(구성 상태에 대한 작업) 29, 142
 WRKDSKSTS(Work with Disk Status) 146
 WRKJOB(작업(Job)에 대한 작업) 98
 WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 79, 124
 WRKSYSSTS(Work with System Status) 146
 WRKJOB(작업(Job)에 대한 작업) 98
 WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 79, 124
 WRKSYSSTS(Work with System Status) 146
 WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 99
 WRKUSRJOB(사용자 작업(Jobs)에 대한 작업) 99
 명령, 사전컴파일러
 CRTSQLCBLI(SQL COBOL ILE 작성) 216
 CRTSQLCBL(SQL COBOL 작성) 216
 CRTSQLCI(SQL C ILE 작성) 216
 CRTSQLFTN(SQL FORTRAN 작성) 216
 CRTSQLPLI(SQL PL/I 작성) 216
 CRTSQLRPGI(SQL RPG ILE 작성) 216
 CRTSQLRPG(SQL RPG 작성) 216
 SQL C ILE 작성(CRTSQLCI) 216
 SQL COBOL ILE 작성(CRTSQLCBLI) 216
 SQL COBOL 작성(CRTSQLCBL) 216
 SQL FORTRAN 작성(CRTSQLFTN) 216
 SQL PL/I 작성(CRTSQLPLI) 216
 SQL RPG ILE 작성(CRTSQLRPGI) 216
 SQL RPG 작성(CRTSQLRPG) 216
 명령 규칙
 디폴트 콜렉션 명 196
 시스템 195
 SQL 195

명시적 연결 203

모니터
 관계형 데이터베이스 활동 97
 무정전 전원 장치(uninterruptible power supply) 128
 문자 변환 7
 문자 자료 표시 구조(CDRA) 7, 209
 DRDA 사용 7
 문제점
 사용자 감지 154
 시스템 감지 154
 처리 153
 문제점 기록부 176
 문제점 분석, 계획 23
 문제점 처리 273
 개요 153
 대기, 루프, 성능 문제점
 어플리케이션 리퀘스터 155
 어플리케이션 서버 156
 메세지 분류 160
 메세지 설명 표시 161
 메세지 심각도 162
 문제점 기록부 176
 문제점 분석(ANZPRB) 명령 176
 분리되는 분산 관계형 데이터베이스(RDB) 문제점 154
 분산 관계형 데이터베이스(RDB) 메세지 163
 사용자 감지 문제점 154
 사용자에 대한 작업 158
 시스템 감지 문제점 154
 시스템 메세지 160
 어플리케이션 문제점 168
 오류 기록부 179
 작업 기록부 178
 작업 추적 180
 추가 메세지 정보 화면 161
 통신 추적 181
 표시장치 복사 159
 표시장치 Pass-Through 사용 159
 프로그램 요구 실패 165

[바]

바인드 옵션 260
 백업
 계획 24

- 변경
 - 관계형 데이터베이스(RDB) 디렉토리 항목 80, 188
 - 네트워크 속성 28
 - 서브시스템 설명 71
 - 오브젝트 감사 값 127
 - 작업 108
- 변경된 오브젝트
 - 저장 137
- 변경된 오브젝트 저장(SAVCHGOBJ) 명령 137
- 변환
 - 구성 29, 142
- 변환 고려사항
 - CCSID 253, 255
 - DB2 255
 - DB2 Connect 사용권 프로그램 253
 - DB2 Server for VM 데이터베이스 관리자 255
- 별명
 - RDB 76, 78
- 보류 연결 상태 199
- 보류 작업 111
- 보안
 - 감사 124
 - 계획 22
 - 네트워크 전역에 대한 일정한 시스템 레벨 42
 - 분산 데이터베이스 개요 42
 - 설정 84
 - 암호 22, 205
 - 암호화 22
 - 어플리케이션
 - 리퀘스터 42
 - 서버 42
 - 프로파일과 권한 복원 137
 - 프로파일과 권한 저장 137
 - iSeries 분산 관계형 데이터베이스에 대한 41
- 보안 자료
 - 저장 137
- 보조 기억장치 풀 129
- 보호
 - 시스템 관리식 액세스 경로 132
- 복원
 - 관계형 데이터베이스 디렉토리 138
 - 구성 137
 - 권한 137

