



IBM 시스템 - iSeries

데이터베이스

분산 데이터베이스 프로그래밍

버전 5 릴리스 4





IBM 시스템 - iSeries

데이터베이스

분산 데이터베이스 프로그래밍

버전 5 릴리스 4

주!

이 정보와 이 정보가 지원하는 제품을 사용하기 전에, 287 페이지의 『주의사항』의 정보를 읽으십시오.

제 7 판(2006년 2월)

이 개정판은 새 개정판에 별도로 명시하지 않는 한, IBM i5/OS(제품 번호 5722-SS1)의 버전 5, 릴리스 4, 수정 0 및 모든 후속 릴리스와 수정에 적용됩니다. 이 버전은 모든 축약 명령어 세트 컴퓨터(RISC) 모델 및 CISC 모델에서도 실행되지 않습니다.

© Copyright International Business Machines Corporation 1998, 2006. All rights reserved.

목차

분산 데이터베이스 프로그래밍	1	DRDA용 작업 관리 환경 설정	28
V5R4의 새로운 사항	1	APPC 서브시스템	29
인쇄 가능한 PDF.	2	TCP/IP 서브시스템.	31
분산 데이터베이스 프로그래밍 소개	2	독립형 보조 기억장치 풀의 사용자 데이터베이스	31
분산 관계형 데이터베이스 처리	3	관계형 데이터베이스 디렉토리 사용	32
리모트 작업 단위.	6	관계형 데이터베이스 디렉토리에 대한 작업	33
분산 작업 단위	6	SNA에 대한 항목 추가 사용법.	33
기타 분산 관계형 데이터베이스 용어 및 개념	7	TCP/IP에 대한 항목 추가 사용법.	35
분산 관계형 데이터베이스 구조 지원	8	관계형 데이터베이스 별명 지정	36
DRDA 및 CDRA 지원	9	어플리케이션 리퀘스터 드라이버에 대한 항목 추가.	37
어플리케이션 리퀘스터 드라이버 프로그램	10	WRKRDBDIRE 명령 사용	37
i5/OS의 분산 관계형 데이터베이스	11	*LOCAL 디렉토리 항목.	38
iSeries 분산 관계형 데이터베이스 관리	12	독립형 보조 기억장치 풀의 사용자 데이터베이스에 대한 디렉토리 항목	39
예: Spiffy사 분산 관계형 데이터베이스	14	예: 관계형 데이터베이스 디렉토리 설정	39
Spiffy 조직 및 시스템 프로파일	14	DRDA 보안 설정	42
Spiffy사의 자동차 서비스 업무 프로세스	16	DRDA에 대한 TCP/IP 서버 설정.	43
Spiffy사에 대한 분산 관계형 데이터베이스 관리	16	대화식 SQL에 대한 SQL 패키지 설정	44
계획 및 설계	17	DDM 파일 설정	44
분산 관계형 데이터베이스에 대한 요구 및 예상 식별.	18	분산 관계형 데이터베이스의 표에 자료 로드	45
분산 관계형 데이터베이스에 대한 자료 요구	18	분산 관계형 데이터베이스의 표에 새 자료 로드	45
분산 관계형 데이터베이스 기능.	18	SQL을 사용하여 표에 자료 로드	45
분산 관계형 데이터베이스에 대한 목표 및 방향	18	iSeries 조회 관리 기능을 사용하여 표 및 파일의 자료 조작	46
분산 관계형 데이터베이스에 대한 어플리케이션, 네트워크 및 자료의 설계	20	자료 파일 유틸리티를 사용하여 자료 입력, 표 갱신 및 조회 작성	47
추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계	20	한 iSeries 서버에서 다른 iSeries 서버로 자료 이동.	47
분산 관계형 데이터베이스에 대한 네트워크 고려사항	21	사용자 작성 어플리케이션 프로그램 작성	48
분산 관계형 데이터베이스에 대한 자료 고려사항	22	대화식 SQL을 사용하여 데이터베이스 조회	48
분산 관계형 데이터베이스에 대한 관리 전략 개발	22	DB2 Universal Database for iSeries 조회 관리 기능을 사용하여 리모트 서버 조회	50
분산 관계형 데이터베이스에 대한 일반 작업	22	테이프 또는 테이프에서 파일 복사.	50
분산 관계형 데이터베이스에 대한 보안 고려사항	24	파일 복사 명령을 사용하여 iSeries 서버 간에 자료 이동.	51
분산 관계형 데이터베이스에 대한 계정	25	네트워크 파일 명령을 사용하여 네트워크를 통해 자료 전송	52
분산 관계형 데이터베이스에 대한 문제점 분석	26	서버 저장 및 복원 명령을 사용하여 표 이동	53
분산 관계형 데이터베이스의 백업 및 회복	26		
초기 설정	26		
iSeries 서버에서 작업 관리	27		

비iSeries 서버에서 iSeries 서버로 데이터베이스 이동.	54
다른 IBM 서버에서 자료 이동.	54
비IBM 서버에서 자료 이동.	56
보안.	56
분산 관계형 데이터베이스 보안의 요소.	57
APPC 네트워크의 보안 요소.	59
APPN 구성 리스트.	60
대화 레벨 보안.	61
APPC 네트워크의 DRDA 어플리케이션 서버 보안.	62
TCP/IP 네트워크의 보안 요소.	65
TCP/IP 네트워크의 어플리케이션 리퀘스터 보안.	65
TCP/IP 네트워크에서 어플리케이션 서버 보안.	68
DDM 및 DRDA에 대한 연결 보안 프로토콜.	69
DDM 및 DRDA에 대한 보안 소켓 층.	69
DDM 및 DRDA에 대한 인터넷 프로토콜 보안 구조.	70
clear text로 송신되는 특정 암호에 대한 고려사항.	71
DDM/DRDA에 대한 포트 및 포트 제한사항.	71
DRDA 서버 액세스 제어 종료 프로그램.	72
예: DRDA 서버 액세스 제어 종료 프로그램.	74
DRDA에 대한 오브젝트 관련 보안.	76
분산 관계형 데이터베이스 오브젝트에 대한 권한.	77
분산 관계형 데이터베이스에 대한 허용된 권한 하에서 실행되는 프로그램.	78
분산 관계형 데이터베이스의 보호 전략.	79
어플리케이션 개발.	81
분산 관계형 데이터베이스 어플리케이션에 대한 프로그래밍 고려사항.	82
분산 관계형 데이터베이스 오브젝트 명명.	82
시스템(*SYS) 명명 규칙.	82
SQL(*SQL) 명명 규칙.	83
디폴트 콜렉션명.	83
분산 관계형 데이터베이스로 연결.	84
리모트 작업 단위.	84
분산 작업 단위.	86
디폴트 활성 그룹에 대한 내재적 연결 관리.	89
디폴트가 아닌 활성 그룹에 대한 내재적 연결 관리.	90
명시적 연결 관리.	91

분산 관계형 데이터베이스 특정 SQL 및 SQL CALL.	94
분산 관계형 데이터베이스 명령문.	94
SQL CALL문(저장 프로시저어).	95
iSeries용 DB2 Universal Database 공존.	96
DRDA 작업 단위 종료.	97
저장 프로시저어, 사용자 정의 기능 및 확약 제어.	97
코드화 문자 세트 ID.	98
iSeries 서버 지원.	99
기타 DRDA 자료 변환.	101
DDM 파일 및 SQL.	102
분산 관계형 데이터베이스 프로그램 준비.	103
SQL문이 있는 프로그램 컴파일.	103
리스팅.	104
임시 소스 파일 멤버.	104
SQL 패키지 작성.	105
사전컴파일러 명령.	105
어플리케이션 프로그램 컴파일.	106
어플리케이션 바인드.	107
테스트 및 디버깅.	108
프로그램 참조.	110
SQL 패키지에 대한 작업.	111
CRTSQLPKG(SQL 패키지 작성) 명령 사용.	111
SQL 패키지 관리.	112
DLTSQLPKG(SQL 패키지 삭제) 명령을 사용하여 SQL 패키지 삭제.	112
패키지 권한 수정.	113
SQL DROP PACKAGE문 사용.	113
관리.	113
관계형 데이터베이스 활동 모니터.	114
분산 관계형 데이터베이스의 작업에 대한 작업.	114
분산 관계형 데이터베이스의 사용자 작업에 대한 작업.	115
분산 관계형 데이터베이스의 활동 작업에 대한 작업.	116
분산 관계형 데이터베이스의 확약 정의에 대한 작업.	118
분산 관계형 데이터베이스의 작업 기록부를 사용하여 요구 정보 추적.	119
분산 관계형 데이터베이스 작업 찾기.	120
리모트 iSeries 서버 조작.	122
DDM 대화 제어.	124
DDM 자원 재생.	126
프로그램이 사용하는 오브젝트 표시.	127

예: 프로그램 참조 표시	128	관계형 데이터베이스 디렉토리 저장 및 복 원	159
분산 관계형 데이터베이스에서 콜렉션 삭제	129	분산 관계형 데이터베이스에 대한 네트워크 중복 고려사항	162
분산 관계형 데이터베이스의 작업 사용 통계	131	분산 관계형 데이터베이스 네트워크에서 자료 중 복	164
TCP/IP 서버 관리.	132	성능	165
DRDA TCP/IP 서버 전문 용어	133	네트워크를 통해 분산 관계형 데이터베이스 성능 향상	166
DDM에 대한 TCP/IP 통신 지원 개념	134	서버를 통해 분산 관계형 데이터베이스 성능 향 상	167
TCP/IP를 통한 DRDA 또는 DDM 연결 설정	134	데이터베이스를 통해 분산 관계형 데이터베이스 성능 향상	168
DRDA 및 DDM 리스너 프로그램	135	DRDA 자료 위치 결정	168
STRTCPSVR(TCP/IP 서버 시작) CL 명 령	135	DRDA에 대한 블로킹에 영향을 미치는 요소 iSeries용 DB2 Universal Database 대 iSeries용 DB2 Universal Database 블로 킹	169
ENDTCPSVR(TCP/IP 서버 종료) CL 명 령	136	iSeries용 DB2 Universal Database 대 비 iSeries용 DB2 Universal Database 블로 킹	169
iSeries Navigator에서 DDM 리스너 시작	136	비iSeries용 DB2 Universal Database 대 iSeries용 DB2 Universal Database 블로 킹	169
DRDA/DDM 서버 작업	137	DRDA 블로킹 규칙 요약	170
DDM과 서브시스템 설명 및 사전시작 작 업 항목	137	DRDA 조희 블록의 크기에 영향을 미치는 요소	171
DRDA 및 DDM 사전시작 작업.	137	문제 해결	172
DDM 서버 작업 서브시스템 구성	141	iSeries 문제점 처리 개요	172
서버 작업 식별.	143	분산 관계형 데이터베이스 문제점 분리.	174
iSeries 작업명	143	DRDA 잘못된 출력 문제점	174
서버 작업 표시.	143	어플리케이션이 예상 시간 내에 완료되지 않 음	175
이력 기록부 표시	144	분산 관계형 데이터베이스 사용자에게 대한 작업 화면 복사	178
관계형 데이터베이스 디렉토리 감사	145	메세지.	179
자료 가용성 및 보호.	146	메세지 유형.	181
분산 관계형 데이터베이스에 대한 회복 지원	147	분산 관계형 데이터베이스 메세지.	182
분산 관계형 데이터베이스에 대한 디스크 장 에 후 자료 회복	148	APPC에 대한 프로그램 시작 요구 실패 처리	185
보조 기억장치 풀	148	TCP/IP에 대한 핸들 연결 요구 실패	186
분산 관계형 데이터베이스의 체크섬 보호	148	서버가 시작되지 않았거나 포트 ID가 유효 하지 않음	186
분산 관계형 데이터베이스에 대한 이중복사 보호	149	DRDA 연결 권한 실패	187
분산 관계형 데이터베이스에 대한 저널 관리	149	서버를 사용할 수 없음	188
색인 회복	150	대화식 SQL 특정 연결 실패	188
색인 리빌드 시간을 줄이도록 표 설계	151	서버에서 사전시작 작업이 충분하지 않음	188
시스템 관리 액세스 경로 보호.	152		
확약 제어를 통해 트랜잭션 회복.	152		
분산 관계형 데이터베이스에 대한 저장 및 복 원 처리	157		
분산 관계형 데이터베이스 환경에서 색인 저장 및 복원	158		
분산 관계형 데이터베이스 환경에 보안 정 보 저장 및 복원	159		
분산 관계형 데이터베이스 환경에서 SQL 패키지 저장 및 복원.	159		