- 복원 (계속)
 - 라이브러리 136, 137
 - 보안 자료 137
 - 사용자 프로파일 137
 - 색인 137
 - 오브젝트 136, 137, 138, 140
 - 저장 파일로부터 136
 - 테이프 또는 디스켓으로부터 136
 - SQL 패키지 138
- 분산 관계형 데이터베이스 구조(DRDA) 지원
 - 개요 6
 - 레벨 1 지원 1
 - 레벨 2 지원 1
 - 현재 iSeries 지원 9
 - CDRA 사용 7
 - DDM과 공존 26
 - 분산 관계형 데이터베이스 기능 16
 - 분산 관계형 데이터베이스 보안 41
 - 분산 관계형 데이터베이스 설정 69
 - 분산 관계형 데이터베이스 오브젝트 명명 195
 - 분산 관계형 데이터베이스의 계획 15
 - 분산 관계형 데이터베이스의 설계 15
 - 분산 관계형 데이터베이스의 특정 SQL 205
 - 분산 관계형 데이터베이스(DRDB)
 - 관리 10
 - 관리 및 조작 97
 - 리모트 작업 단위 196
 - 설정 69
 - 특정 SQL 205
 - 분산 관계형 데이터베이스(DRDB) 문제점 처리 153
 - 분산 관계형 데이터베이스(DRDB) 어플리케이션
 - 분산 관계형 데이터베이스(DRDB)의 고려 사항 194
 - 프로그래밍 고려사항 194
 - 분산 관계형 데이터베이스(DRDB) 어플리케이션 쓰기 193
 - 분산 관계형 데이터베이스(RDB) 문제점
 - 대기, 루프, 성능
 - 어플리케이션 리퀘스터 155
 - 어플리케이션 서버 156
 - 부정확한 출력 155
 - 분산 자료 관리 대화
 - 재생 108
 - 분산 작업 단위(DUW)
 - 어플리케이션 설계 추가 정보 18
 - 정의 4

- 분석
 - RW 추적 자료 268
- 뷰(view)
 - 정의 1
 - 회복 131
- 블로킹
 - 영향을 주는 요소 147
- 블로킹에 영향을 주는 요소 147
- 블록
 - 크기 요소 150

[사]

- 사용자 감지 문제점 154
- 사용자 데이터베이스
 - 정의 1
 - ‘위치’와 연관 19
- 사용자 작업
 - 작업 대상 99
- 사용자 종료 프로그램 63
 - 기능 검사 62
 - 예 62
 - DDM 74
- 사용자 프로파일
 - 복원 137
 - 저장 137
 - CCSID 210
- 사전시작 작업 70
- 사전시작 작업, 사용 117
- 사전컴파일 프로세스
 - 개요 214
 - 명령 215
 - 임시 소스 파일 멤버 215
 - 출력 리스트 214
 - SQL 패키지 215
- 사전컴파일러 명령
 - CRTSQLCBLI(SQL COBOL ILE 작성) 216
 - CRTSQLCBL(SQL COBOL 작성) 216
 - CRTSQLCI(SQL C ILE 작성) 216
 - CRTSQLFTN(SQL FORTRAN 작성) 216
 - CRTSQLPLI(SQL PL/I 작성) 216
 - CRTSQLRPGI(SQL RPG ILE 작성) 216
 - CRTSQLRPG(SQL RPG 작성) 216
 - SQL C ILE 작성(CRTSQLCI) 216

사전검색파일러 명령 (계속)

- SQL COBOL ILE 작성 (CRTSQLCBLI) 216
- SQL COBOL 작성(CRTSQLCBL) 216
- SQL FORTRAN 작성 (CRTSQLFTN) 216
- SQL PL/I 작성(CRTSQLPLI) 216
- SQL RPG ILE 작성 (CRTSQLRPGI) 216
- SQL RPG 작성(CRTSQLRPG) 216

상이한 환경

- 정의 6

상태

- SQL 연결 199

색인

- 재설정 131
- 저널링 131
- 저널링 시작 131
- 저널링 제한사항 132
- 저장 및 복원 137
- 정의 1
- 표 설계 고려사항 132
- 회복 131

서버

- 성능 개선 146
- 어플리케이션 서비스 작업 시작 187

서버 권한 항목 84

서버 메시지

- 분류 160

서브시스템 100, 120

- 사용자 정의 115
- 설명 117
- 설정 고려사항 71
- 정의 70
- 제어 70
- 통신 67
- IBM 제공 70
- QBASE 70, 72
- QBATCH 71
- QCMN 71, 72
- QCTL 71, 72
- QCTLSBSD 시스템 값 71
- QINTER 71, 72
- QSPL 71
- QSYSWRK 71

서브시스템 설명

- 변경 71

서비스 작업

- 시작 187
- 어플리케이션 서버 187
- 서비스 작업 시작 187
- 서비스 작업 시작(STRSRVJOB) 명령 187
- 설계

- 네트워크 18
- 어플리케이션 18
- 자료 18

설명

- FFDC 덤프 출력 277
- RW 추적점 269

설정

- 대화식 SQL 256
- 조회 관리 256

성능

- 네트워크 개선 145
- 데이터베이스 개선 147
- 분산 관계형 데이터베이스(DRDB) 145
- 블로킹 147
- 블록된 조회 자료 258
- 서버 관찰 146
- 서버를 통한 개선 146
- 연결 상 지연 157
- 영향을 주는 요소 147
- 자료 위치 결정 147

성능 문제점

- 어플리케이션 서버 156

스키마

- 정의 1

스플린 작업 70

시스템

- 명명 규칙 195
- 용어 1
- 저장 137

시스템 감지 문제점 154

시스템 값

- QCCSID 254

시스템 데이터베이스

- 정의 1

시스템 메시지

- 리턴된 SQLCODE 163
- 메세지 설명 표시 161
- 분산 관계형 데이터베이스(RDB) 163
- 심각도 코드 162
- 유형 160, 162
- 정보용 160
- 조회 160

시스템 메세지 (계속)

- 추가 메세지 정보 화면 161

시스템 상태

- 작업 대상 146

시스템 상태에 대한 작업(WRKSYSSTS) 명령 146

시스템 서비스 툴(SST) 181

시스템 성능

- 어플리케이션 리퀘스터 문제점 156

시스템 저장(SAVSYS) 명령 137

시작

- 디버그 187
- 서비스 작업 187
- 저널 액세스 경로 131
- 확약 제어 133
- Pass-Through 159

실패 보고를 위한 자료 수집 178

실패 자료 178

[아]

암호

- 대화식 SQL 205
- 송신 205
- 암호화 43, 46, 49, 59, 166
- CONNECT 명령문 205

암호화

- 자료 스트림 183
- Diffie-Hellman 59

액세스 경로

- 보호, 시스템 관리식 132

액세스 계획

- 정의 217
- SQL 패키지 217

어플리케이션

- 고려사항 18
- 분산 관계형 데이터베이스(DRDB) 쓰기 193

설계 18

요구사항 18

어플리케이션 리퀘스터

- 관계형 데이터베이스 디렉토리 75
- 문제점 진단 155
- 정의 3

프로그램 요구 실패 메세지 165

DDM 작업의 확약 제어 212

어플리케이션 리퀘스터 드라이버(ARD) 프로그램 8

- 어플리케이션 바인딩 217
- 어플리케이션 서버 156
 - 관계형 데이터베이스 디렉토리 75
 - 문제점 진단 156
 - 서비스 작업 시작 187
 - 정의 3
 - 프로그램 요구 실패 메시지 165
 - DDM 작업의 확약 제어 212
- 어플리케이션 서버(AS)
 - 리포트 명령 제출 106
- 어플리케이션 프로그래밍 예 223
- 어플리케이션 프로그램
 - 문제점 처리
 - SQLCODE 168
 - SQLSTATE 168
 - 바인딩 217
 - 사전컴파일 214
 - 사전컴파일러 명령 215
 - 시스템 명령 규칙 195
 - 임시 소스 파일 멤버 215
 - 컴파일 216
 - 테스트와 디버깅 218
 - 프로그램 참조 219
 - 호스트 변수 213
 - 호스트 프로그램 213
 - SQL 명령 규칙 195
 - SQL 패키지 삭제 221
 - SQL 패키지 작성 220
 - SQLCODE 171
 - SQLSTATE 171
- 연결
 - SQL 198
 - SQL 대 네트워크 107
- 연결 관리 18
- 연결 문제점 165
- 연결 상태 199
 - 리포트 작업 단위 196
 - 분산 작업 단위 198
 - 활성 그룹 199
 - CONNECT(유형 2) 명령문 198
- 연결 실패 166
- 연결되지 않은 상태 199
- 예 81
 - 어플리케이션 프로그래밍 223
 - 프로그래밍
 - 데이터베이스 설정 224
 - 자료 입력 225
 - COBOL/400 언어 239