어플리케이션 문제점	189	예: RPG 프로그램	220
리스팅	189	예: COBOL 프로그램	229
사전컴파일러 리스팅	189	예: 삽입된 SQL을 사용하는 C 프로그램	237
CRTSQLPKG 리스팅	191	예: Java 프로그램	243
SQLCODE 및 SQLSTATE	192	예: 프로그램 출력	248
분산 관계형 데이터베이스 SQLCODE 및 SQLSTATE	193	사용자 FAQ	249
시스템 및 통신 문제점	196	분산 관계형 데이터베이스로 연결	249
실패 보고를 위한 자료 수집	198	iSeries 시스템 값 QCCSID	250
작업 기록부 인쇄	198	DB2 Universal Database for z/OS 및 DB2 Universal Database Server for VM 데이터베 이스 관리자에 대한 CCSID(코드화 문자 세트 ID) 변환 고려사항	251
TCP/IP 서버 사전시작 작업에서 작업 기록부 찾기	199	Linux, UNIX 및 Windows용 DB2 Universal Database에서 연결하려고 시도할 때 왜 SQL5048N 메시지가 표시됩니까?	251
제품 활동 기록부 인쇄	200	iSeries 파일을 저널링해야 합니까?	252
작업 추적	200	조회 자료는 성능 향상을 위해 언제 블로킹됩니 까?	252
추적 작업	201	DBM SQL0969N 오류 메시지에 보고된 SQLCODE 및 연관된 토큰을 어떻게 해석합니 까?	253
추적 시작	201	WHERE절의 호스트 변수 유형은 성능에 어떤 영향을 미칠 수 있습니까?	254
통신 추적	202	규정되지 않은 표 및 뷰 이름을 분석하기 위해 라이브러리 리스트를 사용할 수 있습니까?	255
표준 통신 추적	203	Linux, UNIX 및 Windows용 DB2 Universal Database 사용자가 일반 EBCDIC 순서 대신 iSeries 서버의 DRDA 작업 NLSS 정렬 순서표 가 사용되도록 지정할 수 있습니까?	256
TCP/IP 통신 추적	205	조회를 실행할 때 행이 리턴되지 않는 이유는 무 엇입니까?	256
TCP/IP 통신 추적 포매팅	205	iSeries용 DB2 Universal Database와 대화하려 면 어떤 레벨의 Linux, UNIX, Windows용 DB2 Universal Database가 필요합니까?	256
첫실패 자료 캡처 자료	208	Linux, Unix 및 Windows용 DB2 Universal Database 버전 8에서 iSeries까지 사용 가능한 화면 이동 커서 지원을 얻는 방법은 무엇입니까? 257	257
어플리케이션 서버 문제점을 진단하기 위한 서비 스 작업 시작	209	상이한 환경에서 상호조작에 대한 기타 추가 정 보	257
APPC 서버에 대한 서비스 작업	209	추적 작업 및 FFDC 자료 해석	261
사용자 자신의 TPN 작성 및 QCNTRVC 설정	210	추적 작업의 RW 구성요소에 대한 자료 항목 해 석	261
iSeries용 DB2 Universal Database 어플 리케이션 리퀘스터의 TPN으로 QCNTRVC 설정	210	예: RW 추적 자료 분석	262
iSeries용 DB2 Universal Database 어플 리케이션 서버 작업을 디버깅하기 위한 사 용자 자신의 TPN 작성	210	RW 추적점의 설명	263
DB2 Universal Database for VM 어플 리케이션 리퀘스터의 TPN으로 QCNTRVC 설정	211	RWff RC — 수신 자료 스트림 추적점	263
DB2 Universal Database for z/OS 어플 리케이션 리퀘스터의 TPN으로 QCNTRVC 설정	211	RWff SN — 송신 자료 스트림 추적점	264
Linux, UNIX 및 Windows용 DB2 Universal Database 어플리케이션 리퀘스 터의 TPN으로 QCNTRVC 설정	211		
TCP/IP 서버에 대한 서비스 작업	212		
QRWOPTIONS 자료 영역	213		
예: 자료 영역을 작성하는 CL 명령	214		
예: 어플리케이션 프로그래밍	215		
예: 프로그램 정의	216		

RWQY S1 — 부분 송신 자료 스트림 추적점 1	264	RW_ff_m — 어플리케이션 리퀘스터 드라이버(ARD) 제어 블록 추적점	267
RWQY S2 — 부분 송신 자료 스트림 추적점 2	264	첫실패 자료 캡처(FFDC)	268
RWQY BP — 성공적인 페치 추적점	265	첫실패 자료 캡처 덤프	268
RWQY NB — 성공적이지 않은 페치 추적점	265	FFDC 덤프 출력 설명	271
RWQY P0 — 결과 세트 의사-개방	265	DDM 오류 코드	276
RWQY AR—처리된 배열 결과 세트	265	명령 검사 코드.	276
RWQY DA—배열 결과 세트 SQLDA	265	대화 프로토콜 오류 코드 설명.	278
RWQY DO—디버그 옵션.	265	DDM 구문 오류 코드 설명	279
RWQY L1 및 RWEX L1 — 아웃바운드 LOB 표 추적점에 저장됨	265	분산 데이터베이스 프로그래밍에 대한 관련 정보	280
RWQY L2 및 RWEX L2 - LOB 표 추적점으로부터 자료 스트림에 빌드됨	265	iSeries 서버 정보.	280
RWQY L0 및 RWEX L0 — 인바운드 LOB 표 추적점에 저장됨	265	분산 관계형 데이터베이스 라이브러리	282
RWAC RQ — RDB 액세스 요구 추적점	266	기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리	283
RWAC cb — 액세스 RDB 제어 블록 추적점.	266	구조 서적	284
RWSY FN: SYNCxxx [TYPE:x] -- 소스 TCP SYNC/RESYNC 추적점	267	IBM Redbook.	285
RWSY xx: yyyyyyy... -- 목표 TCP SYNC/RESYNC 추적점	267	코드 라이선스 및 면책사항 정보	286
		부록. 주의사항	287
		프로그래밍 인터페이스 정보	289
		상표	289
		조건	290

분산 데이터베이스 프로그래밍

분산 데이터베이스 프로그래밍은 i5/OS™ 라이선스 프로그램의 분산 관계형 데이터베이스 관리 부분에 대해 설명합니다. 분산 관계형 데이터베이스 관리는 어플리케이션 외부에 있으며 일반적으로 컴퓨터 네트워크 전역에 있는 자료에 액세스할 수 있는 기능을 제공합니다.

이 정보는 주로 하나 이상의 iSeries™ 서버에서 분산 관계형 데이터베이스의 개발, 관리 및 지원을 담당하는 어플리케이션 프로그래머를 위한 것입니다.

iSeries 데이터베이스에 익숙하지 않은 어플리케이션 프로그래머도 i5/OS(i5/OS)가 제공하는 전체 데이터베이스 지원 범위를 파악할 수 있습니다. 어플리케이션 프로그래머는 이 정보를 사용하여 분산 관계형 데이터베이스 어플리케이션이 실행되는 서버 문맥을 볼 수 있습니다.

이 설명서를 사용하려면 먼저 일반 프로그래밍 개념 및 전문용어에 익숙해야 하며 iSeries 서버 및 i5/OS 오퍼레이팅 시스템에 대해 전반적으로 이해해야 합니다.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.



V5R4의 새로운 사항

이 주제는 V5R4에 대해 작성된 변경사항을 요약적으로 설명합니다.

- | V5R4에서는 분산 관계형 데이터베이스 구조™(DRDA®) 지원이 다음 기능을 포함하도록 확장되었습니다. V5R4 이전에는 DRDA에서 이러한 지원을 사용할 수 없었습니다.
- | • 분산 트랜잭션 처리(XA/JTA). (자세한 정보는 분산 트랜잭션을 참조하십시오.)
- | • 프로파일 토큰에 대한 서버 지원. (자세한 정보는 보안 관련 API를 참조하십시오.)
- | • TCP/IP를 통한 DB2® Multisystem. (자세한 정보는 DB2 Multisystem을 참조하십시오.)
- | 다음 기능은 V5 R4의 iSeries에 새로운 기능입니다. DRDA를 통해 지원되기도 합니다.
- | • SQL문의 최대 한계가 2MB로 증가되었습니다.
- | • SQL 설명자 영역. 자세한 정보는 SQL 설명자 영역 및 SQLDA(SQL 설명자 영역을 참조하십시오.)
- | V5R4에서는 세 개의 프로그램 예가 다음 주제에 추가 또는 갱신되었습니다.
- | • 237 페이지의 『예: 삽입된 SQL을 사용하는 C 프로그램』
- | • 243 페이지의 『예: Java 프로그램』
- | • 216 페이지의 『예; 프로그램 정의』

변경되거나 새로운 사항을 보는 방법

기술 변경사항이 작성된 위치를 볼 수 있도록 하기 위해 이 정보는 다음을 사용합니다.

- 새로 작성되거나 변경된 정보가 시작되는 위치를 표시하기 위한  이미지.
- 새로 작성되거나 변경된 정보가 끝나는 위치를 표시하기 위한  이미지.

이 릴리스에서 새로 작성되거나 변경된 사항에 대한 다른 정보를 찾으려면 사용자에게 대한 메모를 참조하십시오.

인쇄 가능한 PDF

이 PDF를 보고 인쇄하려면 이 정보를 참조하십시오.


이 문서의 PDF 버전을 보거나 다운로드 하려면 분산 데이터베이스 프로그래밍(약 2964KB)을 선택하십시오.

PDF 파일 저장

PDF 파일을 보거나 인쇄하기 위해 워크스테이션에 저장하려면 다음을 수행하십시오.

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

Adobe Reader 다운로드

- 1 이 PDF를 보거나 인쇄하려면 시스템에 Adobe Reader가 설치되어 있어야 합니다. Adobe 웹 사이트
(www.adobe.com/products/acrobat/readstep.html)  에서 무료로 다운로드할 수 있습니다.

분산 데이터베이스 프로그래밍 소개

iSeries 서버의 분산 관계형 데이터베이스 지원은 IBM® 분산 관계형 데이터베이스 구조(DRDA)의 구현과 어플리케이션 리퀘스터 드라이버(ARD) 프로그램을 사용하는 기타 SQL 클라이언트의 통합으로 구성됩니다.

i5/OS, iSeries용 DB2 Universal Database™ 조회 관리자 및 SQL 개발 킷이 결합되어 이 지원을 제공합니다.

이 주제는 분산 관계형 데이터베이스 및 iSeries 서버에서 이 데이터베이스를 사용하는 방법에 대해 설명합니다.

또한, Spiffy사를 예를 들어 분산 관계형 데이터베이스를 설명합니다. 이 가상의 회사는 분산 관계형 데이터베이스 어플리케이션 프로그램에서 iSeries 서버를 사용합니다. 이 Spiffy사의 샘플은 이 주제에서 사용되는 모든 샘플에 대한 백그라운드를 형성합니다.

분산 관계형 데이터베이스 처리

관계형 데이터베이스는 컴퓨터에 있는 하나 이상의 표에 저장된 자료의 집합입니다.

표는 다음 표에서와 같이 수평 행과 수직 열로 구성된 자료의 1차원 배열입니다. 각 행에는 표의 각 열마다 하나씩 순서 값이 있습니다. 열에는 이름과 특정 자료 유형이 있습니다(예를 들어, 문자, 소수 또는 정수).

표 1. 일반 관계형 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 SQL과 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 상응하는 서버 오브젝트가 있습니다. 다음 표는 SQL 관계형 데이터베이스 용어와 iSeries 서버 용어 간의 관계를 보여줍니다.

표 2. SQL 용어 대 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 열에 나열된 대로 세 가지 유형이 있습니다. 자세한 정보는 관계형 데이터베이스 주제를 참조하십시오.</p>	<p>시스템 관계형 데이터베이스 또는 시스템 데이터베이스. 독립형 보조 기억 장치 풀에 저장되지 않고 iSeries 서버에 접속된 디스크에 존재하는 모든 데이터베이스 오브젝트.</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.</p>
	<p>주:</p>
	<ul style="list-style-type: none"> • V5R2에서 iSeries 서버는 독립형 보조 기억장치 풀이 서버에 구성되어 있는 경우 다중 관계형 데이터베이스에 대해 호스트가 될 수 있습니다. 시스템 관계형 데이터베이스는 항상 하나가 존재하고 사용자 관계형 데이터베이스는 하나 이상 존재할 수 있습니다. 각 사용자 데이터베이스는 시스템 데이터베이스의 모든 오브젝트를 포함합니다. • 그러나 사용자는 SQL 관점에서는 시스템 데이터베이스가 사용자 데이터베이스에 포함되는 것으로 보여지지만 확약 제어 관점에서는 시스템 데이터베이스가 별도 데이터베이스로 취급된다는 점을 유념해야 합니다. 자세한 정보는 트랜잭션 및 확약 제어 주제를 참조하십시오.
	<p>리모트 관계형 데이터베이스 또는 리모트 데이터베이스. 리모트로 액세스할 수 있는 iSeries 또는 다른 서버에 상주하는 데이터베이스.</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>라이브러리. 관련 오브젝트를 그룹화하며 이름으로 오브젝트를 찾을 수 있습니다.</p>
<p>주: 스키마는 일반적으로 콜렉션으로도 참조될 수 있습니다.</p>	
<p>표. 열과 행의 세트.</p>	<p>실제 파일(PF). 레코드 세트.</p>
<p>행. 열의 세트가 있는 표의 수평 부분.</p>	<p>레코드. 필드 세트.</p>
<p>열. 한 자료 유형의 표의 수직 부분.</p>	<p>필드. 자료 유형 관련 정보의 하나 이상의 바이트.</p>
<p>뷰. 하나 이상의 표에 있는 열과 행의 서브세트.</p>	<p>논리 파일. 최대 32개 실제 파일의 필드, 레코드 또는 둘 다의 서브세트.</p>
<p>색인. 표의 열에 있는 자료의 콜렉션을 오름차순이나 내림</p>	<p>논리 파일의 유형.</p>
<p>차순으로 논리적으로 배열한 것.</p>	
<p>패키지. 어플리케이션 서버가 사용하는 SQL문에 대한 제어</p>	<p>SQL 패키지. SQL 용어와 같은 의미.</p>
<p>구조가 들어 있는 오브젝트.</p>	
<p>카탈로그. 표, 패키지, 뷰, 색인 및 제한조건에 대한 정보가</p>	<p>상이한 오브젝트. 그러나 DSPFD(파일 설명 표시) 명령 및 DSPFFD(파</p>
<p>들어 있는 표와 뷰의 세트. QSYS2의 카탈로그 뷰에는 모</p>	<p>일 필드 설명 표시) 명령은 SQL 카탈로그 조회로 제공되는 것과 동일한</p>
<p>든 표, 패키지, 뷰, 색인 및 iSeries 서버의 제한조건에 대</p>	<p>정보의 일부를 제공합니다.</p>
<p>한 정보가 들어 있습니다. 또한 SQL 스키마에는 스키마의</p>	
<p>표, 패키지, 뷰, 색인 및 제한조건에 대한 정보만 들어 있</p>	
<p>는 뷰의 집합이 있습니다.</p>	

분산 관계형 데이터베이스는 자료를 사용하는 어플리케이션 프로그램과 자료 자체가 서로 다른 시스템에 위치할 때 또는 프로그램이 같은 서버에서 여러 데이터베이스에 위치한 자료를 사용할 때 존재합니다. 후자의 경우, 데이터베이스는 DRDA 프로토콜을 사용하여 한 서버 내에 있는 하나 이상의 데이터베이스에 액세스하는 식으로 분산되어 있습니다. 이런 환경에서 데이터베이스 연결은 두 가지 유형 중 하나가 됩니다. 로컬 또는

DRDA입니다. 대부분 한 번에 하나의 로컬 데이터베이스만 연결됩니다. 분산 관계형 데이터베이스의 한 가지 단순 형식은 다음 그림으로 나타나는데, 여기서 어플리케이션 프로그램은 하나의 시스템에서 실행되고 자료는 리모트 서버에 위치하고 있습니다.