- 예 (계속)
 - 프로그래밍 (계속)
 - C/400 언어 246
 - RPG/400 언어 230
 - 프로그램 참조 표시 110
 - APPN 구성
 - 교환 회선 설명 33, 37
 - 네트워크 노드에서 네트워크 노드 30
 - 네트워크 속성 32, 34, 35
 - 비교환 회선 설명 32, 36
 - 제어기 설명, 교환 33, 37
 - 제어기 설명, 비교환 32, 36
 - FFDC 덤프 274
 - RW 추적 자료 분석 268
 - spiffy사 11
 - SQL 패키지 참조 표시 111
- 예상과 요구
 - 식별 15
- 오류 기록부 179
 - FFDC 자료 187
- 오류 메시지
 - DRDA 연결 권한 부여 실패 166
- 오류 보고서
 - 오류 기록부 인쇄 179
 - 작업 기록부 인쇄 178
 - 정의 273
 - 첫 번째 실패 자료 캡처 273
 - 추적 작업 180
 - 통신 추적 181
- 오류 회복
 - 관계형 데이터베이스 127
- 오브젝트
 - 복원 137, 138, 140
 - 분산 관계형 데이터베이스명명 195
 - 저장 132, 137, 138
- 오브젝트 감사 값
 - 변경 127
- 요구와 예상
 - 식별 15
 - 요구와 예상 식별 15
 - 용어 및 개념 5
 - 위치, 정의 19
- 유사 환경
 - 정의 6
- 이력 기록부, 표시 123
- 이중복사 보호 129
- 일괄처리 작업 70
- 일반 바인드 옵션 260

- 임시 소스 파일 멤버 215

[자]

- 자국어 지원 209
- 자동시작 작업 70
- 자료
 - 고려사항 20
 - 문자 변환 7
 - 설계 18
 - 성능 향상을 위한 블록 258
 - 실패 178
 - 요구사항 20
 - DB2 Connect를 통해 액세스 256
- 자료 가용성 및 보호 127
- 자료 로드
 - 표 안에 86
 - DB2 for iSeries 조회 관리 87
 - DFU(자료 파일 유틸리티) 사용 88
 - SQL 사용 86
- 자료 변환
 - 비문자 자료 212
- 자료 요구
 - 판별 15
- 자료 위치
 - 결정 147
- 자료 이동
 - 다른 서버 간에
 - 테이프나 디스켓 사용 94
 - 통신 사용 94
 - 파일 전송 프로토콜 사용 96
 - OSI File Services/400 사용권 프로그램 램 사용 96
 - SQL 기능 사용 94
 - TCP/IP Connectivity Utilities/400 사용권 프로그램 사용 96
 - 대화식 SQL 사용 89
 - 저장 및 복원 사용 93
 - 파일 복사 명령 사용 91
 - DB2 UDB for iSeries 조회 관리 90
 - DDM으로 파일 복사 91
 - iSeries 서버간 88
- 자료 중복 142
- 자료 캡처
 - FFDC 186
- 자료 항목
 - 해석 267

- 자료 해석
 - 비문자 자료 212
 - CCSID 209
- 자원
 - 재생 108
- 작성
 - SQL 패키지 220
- 작업
 - 계정 112
 - 변경 108
 - 유형 70
 - 작업 대상 98
 - 활동에 대한 작업 147
- 작업 관리
 - 서브시스템 70
 - 서브시스템 설정 71
 - 작업 유형 70
- 작업 기록부
 - 경고 178
 - 작업 찾기 103
 - 표시 102
- 작업 단위
 - 정의 3
- 작업 대상
 - 관계형 데이터베이스 디렉토리 항목 79
 - 구성 상태 29, 142
 - 디스크 상태 146
 - 사용자 작업 99
 - 시스템 상태 146
 - 작업 98
 - 활동 작업 100, 146, 147
- 작업 추적 180
- 작업, 일반
 - 계획 20
- 재생
 - 분산 자료 관리 대화 108
 - 자원 108
- 저널
 - 표시 23, 131
- 저널 관리
 - 개요 130
 - 색인 131
 - 색인 저널링 시작 131
 - 저널 리시버 130
 - 중단 130
 - 확약 제어 130
- 저널 리시버 130

- 저널 액세스 경로
 - 시작 131
- 저널링
 - iSeries 파일 257
- 저장
 - 관계형 데이터베이스 디렉토리 138
 - 라이브러리 132, 136
 - 변경된 오브젝트 136, 137
 - 보안 자료 137
 - 색인 137
 - 시스템 137
 - 오브젝트 132, 136, 137, 138
 - 저널 리시버 131
 - 저장 파일 자료 136, 137
 - 저장 파일로 136
 - 테이프나 디스켓으로 136
 - SQL 패키지 138
- 저장 파일 136
- 저장 파일 자료
 - 저장 137
- 저장된 프로시저 20, 117, 150
 - 사용 148
 - 정의 206
- 전문용어 114
- 정렬 순서
 - 정의 261
- 정보용 메세지 160
- 정의
 - 네트워크 인터페이스 설명 28
 - 제어기 설명 29
 - 회선 설명 28
- 정의된 요약 정의 101
- 정지 연결 상태 199
- 제거
 - 관계형 데이터베이스(RDB) 디렉토리 항목 79
- 제어 서브시스템
 - 정의 70
 - QBASE 70
 - QCTL 71
- 제출
 - 리모트 명령 105, 106, 111
- 조작 및 관리 97
- 조회 관리와 대화식 SQL 설정 256
- 조회 블록 크기
 - 영향을 주는 요소 150
- 조회 블록 크기에 영향을 주는 요소 150

- 조회 자료
 - 성능 향상을 위한 블록 258
- 주
 - 공동-플랫폼 DRDB 253
- 주 프레임으로부터 자료의 마이그레이션 91, 95
- 주의사항 301
- 중복
 - 자료 142
 - 통신 네트워크 141
- 지원 제품
 - DB2 6
 - DB2 Connect 6
 - DB2 for VSE 및 VM 6
- 진단 옵션 191

[차]

- 첫 번째 실패 자료 캡처 자료(FFDC)
 - 해석 267
- 첫실패 자료 캡처 자료 검색 186
- 체크섬 보호 129
- 추가
 - 관계형 데이터베이스(RDB) 디렉토리 항목 76, 138, 188
- 추적
 - 작업 180
 - 통신 181
- 추적 시작
 - STRTRC 180
- 추적 자료
 - 분석 268
- 추적 작업
 - TRCJOB 180
- 추적 작업 자료
 - 해석 267
- 추적점
 - 설명 269
 - 부분 송신 자료 스트림 270
 - 성공적이지 않은 폐치 270
 - 성공적인 폐치 270
 - 송신 자료 스트림 270
 - 수신 자료 스트림 269
 - 아웃바운드 LOB 표에 저장된 271
 - 인바운드 LOB 표에 저장되는 271
 - LOB 표로부터의 자료 스트림에서 빌드 되는 271

[카]

카탈로그

- 정의 1
- 코드 페이지 8
- 코드화 문자 세트 ID(CCSID) 7
 - 개요 209
 - 변경 210
 - 사용자 프로파일 210
 - 자료 변환 방법 211
 - 태그 209, 211
 - 허용된 값 209
- DB2 UDB Server for VM 고려사항 255
- DB2 고려사항 255
- 코드화 문자 세트 ID(CCSID) 고려사항 253
- 코드화, 문자 변환 7
- 컬렉션
 - 정의 1
 - SQL 명명 규칙 195
 - SQLCA(SQL communication area) 171
- 컬렉션 제거 111

[타]

테스트와 디버깅

- 어플리케이션 프로그램 218

통신

- 개요 25
 - 구성
 - 네트워크 인터페이스 설명 28
 - 단계 28
 - 연결변환 또는 연결변환 중지 29
 - 위치 리스트 29
 - 제어기 설명 29
 - 회선 설명 28
 - 네트워크 고려사항
 - DRDA 지원 26
 - 작업 70, 71
 - APPC 지원 25
 - APPN 지원 25
 - DDM 대화 107
 - DDM과 DRDA의 공존 26, 85
- ### 통신 구성
- 네트워크 속성 28
- ### 통신 추적
- 메세지 181
 - 시스템 서비스 툴(SST) 181

통신 툴 25

툴

- 통신 25
- 트랜잭션 프로그래밍 매개변수
 - iSeries 서버(TNSPGM) 77
 - SNA에서(TPN) 77

[파]

파일

- 저널링 257
- 패키지
 - 작업 대상 220
- 패키지 관리
 - SQL 221

표

- 정의 1
- 표시
 - 관계형 데이터베이스(RDB) 디렉토리 항목 79, 138
 - 메세지 설명 161
 - 오브젝트 108
 - 작업 기록부 102
 - 저널 23, 131
 - 프로그램 참조 108, 219