분산 관계형 데이터베이스를 사용할 때 어플리케이션 프로그램이 실행되는 시스템을 어플리케이션 리퀘스터(AR)라고 하고 리모트 자료가 상주하는 시스템을 어플리케이션 서버(AS)라고 합니다. 클라이언트라는 용어는 종종 AR로 바꿔 쓸 수 있으며 서버는 AS로 바꿔 쓸 수 있습니다.

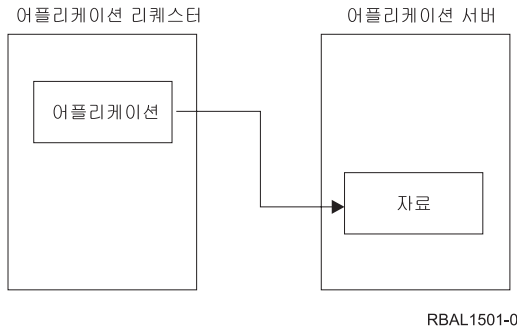


그림 1. 분산 관계형 데이터베이스

작업 단위는 다음 그림에서와 같이 하나 이상의 데이터베이스 요구와 완료된 작업 부분을 구성하는 연관된 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. 트랜잭션이란 용어는 작업 단위 개념을 설명하는 다른 표현입니다.

앞의 예에서 작업 단위는 부품을 공장 재고 계정 표에서 제거하고 재주문 표에 추가할 때까지 완료되지 않습니다. 요구가 완료되면 어플리케이션 프로그램은 작업 단위를 파악할 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 파악이나 롤백 연산이 적용되지 않았기 때문에 변경이 이루어집니다. 따라서 어플리케이션 프로그램은 데이터베이스에 대한 요구 집합을 단위로 처리합니다.

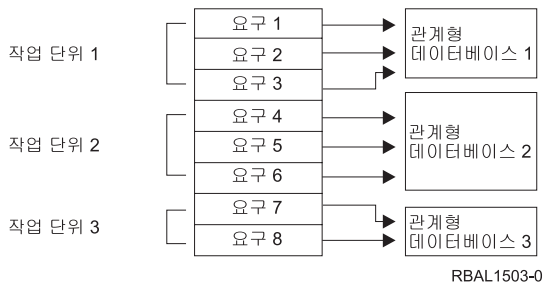


그림 2. 로컬 관계형 데이터베이스의 작업 단위

관련 개념

관계형 데이터베이스

트랜잭션 및 제약 제어 문제점 해결
 제약 제어에 대한 XA 트랜잭션 지원

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.

다음 그림을 참조하십시오.

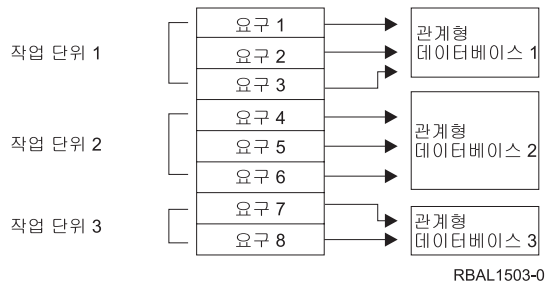


그림 3. 분산 관계형 데이터베이스의 리모트 작업 단위

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 사용자 또는 어플리케이션 프로그램이 한 작업 단위 내에서 여러 위치에 있는 자료를 읽거나 갱신할 수 있습니다. 한 작업 단위내의 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. 다음 그림은 이러한 개념을 나타낸 것입니다.

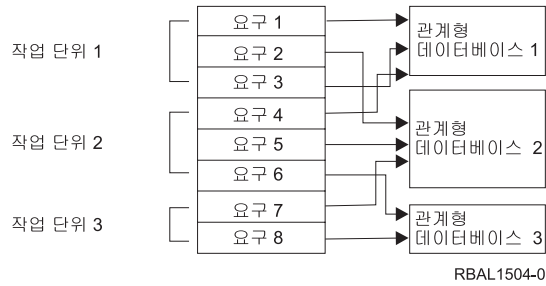


그림 4. 분산 관계형 데이터베이스의 분산 작업 단위

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

어플리케이션은 작업을 예약할 준비가 되면 예약을 시작하고 동기화 점 관리자 또는 트랜잭션 관리자는 예약 조정을 수행합니다.

DUW는 한 작업 단위에서 여러 데이터베이스 관리 시스템에 대해 갱신 액세스 또는 읽기 전용 액세스를 할 수 있도록 합니다.

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

이 설명은 추가 분산 관계형 데이터베이스 개념의 개요를 제공합니다.

IBM 시스템에서 일부 분산 관계형 데이터베이스 지원은 Linux®, UNIX® 및 Windows®용 DB2 Universal Database와 iSeries용 DataPropagator™ 제품에 의해 제공됩니다. 또한 iSeries 어플리케이션 프로그램을 작성할 때 이들 개념의 일부를 사용할 수 있습니다.

iSeries용 DB2 Universal Database는 OS/400® V5R1에서 시작할 때 APPC 및 TCP/IP 통신으로 리모트 작업 단위 및 분산 작업 단위를 모두 지원합니다. 분산 작업 단위를 넘는 복잡한 처리는 분산 요구입니다. 이 유형의 분산 관계형 데이터베이스 액세스를 사용하면 사용자 또는 어플리케이션 프로그램이 여러 위치에서 자료를 읽거나 갱신할 수 있는 단일 SQL문을 발행할 수 있습니다.

분산 관계형 데이터베이스의 표는 서로 구분될 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. 언로드 및 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 매월 등 사용자가 지정한 주기에 따라 소스 표에서 이들 사본을 새로 고칩니다. 스냅샷은 주기적으로 갱신된 정보를 수신하기 위해 자동 프로세스를 탐색하는 위치에 가장 유용합니다.

자료 복제는 서버가 표의 복사를 자동으로 갱신하는 것을 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. 분산된 표는 특정 부분에 대한 액세스나 갱신을 표의 일부가 있는 같은 위치에서 요구할 경우에 가장 효과적입니다.

관련 개념

11 페이지의 『i5/OS의 분산 관계형 데이터베이스』

i5/OS용 DB2 Universal Database는 i5/OS에 대한 모든 데이터베이스 관리 기능을 제공합니다. 시스템의 분산 관계형 데이터베이스 지원은 통신, 작업 관리, 보안 기능 및 기타 기능에 대한 지원과 마찬가지로 오피레이팅 시스템의 필수 파트입니다.

분산 관계형 데이터베이스 구조 지원

IBM 관계형 데이터베이스 제품은 분산 관계형 데이터베이스 처리에 대한 DRDA 지원을 사용합니다. DRDA 지원은 어플리케이션 프로그램과 리모트 관계형 데이터베이스 사이의 통신 프로토콜을 정의합니다.

DRDA(Distributed Relational Database Architecture) 지원은 IBM과 비IBM 환경 모두에서 분산 관계형 데이터베이스 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 액세스하는 구조를 제공합니다. 예를 들어, 둘 이상의 iSeries용 DB2 간의 관계 자료에 대한 액세스는 유사 환경에서의 분배입니다. iSeries용 DB2 UDB와 i5/OS에 내장된 것과 다른 클라이언트 또는 다른 유형의 시스템 간의 관계 자료에 대한 액세스는 다른 환경에서의 분배입니다. 이것의 한 가지 특정한 예로는 DB2 UDB for iSeries와 IBM DB2 Universal Driver for SQLJ 및 JDBC 간의 관계 자료에 대한 액세스입니다.

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

- | 분산 관계형 데이터베이스 구조는 분산 자료 관리(DDM) 구조의 확장입니다. 그러나 자료를 액세스하는 데 대한 DRDA와 DDM 메소드는 서로 다릅니다. DRDA는 SQL의 확장인 반면 DDM은 원시 I/O의 확장입니다.
- | 분산 관계형 데이터베이스 처리를 사용할 때 어플리케이션은 로컬 시스템의 관계형 데이터베이스 디렉토리를 사용하여 리모트 서버에 연결할 수 있습니다. 관계형 데이터베이스 디렉토리는 관계형 데이터베이스명과 해당 데이터베이스의 통신 경로 간에 필요한 링크를 제공합니다. 분산 관계형 데이터베이스에서만 실행하는 어플리케이션은 이 데이터베이스명을 식별하고 처리에 필요한 SQL문을 실행해야 합니다.

관련 참조

IBM DB2 Universal Driver for SQLJ and JDBC 1.0

DRDA 및 CDRA 지원

분산 관계형 데이터베이스의 흥미로운 가능성 중 한 가지는 데이터베이스가 여러 유형의 컴퓨터에 확산될 수 있을 뿐 아니라 이들 컴퓨터 역시 여러 국가 또는 영역에 걸쳐 있을 수 있다는 점입니다. iSeries 서버와 같은 동일 서버는 서버에서 사용되는 언어에 따라 자료를 다르게 코드화할 수 있습니다.

서로 다른 유형의 서버는 서로 다르게 코드화합니다. 예를 들어, S/390®, iSeries 서버 및 PS/2® 시스템은 숫자 자료를 자체의 고유 형식으로 코드화합니다. 그 밖에 S/390 및 iSeries 서버는 EBCDIC 코드화 체계를 사용하여 문자 자료를 코드화하는 반면, 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 Universal Database에서 숫자 자료를 읽으면 iSeries용 DB2 UDB는 이 숫자 자료를 S/390 형식으로 송신하고 i5/OS 데이터베이스 관리 시스템은 이를 iSeries 숫자 형식으로 변환합니다.

문자 자료의 처리는 더 복잡하지만 이 또한 분산 관계형 데이터베이스 내에서 처리할 수 있습니다.

CDRA를 사용하여 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, 미국 영어용으로 구성된 시스템은 덴마크어용으로 구성된 시스템이 å에 지정하는 것과 동일한 코드를 문자 }에 지정할 수 있지만 이들 두 시스템은 \$와 같은 동일한 문자에 서로 다른 코드를 지정할 수 있습니다.

다른 서버 간에 자료를 공유하는 경우, 사용자와 어플리케이션이 같은 방식으로 문자 자료를 볼 수 있어야 합니다. 다시 말해, 뉴욕의 PS/2 사용자와 코펜하겐의 iSeries 서버 사용자는 모두 각 서버에서 \$를 다르게 코드

화할 수 있더라도 \$를 \$로 볼 수 있어야 합니다. 더구나 코펜하겐의 사용자는 }가 뉴욕에서 저장된 문자인 경우 해당 코드가 덴마크의 å와 동일한 코드일 수 있더라도 }를 볼 수 있어야 합니다. 이렇게 하려면 \$는 PS/2 시스템에 적절한 문자 코드화(즉, 미국 영어 문자 세트인 ASCII)로 변환되어야 하며 뉴욕에서 코펜하겐으로 갈 때는 다시 덴마크 코드화(즉, 덴마크 문자 세트인 EBCDIC)로 변환되어야 합니다. 이러한 종류의 문자 변환은 iSeries 서버 및 IBM 분산 관계형 데이터베이스 관리자에 의해서 제공됩니다. 이 변환은 문자 자료 표시 구조(CDRA)에 따라 통일된 방식으로 수행됩니다.

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 이들 속성이 코드화 문자 세트 ID(CCSID)를 통해 식별되도록 지정합니다. DB2 Universal Database for z/OS, DB2 Universal Database for VM 및 i5/OS 데이터베이스 관리 시스템의 모든 문자 자료는 코드화 체계, 문자 세트 및 코드 페이지의 특정 조합을 나타내는 CCSID를 가집니다. 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 i5/OS 국제화를 참조하십시오. iSeries 서버에서 CCSID 사용에 대한 설명은 코드화 문자 세트 ID(CCSID)를 참조하십시오.

관련 개념

i5/OS 국제화

관련 참조

98 페이지의 『코드화 문자 세트 ID』

어떤 국가의 자국어 지원은 최소 문자 세트에 대한 적절한 처리를 요구합니다.

어플리케이션 리퀘스터 드라이버 프로그램

어플리케이션 리퀘스터 드라이버(ARD) 프로그램은 SQL 어플리케이션이 iSeries용 DB2 Universal Database 이외의 데이터베이스 관리 시스템이 관리하는 자료에 액세스할 수 있도록 해주는 일종의 종료 프로그램입니다.

iSeries 클라이언트는 다음 조작 중 ARD 프로그램을 호출합니다.

- 관계형 데이터베이스(RDB) 매개변수가 ARD 프로그램에 해당하는 RDB 이름과 일치할 때 CRTSQLPKG(SQL 패키지 작성) 명령 또는 CRTSQLxxx 명령을 사용하여 수행되는 SQL 사전컴파일의 패키지 작성 단계.
- 현재 연결이 ARD 프로그램에 대응하는 이름일 때의 SQL문의 처리.

ARD 프로그램은 이들 호출을 사용하여 SQL문 및 이 명령문의 정보를 리모트 관계형 데이터베이스로 전달하고 결과를 다시 어플리케이션 리퀘스터(AR)로 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(관계형 데이터베이스 디렉토리 항목 추가) 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).

관련 참조

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

SQL 패키지 작성(CRTSQLPKG) 명령

관련 정보

어플리케이션 프로그래밍 인터페이스

i5/OS의 분산 관계형 데이터베이스

i5/OS용 DB2 Universal Database는 i5/OS에 대한 모든 데이터베이스 관리 기능을 제공합니다. 시스템의 분산 관계형 데이터베이스 지원은 통신, 작업 관리, 보안 기능 및 기타 기능에 대한 지원과 마찬가지로 오퍼레이팅 시스템의 필수 파트입니다.

i5/OS 오퍼레이팅 시스템은 분산 관계형 데이터베이스 구조(DRDA) 구현을 지원하는 기타 서버와 함께 분산 관계형 데이터베이스 네트워크의 부분이 될 수 있습니다. i5/OS 오퍼레이팅 시스템은 유사 환경 또는 상이한 환경에서 어플리케이션 리퀘스터(AR) 또는 어플리케이션 서버(AS)가 될 수 있습니다. 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 한 작업 단위 내에서 한 서버에 있는 부품의 재고 수를 점차 감소시키면서 다른 서버에 있는 한 부품의 재고 수를 점차 증가시킨 후 2단계 확약 프로세스를 사용하여 단일 작업 단위가 끝날 때 이들 리모트 데이터베이스에 대한 변경사항을 확약할 수 있습니다. iSeries용 DB2 Universal Database는 분산 요구를 지원하지 않으므로 각 SQL문으로 하나의 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. 자세한 정보는 분산 관계형 데이터베이스로 연결을 참조하십시오.

DRDA 액세스 외에도, 어플리케이션 리퀘스터 드라이버(ARD) 프로그램을 사용하여 DRDA를 지원하지 않는 데이터베이스에 액세스할 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.

iSeries 서버에서 서버는 스냅샷 및 복제의 분배 기능을 자동으로 수행하지 않습니다. iSeries 서버에 iSeries용 DataPropagator V8.1 제품을 설치 및 구성하여 이들 기능을 수행할 수 있습니다. 또한 사용자가 작성한 어플

리케이션 프로그램에서 이 기능을 사용할 수 있습니다. 분산 관계형 데이터베이스에서 이들 기능을 구성하는 방법에 대한 자세한 정보는 분산 관계형 데이터베이스에 대한 자료 가용성 및 보호에 설명되어 있습니다.

iSeries 서버에서 분산 요구 기능은 직접적으로 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.

i5/OS 라이선스 프로그램은 SQL에 대한 런타임 지원을 포함하고 있습니다. 분산 관계형 데이터베이스 요구를 처리하거나 iSeries 서버에 SQL 콜렉션을 작성하기 위해 iSeries용 DB2 UDB 어플리케이션 리퀘스터 또는 어플리케이션 서버에 iSeries용 DB2 UDB 조회 관리자 및 SQL Development Kit 라이선스 프로그램을 설치할 필요는 없습니다. 그러나 SQL문으로 프로그램을 사전컴파일하거나 대화식 SQL을 실행 또는 iSeries용 DB2 Universal Database 조회 관리자를 실행하려면 iSeries용 DB2 Universal Database 조회 관리자 및 SQL Development Kit 프로그램이 필요합니다.

- 1 iSeries에서 DRDA를 구현하기 위한 통신 지원은 APPN(Advanced Peer-to-Peer Networking[®]) 유무에 관
- 1 계없이 APPC(Advanced Program-to-Program Communication) 프로토콜을 통해 IBM SNA(Systems Network
- 1 Architecture) 또는 TCP/IP 하에서 제공됩니다.

관련 개념

7 페이지의 『기타 분산 관계형 데이터베이스 용어 및 개념』

이 설명은 추가 분산 관계형 데이터베이스 개념의 개요를 제공합니다.

84 페이지의 『분산 관계형 데이터베이스로 연결』

분산 관계형 데이터베이스 어플리케이션이 분산하는 것은 다른 서버의 관계형 데이터베이스에 연결하는 기능입니다.

146 페이지의 『자료 가용성 및 보호』

분산 관계형 데이터베이스 환경에서 자료 가용성은 네트워크의 개별 서버에 있는 자료를 보호할 뿐 아니라 사용자가 네트워크 전역에 있는 자료에 액세스할 수 있는지 확인하는 일도 포함합니다.

APPC, APPN 및 HPR

통신 구성 PDF

TCP/IP 구성

OptiConnect

iSeries 분산 관계형 데이터베이스 관리

iSeries 서버에서 분산 관계형 데이터베이스를 관리하려면 i5/OS 라이선스 프로그램의 자원 및 톨에 대한 폭 넓은 지식이 필요합니다.

이 주제는 iSeries 서버에서 분산 관계형 데이터베이스를 관리하는 데 도움이 될 수 있는 오퍼레이팅 시스템과 함께 사용 가능한 여러 가지 기능의 개요를 제공합니다. 이 주제는 iSeries 서버 네트워크(유사 환경)의 분산 관계형 데이터베이스 기능 및 태스크에 대해 설명합니다. 유사 환경과 상이한 환경에서의 iSeries 분산 관계형 데이터베이스 기능 간의 차이점은 이 주제의 일반 설명에서만 제시됩니다.

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

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

- 계획 및 설계 주제는 분산 데이터베이스를 계획하고 설계할 때 몇 가지 중요한 고려사항에 대해 논의합니다.
- 보안 주제는 리모트 관계형 데이터베이스에 대한 DRDA 액세스 및 통신을 포함하여 iSeries 분산 관계형 데이터베이스의 보안 고려사항에 대한 정보를 제공합니다.
- 구성 주제는 iSeries 서버의 서브시스템 및 관계형 데이터베이스 디렉토리에 대한 설명과 함께 분산 데이터베이스에 자료를 입력하는 방법에 대한 정보를 제공합니다.
- 분산 관계형 데이터베이스 관리 주제는 네트워크에서 수행되는 분산 관계형 데이터베이스 작업을 관리할 수 있는 방법에 대해 설명합니다.
- 자료 가용성 및 보호 주제는 문제점 발생 시 회복 시간을 줄이고 iSeries 서버에서 프로그램 및 자료를 보호하기 위한 툴과 기술에 대해 설명합니다. 또한 여기서는 네트워크 사용자가 필요할 때에 해당 네트워크 전역에서 관계형 데이터베이스와 표에 액세스하도록 보장하는 대안에 대한 내용을 제공합니다.
- 성능 주제는 네트워크, 시스템 및 데이터베이스의 설계를 개선할 수 있는 방법에 대해 설명합니다.

DRDA를 지원하는 다른 IBM 시스템에 대한 자세한 정보를 원할 경우, 이 시스템과 함께 제공되는 정보를 참조하거나 관련 정보의 분산 관계형 데이터베이스 라이브러리 및 기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리에 나열된 서적을 참조하십시오.

관련 개념

113 페이지의 『관리』

사용자는 분산 관계형 데이터베이스의 관리자로서, 여러 서버에서 실행하는 작업에 대해 책임이 있습니다.

146 페이지의 『자료 가용성 및 보호』

분산 관계형 데이터베이스 환경에서 자료 가용성은 네트워크의 개별 서버에 있는 자료를 보호할 뿐 아니라 사용자가 네트워크 전역에 있는 자료에 액세스할 수 있는지 확인하는 일도 포함합니다.

26 페이지의 『초기 설정』

iSeries 분산 관계형 데이터베이스에 대한 런타임 지원은 오퍼레이팅 시스템에서 제공됩니다. 따라서 오퍼레이팅 시스템을 설치하면 분산 관계형 데이터베이스 지원이 설치됩니다.

165 페이지의 『성능』

서버에서 실행되는 어플리케이션 프로그램의 종류에 관계없이 성능은 항상 주요 관심사입니다. 분산 관계형 데이터베이스의 경우, 네트워크, 서버 및 어플리케이션 성능이 모두 중요합니다.

17 페이지의 『계획 및 설계』

분산 관계형 데이터베이스의 성공적인 작업을 위한 첫 번째 요구사항은 계획을 통해 이루어집니다. 분산 관계형 데이터베이스를 사용하기 위한 결정을 할 때 기업의 요구와 목표를 고려해야 합니다.

56 페이지의 『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.

249 페이지의 『사용자 FAQ』

이 주제에서는 iSeries 서버의 네트워크(유사 환경)에서 분산 관계형 데이터베이스에 대한 iSeries 지원을 중점적으로 설명합니다.

관련 참조

282 페이지의 『분산 관계형 데이터베이스 라이브러리』

이 주제에서는 IBM 분산 관계형 데이터베이스 구조(DRDA) 구현에 대한 백그라운드 및 일반 지원 정보를 제공합니다.

283 페이지의 『기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리』

이 주제는 기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리에 대해 설명합니다.

280 페이지의 『분산 데이터베이스 프로그래밍에 대한 관련 정보』

다음은 분산 데이터베이스 프로그래밍 주제에 관련된 제품 매뉴얼과 IBM Redbook™(PDF 형식), 웹 사이트 및 Information Center 주제입니다. 모든 PDF를 보거나 인쇄할 수 있습니다.

예: Spiffy사 분산 관계형 데이터베이스

Spiffy사는 여러 IBM 매뉴얼에서 분산 관계형 데이터베이스 지원을 설명하는 데 사용되었습니다. 이 주제에서 이 가상의 회사는 iSeries 서버 네트워크에서 DRDA에 대한 iSeries 서버 지원을 설명하기 위해 다소 변경되었습니다.

이 주제 전체에서 사용되는 예는 특정 기능, 연결 및 프로세스를 설명합니다. 이들 예는 다른 분산 관계형 데이터베이스 서적에 사용된 예들과 정확하게 일치하지는 않지만 유사하게 작성되었습니다.

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.

Spiffy 조직 및 시스템 프로파일

Spiffy사는 지역 사무소 및 로컬 판매 대리점의 네트워크를 통해 고객에게 자동차를 판매하고 서비스하는 가상의 국내 제품 도매 상인입니다.

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.

다음 그림은 Spiffy사의 시스템 구성도입니다.

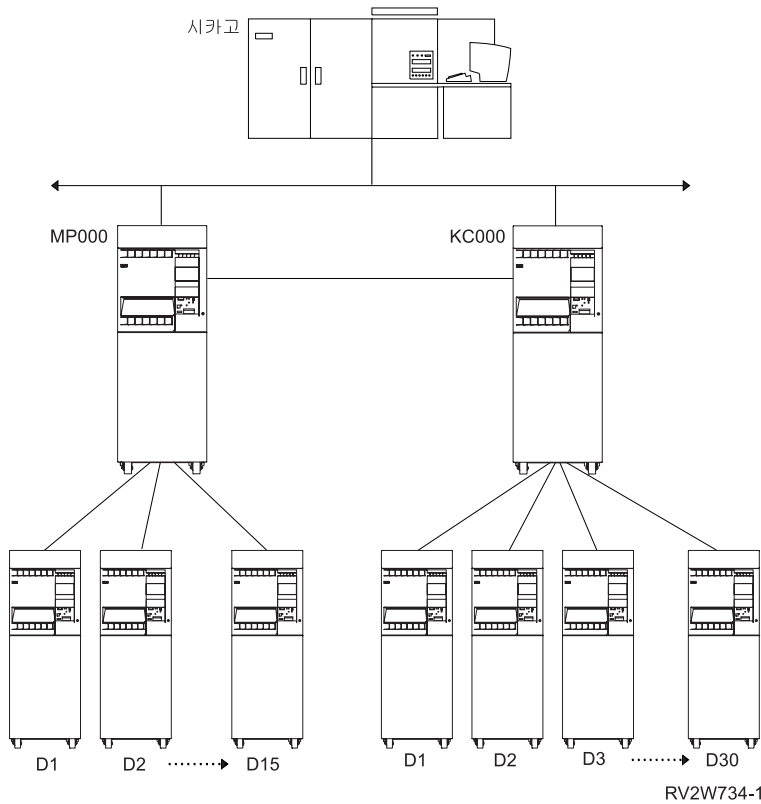


그림 5. Spiffy사의 시스템 구성

Spiffy사는 전국적인 통합 텔레커뮤니케이션을 빌드중입니다. 자동차 부문의 경우, 지역 유통 센터와 판매 대리점에 iSeries 서버의 네트워크를 설정하고 있습니다. 이들 네트워크는 중앙 차량 대리점에 있는 S/390으로 연결됩니다. 이 네트워크는 경쟁에서의 우위를 유지하기 위해 중요한 업무 자산을 고려하였습니다.

중앙 대리점은 관련 의사결정 지원 소프트웨어와 함께 S/390에서 z/OS용 DB2 Universal Database를 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 2000명에서 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.

Spiffy사에 대한 분산 관계형 데이터베이스 관리

각 대리점은 독립 기업으로 자료 처리 자원과 프로그래머를 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. 이 회사는 자체 판매 대리점을 운영하므로 분산 관계형 데이터베이스 환경에 액세스하거나 액세스하지 않는 완전한 판매 대리점 소프트웨어를 가지고 있습니다. Spiffy사 대리점은 소프트웨어 툴 세트를 모두 사용합니다. 대부분의 개인 가맹점도 이들 소프트웨어가 특정하게 Spiffy사의 업무 방식에 맞게 만들어졌기 때문에 이를 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 포함됩니다.

다음은 네트워크에서 각 활동 레벨에 대한 데이터베이스 책임사항입니다.

판매 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

기타 활동

지역 유통 센터 활동 외에도 미니아폴리스 iSeries 서버 소관 센터는 다음 활동을 수행합니다.

- iSeries 서버 어플리케이션 개발
- 지역 센터의 헬프 데스크 운영
- 데이터베이스 성능 조정
- 데이터베이스 문제 해결

이 주제 전체에서 사용된 예는 이들 활동 중 하나 이상과 연관됩니다. 많은 예가 고객 서비스나 수리 스케줄로 인해 재고로부터 부품을 얻는 프로세스입니다. 기타 예는 Spiffy사의 분산 관계형 데이터베이스 네트워크의 서버에 대한 설정, 보안, 모니터 및 문제점 해결에 사용되는 분산 관계형 데이터베이스 관리 타스크를 보여줍니다.