표시장치 복사 159

표시, 복사 159

프로그래밍 고려사항

- 분산 관계형 데이터베이스(DRDB) 어플리 케이션 194

프로그래밍 예

- 어플리케이션 223
- 프로그램 요구 실패 165

프로그램 참조

- 표시 108, 219

프로그램 참조 표시(DSPPGMREF) 명령 108, 219

프로그램 컴파일 216

[하]

해석

- 오류 기록부로부터의 FFDC 자료 187
- 자료 항목
 - 추적 작업의 RW 구성요소 267
- 추적 작업 267
- FFDC 자료 267
- 해제 연결 상태 199

현재 연결 상태 199

호스트 변수

- 정의 213
- 호스트 프로그램
 - 정의 213
- 화면 복사 시작(STRPCYSCRN) 명령 159
- 확약 제어
 - 개요 3
 - 레코드 잠금 지속 134
 - 분산 관계형 데이터베이스(RDB)와 DDM 작업 212
 - 시작 133
 - 잠금 레벨 133
 - 저널 관리 130
 - 트랜잭션 회복 133
 - NFYOBJ(통지 오브젝트) 매개변수 133

확약된 작업

- 정의 3
- 확장 프로그램간 통신(APPC) DRDA 지원 25

환경

- 상이한 6
- 유사 6
- 활동 작업
 - 작업 대상 100, 146, 147
- 활동 작업에 대한 작업(WRKACTJOB) 명령 100, 146

회복

- 계획 24
- 디스크 장애 128
- 메소드 127
- 무정전 전원 장치(uninterruptible power supply) 128
- 보조 기억장치 풀 129
- 이중복사 보호 129
- 장애 유형 127
- 저널 관리 130
- 체크섬 보호 129

A

ADDRDBDIRE(Add Relational Database

- Directory Entry) 명령 76, 124, 138

ADDRDBDIRE(관계형 데이터베이스 디렉토리

- 항목 추가) 명령 188

ADT(어플리케이션 개발 툴) 88

APPC 서버 작업 디버깅을 위한 특별한

- TPN 188

APPC(확장 프로그램간 통신)
 DRDA 지원 25

APPN 위치 리스트 29

APPN(Advanced Peer-to-Peer Networking)
 구성 예 30
 리모트 위치 리스트 29
 DRDA 지원 25

ARD(어플리케이션 리퀘스터 드라이버) 프로그
 램 8

ASP 그룹
 사용 19
 정의 60

ASP(보조 기억장치 풀) 129

C

CCSID
 변환 고려사항 253, 255
 DB2 255
 DB2 Connect 사용권 프로그램 253
 DB2 Server for VM 데이터베이스 관리자
 255

CCSID(코드화 문자 세트 ID) 7
 개요 209
 변경 210
 사용자 프로파일 210
 자료 변환 방법 211
 태그 209, 211
 허용된 값 209
 DB2 UDB Server for VM 고려사항
 255
 DB2 고려사항 255