계획 및 설계

분산 관계형 데이터베이스의 성공적인 작업을 위한 첫 번째 요구사항은 계획을 통해 이루어집니다. 분산 관계형 데이터베이스를 사용하기 위한 결정을 할 때 기업의 요구와 목표를 고려해야 합니다.

어플리케이션 프로그램을 코딩하는 방법, 자료를 저장하는 장소, 어플리케이션 프로그램과 자료를 연결하는 네트워크 설계 모두 중요한 설계 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. 다음은 전략을 세울 때 특별한 주의가 필요한 작업입니다.

- 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.

관련 개념

168 페이지의 『데이터베이스를 통해 분산 관계형 데이터베이스 성능 향상』

분산 관계형 데이터베이스 성능은 전체적인 데이터베이스의 설계에 의해 영향을 받습니다. 분산 자료의 위치, 사용하는 제약 제어의 레벨 및 SQL 색인의 설계는 모두 성능에 영향을 줍니다.

분산 관계형 데이터베이스에 대한 요구 및 예상 식별

분산 관계형 데이터베이스의 요구 및 예상을 분석할 때 다음 사항을 고려하십시오.

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 포함시킬 것인지를 결정하는 것입니다.

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

다음 요구사항이 있는 어플리케이션은 일반적으로 분산 관계형 데이터베이스의 부적합한 후보입니다.

- 자료를 중심부에 유지하고 그리고 리모트 사용자가 해야 할 대부분의 작업이 중심부에 있습니다.
- 꾸준히 높은 성능, 특히 빠른 응답 시간을 요구합니다. 네트워크에서 자료를 이동시키는 시간은 오래 걸립니다.
- 꾸준히 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.

iSeries용 DataPropagator, V8.1이 제공하는 추가 자료 복사 기능과 함께 분산 작업 단위의 분산 데이터베이스 기능은 사용자가 iSeries 서버에서 수행할 수 있는 활동의 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 들어 다음 중 하나를 수행할 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 다음 표에 나열됩니다. 이 대안은 사용할 수 있는 기능을 보충하거나 대체하는 데 사용될 수 있습니다.

표 3. 분산 관계형 데이터베이스에 대한 대체 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®에 액세스하는 데 사용될 수 있습니다. 	<ul style="list-style-type: none"> • SQL은 복잡한 기능에 보다 효율적입니다. • 다른 분산 관계형 데이터베이스 플랫폼에 액세스하지 못할 수 있습니다. • CCSID 및 숫자 자료 변환을 수행하지 않습니다.
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 데이터베이스의 설계는 어플리케이션, 네트워크 고려사항 및 자료 고려사항에 대한 선택을 포함합니다.

추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계

분산 관계형 데이터베이스는 로컬 데이터베이스에서 전용으로 사용하기 위해 개발한 어플리케이션과는 다른 요구사항을 갖고 있습니다.

이들 차이점에 대해 적절히 계획하려면 어플리케이션을 설계할 때 다음 고려사항을 유념하십시오.

- 가능하면 분산 작업 단위(DUW)를 사용하십시오.

주: OS/400 버전 5 릴리스 1 이전에는 iSeries 서버의 TCP/IP에서 2단계 확약 지원을 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 작업으로 제한되지 않습니다. 통합 데이터베이스 입/출력을 사용하거나 다른 유형의 처리를 수행할 수 있습니다.
- 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 준비될 때 리모트 작업 단위(RUW) 연결 메소드가 사용되거나 분산 어플리케이션의 다른 노드가 DUW를 지원하지 않을 경우 한 사이트에서 수행되어야 하는 점을 주의하십시오.
- 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 디폴트 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 메소드에서 CONNECT 명령문은 관계형 데이터베이스로 새 연결을 설정하기 전에 기존 연결을 모두 종료합니다. DUW 연결 관리 메소드에서 CONNECT 명령문은 기존 연결을 종료하지 않습니다.

분산 관계형 데이터베이스에 대한 네트워크 고려사항

네트워크 설계는 분산 관계형 데이터베이스의 성능에 직접 영향을 줍니다.

특정 네트워크에서 제대로 작동하는 분산 관계형 데이터베이스를 적절히 설계하려면 다음을 수행하십시오.

- 회선 속도는 어플리케이션 성능에 매우 중요하므로 기본 분산 관계형 데이터베이스 어플리케이션이 효율적인 성능을 달성하려면 네트워크의 적절한 위치에 충분한 용량을 제공하십시오.
- 필요한 경우, 사용자의 능력을 갱신하는 데 사용될 수 있는 통신 하드웨어와 소프트웨어를 평가하십시오.
- APPC(Advanced Program-to-Program Communication) 연결의 경우, 네트워크 정의 시 지정된 세션 한계 및 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 스레드에 대한 보조 기억장치 풀(ASP) 그룹으로 설정된 어플리케이션 리퀘스터(AR)에서 데이터베이스로 연결하는 데 APPC 보호 DUW 대화를 사용할 수 없다는 점을 이해하십시오.
- 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 위치는 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 어플리케이션이 의도하지 않은 리모트 데이터베이스로 연결되게 할 수 있으며, 일단 연결되면 의도하지 않은 오브젝트에 액세스하게 될 수 있습니다. 네트워크를 연결할 때는 명명에 특별한 주의를 기울이십시오.
 - 사용자 데이터베이스의 각 위치는 고유한 ID를 제공해야 합니다. 두 개의 서로 다른 서버에 있는 사용자 데이터베이스가 PAYROLL로 이름이 지정될 경우, 한 어플리케이션이 같은 서버로부터 이들 둘 다에 액세스할 필요가 있을 때 이름 지정 충돌이 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.

관련 개념

통신 관리 PDF

분산 관계형 데이터베이스에 대한 자료 고려사항

필요한 어플리케이션에 관한 자료 위치는 분산 관계형 데이터베이스를 설계할 때 중요한 고려사항입니다.

이러한 배치 결정을 할 때 다음 항목을 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

분산 관계형 데이터베이스를 관리할 때 다음 전략을 유념하십시오.

분산 관계형 데이터베이스에 대한 일반 작업

분산 관계형 데이터베이스의 일반 작업을 계획하려면 성능과 가용성을 모두 고려하십시오.

다음 설계 고려사항은 분산 관계형 데이터베이스의 성능 및 가용성을 모두 향상시키는 데 도움이 될 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.

다음은 수행하여 성능을 향상시킬 수도 있습니다.

- 자료 및 어플리케이션을 서로 다른 위치에 보유해야 하는 경우, 그 성능을 허용 가능한 한계 내에서 유지하려면 다음을 수행하십시오.
 - 어플리케이션이 사용할 자료 열만 검색하여 네트워크의 자료 통신량을 가능한 낮게 유지하십시오. 즉, SELECT 명령문의 부분으로 열 이름 리스트 자리에 *를 사용하는 것을 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 대한 CONNECT 다음에 SQL CALL문을 사용하면 어플리케이션 서버(AS)로 처리가 분산되어 성능이 향상되므로 회선 통신량을 줄일 수 있습니다.
 - FOR FETCH ONLY절을 지정하여 읽기 전용 조회를 사용하십시오.
 - 조회 블록에 대한 규칙을 확인하십시오. 예를 들어, iSeries 대 iSeries 조회에서 읽기 전용 자료의 블록은 COMMIT(*NONE)에 대해서만 수행되거나 ALWBLK(*ALLREAD)가 지정된 경우 COMMIT(*CHG) 및 COMMIT(*CS)에 대해 수행됩니다.
 - 가능하면 리모트 자료의 위치에 있는 로컬 자료를 사용하여 리모트 자료에 대한 액세스 수를 낮게 유지하십시오.
 - SQL 세트 작업을 사용하여 단일 SQL 요구로 어플리케이션 리퀘스터에서 복수 행을 처리하십시오.
 - 리모트 작업 단위(RUW) 연결 관리 실행 시 DDMCNV(*KEEP)를 사용하거나 분산 작업 단위(DUW) 연결 관리를 실행하여 연결이 드롭되지 않도록 하십시오.
- 다음을 수행하여 충분한 네트워크 용량을 제공하십시오.
 - 고속, 고 대역폭의 회선을 설치하거나 네트워크의 적절한 점에 회선을 추가하여 네트워크 용량을 증가시키십시오.
 - 경합을 줄이거나 특정 프로세서에서의 경합 균형을 개선하십시오. 예를 들면, 기존 어플리케이션을 호스트 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 적절히 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.

다음은 수행하여 가용성을 향상시킬 수도 있습니다.

- 일반적으로 네트워크의 자료 통신량을 제한하십시오.
- 자료 및 어플리케이션을 서로 다른 위치에 보유해야 하는 경우 그 가용성을 허용 가능한 한계 내에 유지하려면 다음을 수행하십시오.
 - 대체 네트워크 라우트를 설정하십시오.
 - 가용성에 대한 시간 존 차이점의 효과를 고려하십시오.
 - 서버를 시작하는 데 규정된 사용자를 사용할 수 있는가?

- 근무 시간의 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 합니다. 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 위치를 선택할 때 다음을 고려하십시오.

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

APPC(Advanced Program-to-Program Communication)로 연결된 모든 서버는 다음을 수행할 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 수행할 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 권한이 없는 사용자가 자료에 액세스하려는 시도를 추적하거나 보안에 관한 기타 이벤트를 추적할 수 있도록 하는 보안 감사 기능을 포함합니다. 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.

관련 개념

56 페이지의 『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.

분산 관계형 데이터베이스에 대한 계정

분산 자료 사용에 대한 통계와 비용을 계산할 수 있어야 합니다.

다음 사항을 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.
- 계정 정보를 상관시킬 경우 다양한 서버의 시간대를 고려할 필요가 있을 수 있습니다. 각 서버 시계가 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 쓰기 위해 DSPJRN(저널 표시) 명령을 사용할 수 있습니다. 그런 다음, 사용자 작성 프로그램이나 조회 기능을 사용하여 계정 자료를 분석할 수 있습니다.
- NetView[®] 계정 자료. 네트워크 자원 사용에 대한 계정 자료를 기록하는 데 NetView 라이선스 프로그램을 사용할 수 있습니다.

관련 참조

DSPJRN(저널 표시) 명령

131 페이지의 『분산 관계형 데이터베이스의 작업 사용 통계』

서버를 사용하고 있는 사용자와 이 사용자가 사용하고 있는 서버 자원을 판별할 수 있도록 iSeries 서버에

서 작업 사용 계정 기능이 자료를 수집합니다. 일반적인 작업 사용 통계는 처리 장치, 프린터, 표시장치 및 데이터베이스와 통신 기능의 사용과 같이 사용된 서버 및 자원에서 실행되는 작업의 세부사항을 제공합니다.

분산 관계형 데이터베이스에 대한 문제점 분석

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

다음 사항을 고려하십시오.

- 분산 데이터베이스 처리 문제는 다양한 방법으로 명시됩니다. 예를 들어, 문제점을 감지한 서버가 분산 데이터베이스 어플리케이션에 오류 리턴 코드를 전달할 수 있습니다. 또한, 응답이 느리거나 잘못되거나 존재하지 않을 수 있습니다.
- 분산 데이터베이스 처리 문제를 진단하는 데 틀을 사용할 수 있습니다. 예를 들어, 각 분산 관계형 데이터베이스 제품은 분산 자료 처리 문제점을 진단하는 데 도움을 주는 추적 기능을 제공합니다.
- iSeries 서버가 서버 고장을 감지하면 서버는 고장을 감지한 후에 즉시 프로그램 상태에 대한 정보를 기록합니다.

주: IBM 프로그램을 수정해야 하며 네트워크에 네트워크 분배 관리자(NDM)가 있는 S/390이 설치된 경우, 이 NDM 및 Distributed System Node Executive 제품을 사용하여 네트워크의 해당 서버로 갱신사항 및 대체사항을 수신하고 전송할 수 있습니다.

분산 관계형 데이터베이스의 백업 및 회복

단일 서버 환경에서 백업과 회복은 로컬로 발생합니다. 그러나 분산 데이터베이스의 경우 백업과 회복은 리모트 위치에도 영향을 줍니다.

iSeries 서버를 사용하여 개별 표나 컬렉션 그룹을 백업하고 회복시킬 수 있습니다. 백업과 회복을 로컬로만 실행할 수도 있지만 적절한 백업이 지원되지 않는 서버에는 덜 중요한 자료를 두려고 할 수 있습니다. 백업 및 회복 프로시저는 하나 이상의 어플리케이션 서버에 존재할 수 있는 자료에 대해 일치해야 합니다. 네트워크에는 서버가 하나 이상 있으므로 이러한 자료를 2차 서버에 저장하여 어떤 형식으로 네트워크에서 항상 사용할 수 있게 하려 할 수 있습니다. 이와 같은 전략은 데이터베이스를 네트워크에 분산하기 전에 특별히 계획하고 구성해야 합니다.

초기 설정

iSeries 분산 관계형 데이터베이스에 대한 런타임 지원은 오퍼레이팅 시스템에서 제공됩니다. 따라서 오퍼레이팅 시스템을 설치하면 분산 관계형 데이터베이스 지원이 설치됩니다.

그러나 특히 APPC(Advanced Program-to-Program Communication) 환경에서 어플리케이션 리퀘스터 및 어플리케이션 서버가 작업을 송신하고 수신할 준비가 되게 하려면 몇 가지 설정 작업이 필요할 수 있습니다. 대화식, 일괄처리, 스푼 및 통신 작업을 제어하는 데 하나 이상의 서브시스템을 사용할 수 있습니다. 네트워크의 모든 어플리케이션 리퀘스터(AR)에는 연결 정보를 통해 관계형 데이터베이스 디렉토리가 설정되어 있어야 합니다. 마지막으로 네트워크 전체에 있는 어플리케이션 서버의 표에 자료를 넣을 수 있습니다.

관계형 데이터베이스 디렉토리에는 통신 네트워크 매개변수로 변환된 데이터베이스명과 값이 들어 있습니다. AR 은 독립형 보조 기억장치 풀(IASP, 독립형 디스크 풀이라고도 함)에 구성된 모든 사용자 데이터베이스 및 로컬 데이터베이스를 포함하여 네트워크에 있는 각 데이터베이스에 대한 항목을 가지고 있어야 합니다. 이들 로컬 항목은 시스템에 의해 자동으로 또는 수동으로 추가될 수 있습니다. 각 디렉토리 항목은 고유한 관계형 데이터베이스 이름과 대응하는 통신 경로 정보로 구성됩니다. 아웃바운드 연결에 대한 우선 암호 보안에 대한 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
- i5/OS 조회 관리
- 자료 파일 유틸리티(DFU)
- CPYF(파일 복사) 명령

상이한 서버의 분산 관계형 데이터베이스 네트워크에 대한 연결 및 설정 정보는 *Distributed Relational Database Cross-Platform Connectivity* 서적, SG24-4311-02에서 찾을 수 있습니다.

관련 개념

독립형 보조 기억장치 풀(ISAP)

독립형 디스크 풀

관련 참조

CPYF(파일 복사) 명령

iSeries 서버에서 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. 수행되고 있는 작업을 지원하고 원하는 성능 특성을 유지보수하기 위해 필요에 따라 서브시스템을 쉽게 시작하고 종료할 수 있습니다.

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

대화식 작업은 워크스테이션에 사인 온할 때 시작되고 사인 오프할 때 종료됩니다. APPC(Advanced Program-to-Program Communication) 일괄처리 작업은 다른 시스템의 프로그램 시작 요구로부터 시작되는 작업입니다. 비통신 일괄처리 작업은 작업 대기행렬로부터 시작됩니다. 작업 대기행렬은 통신 일괄처리 작업을 시

작 중일 때에는 사용되지 않습니다. 스펀링 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.

관련 개념

132 페이지의 『TCP/IP 서버 관리』

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

DRDA용 작업 관리 환경 설정

제어 서브시스템이라는 한 서브시스템은 사용자가 서버를 로드할 때 자동으로 시작합니다. IBM은 두 가지 제어 서브시스템 구성을 제공합니다.

첫 번째 구성은 다음과 같은 서브시스템을 포함합니다.

- 제어 서브시스템, QBASE는 대화식, 일괄처리 및 통신 작업을 지원합니다.
- QSPL은 스펀링 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 워크스테이션에서 시작된 대화식 작업을 지원합니다.
- QCMN은 통신 작업을 지원합니다.
- QBATCH는 일괄처리 작업을 지원합니다.
- QSPL은 스펀링 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.

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

CHGSBSD(서브시스템 설명 변경) 명령을 사용하여 IBM 제공 서브시스템 설명 또는 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.

관련 개념

통신 관리 PDF

작업 관리

관련 참조

CHGSBSD(서브시스템 설명 변경) 명령

CHGSYSVAL(시스템 값 변경) 명령

APPC 서브시스템

시스템 네트워크 구조(SNA) 네트워크를 사용하는 분산 관계형 데이터베이스에서 관리자가 각 서버에서 관리 계획을 세워야 하는 기본 작업 유형은 통신 작업 및 대화식 작업입니다.

네트워크에 있는 서버는 통신 작업을 시작하여 어플리케이션 리퀘스터(AR)의 요구를 처리합니다. 다른 서버에 대한 AR의 통신 요구는 보통 로컬 시스템의 대화식 또는 일괄처리 작업에서 시작됩니다.

분산 관계형 데이터베이스 네트워크 서버에 대해 효율적인 작업 관리 환경을 설정하면 네트워크에 있는 각 어플리케이션 서버(AS) 및 AR의 특정 요구에 따라 시스템 자원을 할당하여 전체 네트워크 성능을 향상시킬 수 있습니다.

i5/OS 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 서브시스템을 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.

Spiffy사의 분산 관계형 데이터베이스 네트워크의 일부 시스템에 대한 다양한 시스템 요구사항을 충족하려면 최고의 네트워크 효율성을 위해 다양한 작업 관리 환경이 필요할 수 있습니다. 다음은 분산 관계형 데이터베이스 관리자가 Spiffy사 분산 관계형 데이터베이스 네트워크에 있는 각 iSeries 서버의 요구를 충족시키기 위해 작업 관리 서브시스템을 계획하는 방법을 보여줍니다.

Spiffy사의 시스템 조직에서 소형 판매 대리점은 해당 사용자가 서버에서 가지고 있는 여러 가지 작업에 대해 QBASE 제어 레벨로 충족될 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 회사 직원 레코드 보관, 마케팅 및 판매 계획, 보고서 작성 등과 같이 분산 관계형 데이터베이스 관련 작업이 아닌 여러 대화식 작업도 시작할 수 있습니다. 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(켄자스 시) 지역 센터와 여기서 지원되는 판매 대리점이 사용하는 분산 관계형 데이터베이스 네트워크의 일부 특정 부분을 자세히 살펴보면 켄자스 시의 분산 관계형 데이터베이스 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 관리자에게 나타나는 한 가지 문제점은 KC000 서버가 다른 시스템으로부터 받는 모든 통신 요구를 어떻게 공정하게 처리하는 가에 대한 것임

니다. 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 관리자는 추가 통신 서브시스템을 작성하여 각 등급의 판매 대리점(소형, 중형 또는 대형)이 AS의 지원을 요구하여 일반적으로 더 나은 응답을 받을 수 있도록 할 수 있습니다. 각 서브시스템 설명에 대하여 서브시스템 속성, 사전시작 작업 항목, 통신 작업 항목 및 라우팅 항목의 주문에 따라 맞춤으로 써, 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(DQCMN) 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(DQCMN) PGM(QCNTEDDM)
```

관련 개념

통신 관리 PDF

작업 관리

TCP/IP 서브시스템

다플트로 DRDA TCP/IP 연결에 사용되는 DDM TCP/IP 서버 사전시작 작업은 QUSRWRK 서브시스템에서 실행됩니다. QUSRWRK는 사용자 작업 서브시스템입니다. 이는 사용자 대신 작업하기 위해 서버에 의해 시작되는 작업을 포함합니다. 사전시작 작업으로 작업을 디스패치하는 DRDA 리스너 작업은 QSYSWRK에서 실행됩니다.

독립형 보조 기억장치 풀의 사용자 데이터베이스

사용자는 서버에 독립형 보조 기억장치 풀(IASP)을 구성하여 iSeries 서버에 추가로 관계형 데이터베이스를 작성할 수 있습니다. 각 독립형 보조 기억장치 풀 그룹은 관계형 데이터베이스입니다.

이 주제에서 독립형 보조 기억장치 풀 그룹을 사용자 데이터베이스라고 합니다. 이들은 독립형 보조 기억장치 풀 그룹 디스크에 존재하는 모든 데이터베이스 오브젝트로 이루어집니다. 또한 독립형 보조 기억장치 풀이 연결된 iSeries 서버의 시스템 관계형 데이터베이스(이 책에서는 시스템 데이터베이스라고 함)에 있는 모든 데이터베이스 오브젝트는 논리적으로 사용자 관계형 데이터베이스에 포함됩니다. 그러나 확약 제어 측면에서 시스템 데이터베이스는 다르게 취급됩니다.

방금 언급한 확약 제어 고려사항으로 부과되는 것 외에도 사용자 데이터베이스 작성과 사용에 관한 수많은 규칙이 있습니다. 한 가지 예는 사용자가 현재 스텝에 대한 사용자 데이터베이스(보조 기억장치 풀(ASP) 그룹)로 설정된 어플리케이션 리퀘스터(AR)에서 데이터베이스로 연결하는 데 APPC(Advanced Program-to-Program Communication)로 보호되는 분산 작업 단위(DUW) 대화를 사용할 수 없다는 것입니다. 다른 예는 사용자

데이터베이스에서 작성된 임의의 스키마 이름은 그 사용자 데이터베이스 또는 연관된 시스템 데이터베이스에 미리 존재해서는 안됩니다. 이러한 제한사항에 대한 자세한 정보는 SQL 참조 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 디렉토리를 일정 기간 동안 사용할 수 없게 하며 이는 DRDA 어플리케이션 리퀘스터 또는 어플리케이션 서버(AS)가 디렉토리를 사용하려고 시도를 지연되게 하거나 시간종료되게 할 수 있다는 점을 유념해야 합니다. 복수의 데이터베이스가 동시에 연결변환되면 데이터베이스의 연결변환으로 사용불가능이 발생하기 때문에 디렉토리 조작 시간종료에 훨씬 더 많이 노출되게 됩니다. 여기에 설명한 대로 사용자 데이터베이스가 처음 연결변환될 때 서버는 이 데이터베이스에 대한 디렉토리 항목을 추가하려고 시도합니다. 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 RDB명이 네트워크 속성에 지정된 시스템명과 일치할 수 없다는 것입니다(DSPNETA(네트워크 속성 표시) 명령으로 표시된 것과 같습니다).

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.

관련 개념

어플리케이션 CRG IP 주소 관리

트랜잭션 및 확약 제어 문제점 해결

『관계형 데이터베이스 디렉토리 사용』

i5/OS 라이선스 프로그램은 관계형 데이터베이스 디렉토리를 사용하여 iSeries 서버에서 실행하는 어플리케이션이 액세스할 수 있는 관계형 데이터베이스명을 정의하고, 연결할 때 SNA 또는 IP를 사용하는지를 지정하며, 이들 관계형 데이터베이스명을 해당 네트워크 매개변수와 연관시킵니다.

관련 참조

DSPNETA(네트워크 속성 표시) 명령

SQL 참조

관계형 데이터베이스 디렉토리 사용

i5/OS 라이선스 프로그램은 관계형 데이터베이스 디렉토리를 사용하여 iSeries 서버에서 실행하는 어플리케이션이 액세스할 수 있는 관계형 데이터베이스명을 정의하고, 연결할 때 SNA 또는 IP를 사용하는지를 지정하며, 이들 관계형 데이터베이스명을 해당 네트워크 매개변수와 연관시킵니다.

관계형 데이터베이스 디렉토리를 사용하면 어플리케이션 리퀘스터(AR)가 어플리케이션의 관계형 데이터베이스명을 승인하고 이 이름을 적절한 IP(Internet Protocol) 주소 또는 호스트 이름과 포트 또는 적절한 시스템 네트워크 구조 네트워크 ID와 통신 처리용 LU(Local Unit) 이름 값으로 변환할 수 있습니다. V5R2에서 RDB 디렉토리는 사용자의 우선 아웃바운드 연결 보안 메커니즘을 지정하는 데도 사용됩니다. 또한, 관계형 데이터베이스 디렉토리는 ARD 프로그램을 관계형 데이터베이스명과 연관시킬 수 있도록 합니다.

분산 관계형 데이터베이스 네트워크의 각 iSeries 시스템은 관계형 데이터베이스 디렉토리가 구성되도록 해야 합니다. 한 시스템에는 오직 하나의 관계형 데이터베이스 디렉토리가 존재합니다. 분산 관계형 데이터베이스 네트워크의 각 AR은 해당 로컬 관계형 데이터베이스에 대한 관계형 데이터베이스 디렉토리에 항목을 가지고 있어야 하며 AR이 액세스하는 각 리모트 및 로컬 사용자 관계형 데이터베이스에 대해서도 마찬가지입니다. 어플리케이션 서버(AS)로만 활동하는 분산 관계형 데이터베이스 네트워크의 시스템은 해당 디렉토리에 다른 리모트 관계형 데이터베이스의 관계형 데이터베이스명을 포함할 필요가 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 DSPNETA(네트워크 속성 표시) 명령으로 표시되는 현재 시스템명이 됩니다.

관련 참조

DSPNETA(네트워크 속성 표시) 명령

관계형 데이터베이스 디렉토리에 대한 작업

관계형 데이터베이스 디렉토리에 대해 작업하려면 다음 명령을 사용하십시오.

관련 참조

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 명령

DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령

RMVRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 명령

WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 명령

SNA에 대한 항목 추가 사용법: 다음은 ADDRDBDIRE(RDB 디렉토리 항목 추가) 화면입니다. 이 화면의 프롬프트를 사용하거나 ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령을 사용하여 관계형 데이터베이스 디렉토리에 항목을 추가할 수 있습니다.

ADDRDBDIRE(REC 디렉토리 항목 추가)

선택사항을 입력하고 Enter 키를 누르십시오.

```

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

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. RDB(관계형 데이터베이스명) 매개변수와 RMTLOCNAME(리모트 위치명) 매개변수는 ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령에 필요합니다. 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. 이는 데이터베이스명을 식별하고 이를 사용자의 분산 관계형 데이터베이스 네트워크에 있는 특정 서버에 상관시키는 데 도움을 줄 수 있습니다(특히 네트워크가 복잡한 경우).

이 명령의 다른 선택적 매개변수를 보려면 ADDRDBDIRE(RDB 디렉토리 항목 추가) 화면에서 F10을 누르십시오. 이들 선택적 매개변수는 다음과 같습니다.