CDRA(문자 자료 표시 구조) 7, 209

CHGDMMTCPA 명령 84

CHGJOB(작업 변경) 명령 108

CHGNETA(네트워크 속성 변경) 명령 28

CHGOBJAUD(오브젝트 감사 값 변경) 명령
 127

CHGRDBDIRE(Change Relational Database
 Directory Entry) 명령 80, 124

CHGRDBDIRE(관계형 데이터베이스 디렉토리
 항목 변경) 명령 188

CHGSBSD(서브시스템 설명 변경) 명령 71

COBOL/400
 프로그래밍
 예 239

CP13E34
 QRWOPTIONS 참조 53

CRTSQCBLLI(SQL COBOL ILE 작성) 명령
 216

CRTSQCBLLI(SQL COBOL 작성) 명령
 216

CRTSQCLCI(SQL C ILE 작성) 명령 216

CRTSQCLFTN(SQL FORTRAN 작성) 명령
 216

CRTSQCLPKG(SQL 패키지 작성) 명령 220

CRTSQCLPLI(SQL PL/I 작성) 명령 216

CRTSQCLRPGI(SQL RPG ILE 작성) 명령
 216

CRTSQCLRPG(SQL RPG 작성) 명령 216

C/400
 프로그래밍
 예 246

D

DB2 6
 변환 고려사항 255
 CCSID 255
 DB2 Connect 6
 iSeries 서버 자료 액세스 256
 DB2 Connect 사용권 프로그램
 변환 고려사항 253
 CCSID 253
 DB2 Connect 사용자의 FAQ 256
 DB2 Connect를 통해 iSeries 자료에 액세스
 256
 DB2 for iSeries 조회 관리 기능
 자료를 표 안에 로딩 87
 DB2 for VSE 및 VM 6
 DB2 Server for VM 데이터베이스 관리자
 변환 고려사항 255
 CCSID 255
 DB2 UDB for iSeries 조회 관리 기능
 iSeries 서버간 자료 이동 90
 DB2 UDB 조회 관리자 및 SQL 개발 킷
 분산 관계형 데이터베이스(RDB) 명령문
 205
 DRDA 플랫폼에서의 공존 208
 DDM 작업 시작 메시지 103
 DDM 파일
 설정 85
 SQL 확약 제어 212
 DDM 파일 액세스 85
 DDMCNV(DDM 대화) 작업 속성 107, 108

DDM(분산 자료 관리)
 대화 보유 107
 대화 제거 107, 108
 대화를 활동 중으로 유지 107
 사용되지 않는 대화 107
 재생
 대화 108
 자원 108
 제거된 대화 107
 파일 복사 명령 사용 91
 CHGJOB 명령 108
 DDMCNV 작업 속성 107, 108
 DRDA 지원과 공존 26
 iSeries 서버간 자료 이동 91

DFU(자료 파일 유틸리티) 88

Diffie-Hellman
 암호화 59

DISCONNECT 107

DRDA 연결 권한 부여 실패 166

DRDA(분산 관계형 데이터베이스 구조) 레벨
 2 지원 4
 DRDA(분산 관계형 데이터베이스 구조) 지원
 개요 6
 현재 iSeries 지원 9
 CDRA 사용 7
 DDM과 공존 26

DRDB
 공통-플랫폼 253

DROP PACKAGE 명령문 221

DSPJOBLOG(작업 기록부 표시) 명령 102

DSPJRN(저널 표시) 명령 23, 131

DSPMSGD(Display Message Descriptions)
 명령 161

DSPPGMREF(프로그램 참조 표시) 명령
 108, 219

DSPRDBDIRE(관계형 데이터베이스 디렉토리
 항목 표시) 명령 79, 124, 138

DUW(분산 작업 단위)
 정의 4

E

EBCDIC 8

F

FFDC 덤프 274

FFDC 자료
해석 187
FFDC(First-Failure Data Capture)
해석 267
FFDC에서의 DDM 오류 코드 282
FFDC(첫실패 자료 캡처) 186
덤프 출력 설명 277
DDM 오류 코드 282

I

IBM이 공급하는 서비스시스템
QBASE 70
QBATCH 71
QCMN 71
QCTL 71
QINTER 71
QSPL 71
QSYSWRK 71
iSeries QCCSID 값 254
iSeries 분산 관계형 데이터베이스(DRDB)
관리 10
iSeries 분산 관계형 데이터베이스(DRDB) 관
리 10
iSeries 서버 자료
DB2 Connect를 통해 액세스 256
iSeries 파일
저널링 257

K

Kerberos 59
소스 구성 53
인증 51, 84
정의명 54

L

LCKLVL 매개변수 133

N

NFYOBJ(통지 오브젝트) 매개변수 133

P

Pass-Through
시작 159

Pass-Through 시작(STRPASTHR) 명령 159

Q

QADBXRDBD 124
QBASE 제어 서비스시스템 70
QCCSID
시스템 값 254
QCNTSRVC 188, 189
QCTL 제어 서비스시스템 71
QCTLSBSD 시스템 값 71
QPSRVDMP FFDC 스펴 파일 274
QRWOPTIONS
자료 사용법 190

R

RCLDDMCNV(분산 자료 관리 대화 재생) 명
령 108
RCLRSC(자원 재생) 명령 108
RDB(관계형 데이터베이스) 매개변수
관계형 데이터베이스 디렉토리에서 76
내재 CONNECT 201
RELEASE 107
RMVRDBDIRE(관계형 데이터베이스 디렉토리
항목 제거) 명령 79, 124
RPG/400
프로그래밍
예 230
RSTAUT(권한 복원) 명령 137
RSTCFG(구성 복원) 명령 137
RSTLIB(라이브러리 복원) 명령 137
RSTOBJ(오브젝트 복원) 명령 137, 138, 140
iSeries 서버간 자료 이동 93
RSTUSRPRF(사용자 프로파일 복원) 명령
137
RUNRMTCMD 명령 85
RUW(리모트 작업 단위)
정의 4
RW 추적 자료
분석 268