ADDRDBDIRE(REC 디렉토리 항목 추가)

선택사항을 입력하고 Enter 키를 누르십시오.

```

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

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 추가 ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령 매개변수에 대한 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 Communication)로 디폴트를 사용하는 경우 서버는 사용할 장치, 로컬 위치 및 리모트 네트워크 ID를 결정합니다. 네트워크 속성에 정의된 모드명을 사용하며 분산 관계형 데이터베이스 구조(DRDA) 지원에 대한 트랜잭션 프로그램 이름을 사용합니다. 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.

ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령에서 이들 디폴트값을 변경할 수 있습니다. 예를 들어, DB2 Universal Database for VM 서버와 통신하도록 TNSPGM 매개변수를 변경해야 하는 수도 있습니다. DB2 Universal Database for VM 지원에 대한 디폴트로, TNSPGM은 사용자가 연결하려는 DB2 Universal Database for VM 데이터베이스의 이름입니다. DRDA(*DRDA)에 대한 디폴트 TNSPGM 매개변수 값은 X'07F6C4C2'입니다. QCNTEDDM과 DB2DRDA도 X'07F6C4C2'로 맵핑됩니다. 트랜잭션 프로그램명에 대한 자세한 정보는 해당 관련 항목을 참조하십시오.

관련 태스크

211 페이지의 『DB2 Universal Database for VM 어플리케이션 리퀘스터의 TPN으로 QCNTSRVC 설정』

TPN 태그 내에 QCNTSRVC를 지정하기 위해 UCOMDIR NAMES 파일을 변경하십시오.

관련 참조

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

210 페이지의 『iSeries용 DB2 Universal Database 어플리케이션 리퀘스터의 TPN으로 QCNTSRVC 설정』

ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 또는 CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 명령의 TNSPGM 매개변수에 QCNTSRVC를 지정하십시오.

211 페이지의 『DB2 Universal Database for z/OS 어플리케이션 리퀘스터의 TPN으로 QCNTSRVC 설정』

SYSIBM LOCATIONS 표를 갱신하여 iSeries용 DB2 Universal Database 어플리케이션 서버의 RDB-NAME을 수록하고 있는 행에 대한 TPN 열에 QCNTSRVC를 지정하십시오.

211 페이지의 『Linux, UNIX 및 Windows용 DB2 Universal Database 어플리케이션 리퀘스터의 TPN으로 QCNTSRVC 설정』

Linux, UNIX 및 Windows용 DB2 Universal Database에 대해 작업 중이며 이 제품군에서 TPN을 설정하는 방법에 대해 알고 싶으면 해당 웹 페이지를 참조하십시오.

TCP/IP에 대한 항목 추가 사용법: 여기에 표시된 ADDRDBDIRE(RDB 디렉토리 항목 추가) 화면은 RMTLOCNAME 매개변수의 두 번째 요소로 *IP를 입력할 경우 패널이 어떻게 변경되는지 및 TCP/IP를 사

용하는 RDB에 대해 일반적인 항목이 무엇과 같이 표시되는지를 예시합니다. 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 Enter 키를 누르십시오.

관계형 데이터베이스 . . . . . > MP311
관계형 데이터베이스 별명 . . . . . > *NONE
리모트 위치:
이름 또는 주소 . . . . . > MP311.spiffy.com

. . . . .

유형 . . . . . > *IP          *SNA, *IP
텍스트 . . . . .
. . . . . > 'Oak Street Dealership'

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

관계형 데이터베이스 별명 지정: 다음 예에서는 RDB 별명을 지정하는 디렉토리 항목을 추가합니다. 그러면 같은 이름의 관계형 데이터베이스가 있는 네트워크는 분산 관계형 데이터베이스 구조(DRDA) 환경에서 각 관계형 데이터베이스를 고유하게 식별할 수 있습니다. 별명을 사용하는 항목이 RDB 디렉토리에 추가되면 항목은 별명으로 식별됩니다. 항목을 표시하거나 삭제하려면 별명 이름을 지정해야 합니다.

다음 화면은 RDBALS를 관계형 데이터베이스 별명으로 지정한 것입니다.

```

선택사항을 입력하고 Enter 키를 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 Dealership'
    
```

WRKRDBDIRE 및 옵션 1을 사용하여 별명에 대한 항목을 추가할 때 먼저 실제 RDB명을 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.

RMVRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 명령을 사용하여 관계형 데이터베이스 항목을 제거할 때 실제 관계형 데이터베이스명 대신 별명을 사용하여 제거할 항목을 지정합니다.

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

RMTLOCNAME에 대해 MP311.spiffy.com을 지정하는 대신 IP 주소를 지정할 수 있다는 점을 주지하십시오(예: '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 구조(IPSec)를

사용하는 전송에 대해 포트 447을 구성할 수 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. 특정 위치에 있는 DRDA 포트에 대해 유효한 서비스명을 정의한 경우 숫자 대신 이를 사용할 수도 있습니다. 하지만 iSeries에서는 *DRDA가 'drda' 서비스명을 사용하는 것보다 우선됩니다.

어플리케이션 리퀘스터 드라이버에 대한 항목 추가: ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령 프롬프트에 통신 정보 및 어플리케이션 리퀘스터 드라이버(ARD) 프로그램을 지정하려면 F9 및 PgDn 키를 누르십시오. ARD 프로그램이 ADDRDBDIRE 명령에 지정된 통신 정보를 사용하지 않을 경우(일반적인 경우임) RMTLOCNAME 매개변수에 특수 값 *ARDPGM을 사용하십시오. ARD 프로그램은 시스템 데이터베이스의 라이브러리에 상주해야 합니다(ASP 번호 1-32).

관련 참조

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

WRKRDBDIRE 명령 사용: RDB 디렉토리 항목에 대한 작업 화면은 관계형 데이터베이스 디렉토리 항목을 추가, 변경, 표시 또는 제거하기 위한 옵션을 제공합니다.

RDB 디렉토리 항목에 대한 작업

위치 대상
 옵션을 입력한 후 Enter 키를 누르십시오.
 1=추가 2=변경 4=제거 5=세부사항 표시 6=세부사항 인쇄

옵션	관계형 데이터베이스	위치	리모트 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. RMRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 명령을 사용하면 특정 관계형 데이터베이스명, 총칭명, 모든 디렉토리 항목 또는 리모트 항목만을 지정하는 옵션이 제공됩니다.

관계형 데이터베이스 디렉토리 항목에 대한 작업 화면에는 항목의 세부사항을 표시하는 항목이 있습니다. 관계형 데이터베이스 항목에 대한 작업 화면의 출력은 화면으로 출력됩니다. 그러나 DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령을 사용하면 출력을 프린터 또는 출력 파일로 보낼 수 있습니다. 관계형 데이터베이스 디렉토리는 iSeries 오브젝트가 아니므로, 출력 파일의 사용은 관계형 데이터베이스 디렉토리에 대하여 백업 수단을 제공합니다. 관계형 데이터베이스 디렉토리를 백업하기 위한 출력 파일에 대해 DSPRDBDIRE 명령을 사용하는 데 대한 자세한 정보는 관계형 데이터베이스 디렉토리 저장 및 복원을 참조하십시오.

RDB 디렉토리 항목 화면에 대한 작업에 관한 옵션을 갖고 관계형 데이터베이스 디렉토리에 있는 항목을 변경합니다. CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 명령을 사용하여 디렉토리의 항목을 변경

할 수도 있습니다. 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 변경되면, RMVRMTJRN(리모트 저널 제거) 명령 또는 QjoRemoveRemoteJournal API를 사용하여 리모트 저널을 제거한 후 ADDRMTJRN(리모트 저널 추가) 명령 또는 QjoAddRemoteJournal API를 사용하여 다시 추가해야 합니다. 리모트 위치 유형 또는 인증 또는 그 외 다른 항목이 변경된 경우, CHGRMTJRN(리모트 저널 변경) 명령 또는 QjoChangeJournalState API를 사용하여 리모트 저널링을 종료한 후 CHGRMTJRN(리모트 저널 변경) 명령 또는 QjoChangeJournalState API를 사용하여 다시 시작해야 합니다. 분산 파일에 사용된 변경사항을 가져오려면 노드 그룹을 삭제하고 다시 작성한 후 해당 파일을 다시 작성해야 합니다.

관련 태스크

159 페이지의 『관계형 데이터베이스 디렉토리 저장 및 복원』

관계형 데이터베이스 디렉토리는 iSeries 오브젝트가 아닙니다. 대신 IPL 시간에 서버가 여는 파일로 구성됩니다.

관련 참조

ADDRMTJRN(리모트 저널 추가) 명령

관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 명령

CHGRMTJRN(리모트 저널 변경) 명령

DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령

RMVRMTJRN(리모트 저널 제거) 명령

RMVRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 명령

***LOCAL 디렉토리 항목:** *LOCAL이 들어 있는 디렉토리 항목은 디렉토리에서 하나 뿐인 고유한 항목이며 로컬 시스템 데이터베이스의 이름을 지정합니다. 연관된 RDB명을 SQL문 CONNECT TO x(여기서 x는 로컬 서버명임)에서 로컬 데이터베이스에 연결하는 데 사용할 수 있습니다. CONNECT TO x(여기서 x는 로컬 서버명임)는 SQL문 CONNECT RESET를 사용하는 것과 동일한 효과를 가집니다.

프로그램 테스트와 같은 목적으로 로컬 서버 데이터베이스에 DRDA 연결을 하려는 경우 ME 및 MYSELF와 같은 두 개의 특수한 RDB 이름을 사용할 수 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. 그러나 이러한 RDB명을 일반적으로 사용하는 것은 바람직하지 않으며 일부 상황에서 이것을 사용하면 예상치 않은 작동이 발생할 수 있다는 것을 경고하기 위해서만 언급되었습니다.

로컬 RDB 항목의 이름을 변경해야 하는 경우 이 프로시저에는 제거 및 추가 조작의 수행이 포함됩니다. 그러나 로컬 항목에는 일부 시스템 차원의 DRDA 속성 정보가 들어 있으므로 이 로컬 항목을 제거하는 데는 특별한 고려사항이 있습니다. 항목을 제거하려고 시도하면 CPA3E01 메세지(*LOCAL 디렉토리 항목을 제거하거나 변경하면 구성 자료를 유실할 수 있습니다(C G))가 표시되므로 이 조작을 취소(C)하거나 계속(G)할 수 있는 기회가 주어집니다. 메세지 텍스트는 계속해서 해당 항목이 CHGDDMTCPA(DDM TCP/IP 속성 변경)

명령으로 입력된 구성 자료를 저장하는 데 사용되었음을 알려줍니다. *LOCAL 항목을 제거하면 구성 자료가 손상될 수 있으므로 디폴트 구성 자료 값이 사용됩니다. 디폴트 값이 만족스럽지 못하면, 구성 자료는 CHGDDMTCPA 명령으로 재입력되어야 합니다. 항목을 제거하기 전에 CHGDDMTCPA 명령에 지정된 값을 기록하여 *LOCAL 항목을 삭제하고 올바른 로컬 RDB명의로 추가한 후 이들 값을 복원할 수 있도록 할 수 있습니다.

관련 참조

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

독립형 보조 기억장치 풀의 사용자 데이터베이스에 대한 디렉토리 항목: 데이터베이스가 하나 뿐인 서버(즉, 독립형 보조 기억장치 풀(ISAP)이 구성되지 않은)에서 *LOCAL 항목은 단일 로컬 데이터베이스를 나타냅니다. 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 대한 설명은 어플리케이션 CRG IP 주소 관리 주제를 참조하십시오. 이 경우, IP 주소는 리모트 위치 필드에서 사용됩니다.

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.

관련 참조

어플리케이션 CRG IP 주소 관리

예: 관계형 데이터베이스 디렉토리 설정

Spiffy사 네트워크는 관계형 데이터베이스 디렉토리가 분산 관계형 데이터베이스 네트워크의 서버에서 사용되는 방법을 설명하고 각 디렉토리를 설정하는 방법을 보여주는 예를 제공합니다. 이 예는 통신에 APPC(Advanced Program-to-Program Communication)를 사용하는 것으로 가정하며, 이는 TCP/IP에 반대되며 설정하기가 더 간편합니다.

그러나, 예의 일부 요소는 프로토콜 의존성입니다. APPC 사용에 필요한 RDB 디렉토리는 TCP/IP 네트워크에도 필요하지만 해당 매개변수는 다릅니다. 호스트명이나 IP 주소 및 포트 식별은 LU명, 장치 설명, 모드, TPN 등을 대체합니다.

고려해야 할 단순 관계는 아래 그림에 보여주는 두 지역 오피스 간의 관계입니다.

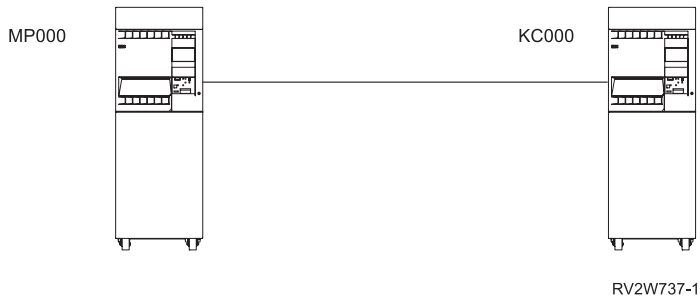


그림 6. 두 서버에 대한 관계형 데이터베이스 디렉토리 설정

각 지역 오피스에 대한 관계형 데이터베이스 디렉토리는 로컬 관계형 데이터베이스에 대한 항목과 리모트 관계형 데이터베이스에 대한 항목을 포함해야 하는데, 이는 각 서버가 어플리케이션 리퀘스터(AR)이면서 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')
```

앞의 예에서 MP000 서버는 RMTLOCNAME 매개변수에 *LOCAL을 지정함으로써 자신을 로컬 관계형 데이터베이스로 식별합니다. iSeries 서버에는 하나의 관계형 데이터베이스만이 존재합니다. RDB 디렉토리에 있는 관계형 데이터베이스 이름을 서버 이름과 동일하게 하여 네트워크 관계형 데이터베이스의 식별을 간소화할 수 있습니다. 로컬 위치에 대한 항목은 로컬 서버명과 동일한 이름을 가질 수 있으며, 로컬 위치명에 대한 항목은 리모트 서버명과 동일한 이름을 가질 수 있습니다.

주: 서버명은 CHGNETA(네트워크 속성 변경) 명령의 SYSNAME 매개변수에 지정됩니다. 로컬 서버는 통신 구성 중 CHGNETA 명령의 LCLLOCNAME 매개변수에서 식별됩니다. SNA(APPC)를 사용하는 리모트 위치는 통신 구성 중 (CRTCTLAPPC)(제어기 설명 작성)(APPC) 명령의 RMTCPNAME 매개변수로 식별됩니다. 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, 다음 그림에서와 같이 네트워크에서 관계형 데이터베이스에 액세스하려면 MP000 서버에 대한 관계형 데이터베이스 디렉토리가 해당 각 판매 대리점에 대한 항목을 포함하도록 확장되어야 합니다.