S

SAVCHGOBJ(변경된 오브젝트 저장) 명령
137
SAVLIB(라이브러리 저장) 명령 132

SAVOBJ(오브젝트 저장) 명령 132, 137,
138
iSeries 서버간 자료 이동 93
SAVSAVFDTA(저장 파일 자료 저장) 명령
137
SAVSECDTA(보안 자료 저장) 명령 137
SAVSYS(시스템 저장) 명령 137
SBMRMTCMD 명령 85
SBMRMTCMD(리모트 명령 제출) 명령 105,
106, 111
SMAPP(시스템 관리식 액세스 경로 보
호) 132
spiffy사의 예 11
SQL CALL 206
SQL 명령문
사전컴파일 214
CALL 206
CONNECT
내재 201
명시 203
DISCONNECT 107
DROP PACKAGE 221
RELEASE 107
SQL 명명 규칙 195
SQL 용어
대응하는 시스템 용어 1
정의 리스트 1
SQL 컬렉션
정의 1
SQL 패키지
대화식 SQL에 대한 85
복원 138
사용되는 오브젝트 표시 111
사전컴파일의 결과로 작성 215
삭제 221
액세스 계획 217
작성 220
작업 대상 220
저장 138
정의 193
CRTSQLPKG로 작성 220
CRTSQLxxx로 작성 220
SQL 패키지 관리 221
SQL 패키지 작성(CRTSQLPKG) 명령 220
SQL 패키지에 대한 작업 220
SQL 프로그램
문제점 처리
SQLCODE 168

SQL 프로그램 (계속)
 문제점 처리 (계속)
 SQLSTATE 168
 사용되는 오브젝트 표시 110
 예 리스트
 사전컴파일러 168
 CRTSQLPKG 170
 SQLCODE 168
 SQLSTATE 168
 컴파일 216
 확약 제어 시작 133
 SQL 프로그램 종료 208
 SQL 프로그램, 종료 208
 SQLCODE
 SQLCODEs 및 SQLSTATE, 리스트
 171
 SQLCODE 오류 코드
 분산 관계형 데이터베이스(RDB) 171
 오류 처리 171
 SQLSTATE
 SQLCODEs 및 SQLSTATE, 리스트
 171
 SQLSTATE 오류 코드
 분산 관계형 데이터베이스(RDB) 172
 오류 처리 171
 SST(System Service Tools) 181
 STRCMTCTL(확약 제어 시작) 명령 133
 STRCPYSCRN(화면 복사 시작) 명령 159
 STRDBG(Start Debug) 명령 187
 STRJRNAP(저널 액세스 경로 시작) 명령
 131
 STRPASTHR(Start Pass-Through) 명령 159
 STRSRVJOB(서비스 작업 시작) 명령 187

T

TCP/IP 18
 보안 84
 서버 작업 기록부 검색 178
 서버 작업 찾기 103
 서버 작업(jobs)에 대한 작업 100
 서비스 작업 190
 작업 기록부 검색 179
 작업 기록부 저장 179
 TCP/IP 서버 사전시작 작업의 작업 기록부 검색 178
 TCP/IP 서버 시작 CL 명령 116
 TCP/IP 서버 CL 명령 116

TCP/IP 통신 지원 개념 114
 TCP/IP 통신, 설치 중 115
 TPN
 QCNTSRVC 설정 188, 189
 TPN으로서 QCNTSRVC 설정
 DB2 연결 어플리케이션 리퀘스터 189
 iSeries용 DB2 UDB 어플리케이션 리퀘스터 188
 VM 어플리케이션 리퀘스터를 위한 DB2 188
 z/OS 어플리케이션 리퀘스터를 위한 DB2 UDB 상 189
 TRCTCPAPP
 기능 181
 명령 49
 추적 183

V

VRYCFG(구성 변환) 명령 29, 142

W

WRKACTJOB(Work with Active Jobs) 명령
 100, 146
 WRKCFGSIS(구성 상태에 대한 작업) 명령
 29, 142
 WRKCFGSTS(구성 상태에 대한 작업) 명령
 29, 142
 WRKDSKSTS(Work with Disk Status) 명령
 146
 WRKJOB(작업(Job)에 대한 작업) 명령 98
 WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 명령 79, 124
 WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 명령 79, 124
 WRKSYSSTS(Work with System Status) 명령
 146
 WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 명령 99
 WRKUSRJOB(사용자 작업(Jobs)에 대한 작업) 명령 99

IBM