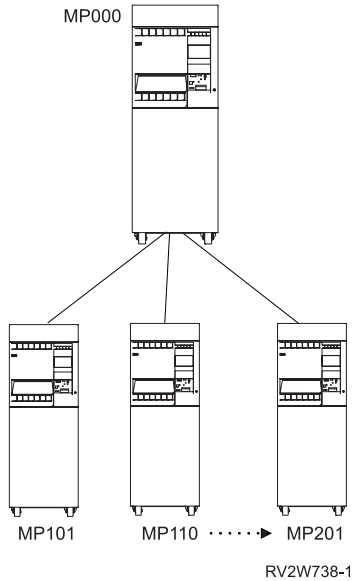


그림 7. 다중 서버에 대한 관계형 데이터베이스 디렉토리 설정

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

```
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
```

앞의 예에서 각 지역 판매 대리점은 미네아폴리스 관계형 데이터베이스 디렉토리에 리모트 관계형 데이터베이스로서 포함되어 있습니다.

각 판매 대리점은 MP000 및 기타 판매 대리점 어플리케이션 서버에 대해 AR의 역할을 할 수 있으므로 이들은 자신에 대한 항목을 로컬 관계형 데이터베이스로 가지며 지역 오피스 및 다른 모든 판매자를 리모트 관계형 데이터베이스로 갖는 관계형 데이터베이스 디렉토리를 가지고 있어야 합니다. 데이터베이스 관리자는 각 판매 대리점 서버에서 관계형 데이터베이스 디렉토리를 작성하는 여러 옵션을 갖고 있습니다.

가장 시간이 많이 들고 오류가 생기기 쉬운 방법은 MP000 분산 관계형 데이터베이스 네트워크를 이루고 있는 모든 서버에서 각 디렉토리 항목을 작성하기 위해 ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령을 사용하여 각 서버에 관계형 데이터베이스 디렉토리를 작성하는 것입니다.

더 좋은 다른 방법은 MP000에 대한 위의 예에서 보여주는 것과 같이 제어 언어(CL) 프로그램을 작성하는 것입니다. 분산 관계형 데이터베이스 관리자는 각 판매 대리점 서버를 위해 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 관리자는 사용자 정의된 CL 프로그램을 각 판매 대리점에 분배하여 해당 서버에서 실행되고 고유 관계형 데이터베이스 디렉토리를 빌드하게 할 수 있습니다.

세 번째 방법은 DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령을 사용한 결과로서 출력 파일로 송신된 관계형 데이터베이스 디렉토리 정보를 읽는 프로그램을 작성하는 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. 그런 다음 CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 명령을 사용하여 MP000 서버 디렉토리를 로컬 서버용으로 사용자 정의할 수 있습니다.

관련 태스크

159 페이지의 『관계형 데이터베이스 디렉토리 저장 및 복원』

관계형 데이터베이스 디렉토리는 iSeries 오브젝트가 아닙니다. 대신 IPL 시간에 서버가 여는 파일로 구성됩니다.

관련 참조

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 명령

CRTCTLAPPC(제어기 설명 작성)(APPC) 명령

DSPNETA(네트워크 속성 표시) 명령

DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령

DRDA 보안 설정

분산 관계형 데이터베이스 구조(DRDA) 보안은 보안 주제에서 다루어지고 있지만 완벽한 설명을 위해 이 주제에서 DRDA를 사용하기 전 또는 APPC에서 TCP/IP 사용으로 네트워크를 변환할 때 고려해야 할 사항으로 언급되었습니다.

TCP/IP에 대한 보안 설정은 APPC에 요구되는 것과는 상당한 차이가 있습니다. 한 가지 유념해야 할 것은 APPC가 가지고 있는 보안 위치 개념이 결여되었다는 것입니다. TCP/IP 서버는 클라이언트 서버가 자신이 누구라고 말하는 것을 완전히 신뢰할 수 없으므로 연결 요구 시 암호 사용은 더욱 중요합니다. 연결 요구 시 암호 송신을 보다 쉽게 하기 위해 TCP/IP 지원과 함께 특정 사용자 프로파일과 연관된 서버 권한 부여 리스트의 사용에 대한 내용을 소개했습니다. 서버 권한 부여 리스트의 항목은 56 페이지의 『보안』에 설명된 xxxSVRAUTHE 명령(여기서 xxx는 ADD, CHG 및 RMV를 나타냄)을 사용하여 유지보수할 수 있습니다. 서버 권한 항목 대신 사용할 수 있는 방법은 SQL CONNECT문의 USER/USING 형식을 사용하여 연결 요구 시 암호를 송신하는 것입니다.

TCP/IP를 사용하는 경우 Kerberos 지원은 다른 보안 옵션을 제공합니다. 네트워크 인증 서비스는 Kerberos 프로토콜을 지원하며 Kerberos에 대한 구성에 사용될 수 있습니다.

서버측의 설정은 인바운드 연결 요구에 대해 어떤 레벨의 보안이 필요한지를 결정하고 지정하는 것을 포함합니다. 예를 들어, 암호화되지 않은 암호를 받아들여야 합니까? 디폴트 설정은 그렇다는 것입니다. CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령을 사용하여 디폴트 설정을 변경할 수 있습니다.

관련 개념

네트워크 인증 서비스 구성

제어 언어(CL)

관련 참조

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

DRDA에 대한 TCP/IP 서버 설정

TCP/IP 프로토콜을 사용할 분산 관계형 데이터베이스 구조(DRDA) 어플리케이션 서버(AS)가 있는 경우, DDM TCP/IP 서버를 설정해야 합니다.

TCP/IP 서버를 설정하는 것은 필요할 때 시작되는지 확인하는 것 만큼이나 간단할 수 있으며, 이는 항상 활성 상태로 남아 있게 할 경우 다음 명령을 실행하여 수행할 수 있습니다.

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

그러나 환경에 맞게 서버를 조정하기 위해 조정할 수 있는 기타 매개변수가 있습니다. 이것은 초기의 사전시작 작업의 수, 작업의 최대수 및 더 많은 작업을 시작할 때에 임계값 등을 포함합니다.

사용자는 모든 클라이언트가 연결할 때 사용할 공통 사용자 프로파일을 설정하거나 서로 다른 클래스의 리모트 사용자를 위한 여러 보안 레벨의 여러 사용자 프로파일 세트를 설정하려 할 수 있습니다. 그런 다음 어플리케이션 리퀘스터(AR)에서 ADDSVRAUTE(서버 인증 항목 추가) 명령을 사용하여 AR에 있는 각 사용자의 프로파일명을 AS에서 이들이 실행될 해당 사용자 프로파일로 맵핑할 수 있습니다.

관련 개념

65 페이지의 『TCP/IP 네트워크의 어플리케이션 리퀘스터 보안』

연결 시나리오에 따라 다른 레벨의 인증을 사용해야 합니다. 따라서 관리자는 각 RDB 디렉토리 항목에서 우선 인증 방법 필드를 설정하여 어플리케이션 서버(AS)에 연결할 때 어플리케이션 리퀘스터(AR)에 필요한 최하위 보안 인증 방법을 설정할 수 있습니다.

132 페이지의 『TCP/IP 서버 관리』

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

관련 참조

ADDSVRAUTE(서버 인증 항목 추가) 명령

대화식 SQL에 대한 SQL 패키지 설정

이 주제는 비iSeries 어플리케이션 서버(AS)에만 적용됩니다.

다음 항목 중 하나가 참일 경우 SQL 패키지가 서버에 작성되었는지 확인해야 합니다.

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

STRSQL iSeries 서버 SQL 패키지를 요구하지 않습니다. 일반적으로 SQL 패키지는 STRSQL 사용자를 위해 비iSeries 어플리케이션 서버(AS)에 작성됩니다. 그러나, STRSQL에 대한 초기 연결이 로컬 서버와 이루어지고, 이 연결이 두 현상의 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 읽기전용이 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. 그런 다음, 리모트 서버로 연결될 수 있으며 이는 첫 번째 연결이므로 갱신할 수 있습니다.

대화식 SQL을 시작할 때 *NONE이 아닌 다른 확약 제어 레벨을 지정해야 합니다. 또한, 사용자가 연결에 사용하는 사용자 ID는 어플리케이션 서버에서 SQL 패키지를 작성하기 위한 적당한 권한을 갖고 있어야 합니다. 연결 시도 시 42501의 SQLSTATE가 표시되면 사용자가 패키지 작성 권한이 없는 것입니다.

관련 참조

188 페이지의 『대화식 SQL 특정 연결 실패』

대화식 SQL에서 CONNECT문을 실행할 때 이따금 일반 SQ30080 메시지가 나타납니다.

DDM 파일 설정

iSeries 서버에서 DRDA 지원 구현은 통신을 위해 분산 자료 관리(DDM) 대화를 사용합니다. 따라서 사용자는 분산 관계형 데이터베이스 처리에 관련하여 DDM을 사용할 수 있습니다.

사용자는 DDM을 사용하여 어플리케이션 서버(AS)에 리모트 명령을 제출하고, 한 iSeries 서버에서 다른 서버로 표를 복사하며, 다른 서버에서 비분산 관계형 데이터베이스 작업을 처리할 수 있습니다.

분산 관계형 데이터베이스를 사용할 경우 어플리케이션 리퀘스터(AR)가 데이터베이스에 연결하는 데 필요한 정보가 관계형 데이터베이스 디렉토리에 제공됩니다. DDM을 사용할 경우 어플리케이션 서버(AS)에서 작업할 각 파일에 대해 별도의 DDM 파일을 작성해야 합니다. 어플리케이션 리퀘스터(AR)의 어플리케이션은 이 DDM 파일을 사용하여 어플리케이션 서버(AS)에 있는 리모트 파일 및 어플리케이션 서버(AS)에 대한 통신 경로를 식별합니다.

또, V5R2에서는 RDB 디렉토리 항목에 대한 참조를 가진 DDM 파일을 작성할 수도 있습니다. 분산 관계형 데이터베이스 관리에서 언급된 일부 데이터베이스 관리 타스크는 DDM을 사용하여 리모트 파일에 액세스합니다. DDM 파일은 CRTDDMF(분산 자료 관리 파일 작성) 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.

관련 개념

113 페이지의 『관리』

사용자는 분산 관계형 데이터베이스의 관리자로서, 여러 서버에서 실행하는 작업에 대해 책임이 있습니다.

분산 자료 관리

122 페이지의 『리모트 iSeries 서버 조작』

사용자는 분산 관계형 데이터베이스의 관리자로서 리모트 iSeries 서버를 조작해야 하는 경우가 있습니다.

관련 참조

CRTDDMF(분산 자료 관리 파일 작성) 명령

분산 관계형 데이터베이스의 표에 자료 로드

분산 관계형 데이터베이스 환경의 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.

분산 관계형 데이터베이스의 표에 새 자료 로드

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

SQL을 사용하여 표에 자료 로드:

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 SPIFFY라는 SQL 컬렉션에 있는 INVENT라는 표에 한 행의 자료를 삽입합니다.

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

이 예에서 지역 센터는 SQL 어플리케이션을 사용하여 어플리케이션 서버(AS)에 있는 표에 자료를 로드합니다. i5/OS 라이선스 프로그램에 SQL에 대한 런타임 지원이 제공되므로 AS는 iSeries용 IBM DB2 조회 관리자 및 SQL 개발 킷 라이선스 프로그램이 필요하지 않습니다. 그러나, iSeries용 IBM DB2 조회 관리자 및 SQL 개발 킷 라이선스 프로그램은 어플리케이션을 기록하는 데 요구됩니다.

관련 개념

SQL 프로그래밍

관련 참조

SQL 참조

iSeries 조회 관리 기능을 사용하여 표 및 파일의 자료 조작:

i5/OS 라이선스 프로그램은 iSeries용 DB2 Universal Database 조회 관리 기능을 제공하여 표 및 파일에 있는 자료를 조작할 수 있도록 합니다. 조회는 SQL 조회 명령문을 사용하여 작성됩니다.

사용자는 CL 명령이나 사용자의 어플리케이션 프로그램에서 조회 호출가능 인터페이스를 통해 조회를 실행할 수 있습니다. 조회 관리 기능을 사용하여 다음과 같이 45 페이지의 『SQL을 사용하여 표에 자료 로드』에 설명된 재고 갱신사항에 대한 표에 한 행의 자료를 삽입할 수 있습니다.

소스 실제 파일 INVLOAD와 SQL 명령문에 소스 멤버 INVLOAD를 작성하십시오.

```
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 Y (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. 이와 같은 이유로 DB2 UDB for iSeries 조회 관리 기능은 표에 자료를 로드하는 데 있어서 SQL 어플리케이션 만큼 효율적이지 않습니다. 그러나 어플리케이션을 작성하려면 iSeries용 IBM DB2 조회 관리자 및 SQL 개발 킷 제품이 필요하며 SQL 및 조회 관리에 대한 런타임 지원은 i5/OS 라이선스 프로그램의 일부입니다.

관련 개념

조회 관리 프로그래밍 PDF

자료 파일 유틸리티를 사용하여 자료 입력, 표 갱신 및 조회 작성:

자료 파일 유틸리티(DFU)는 IBM에서 제공하는 iSeries 어플리케이션 개발 툴(ADT) 패키지의 일부로서 자료를 입력하고 표를 갱신하며 조회를 작성하는 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 데 데이터베이스 갱신 프로그램을 작성할 수 있다는 것입니다. 사용자는 DDM 파일로 DFU를 사용하거나 표시장치 pass-through를 사용하여 어플리케이션 서버(AS)에서 DFU를 실행함으로써 리모트 서버에 있는 자료에 대해 작업할 수 있습니다.

관련 개념

ADTS/400: 자료 파일 유틸리티 PDF

한 iSeries 서버에서 다른 iSeries 서버로 자료 이동

전사적 작업에서 한 iSeries 서버에서 iSeries 서버로 자료를 이동해야 하는 여러 가지 상황이 발생합니다.

iSeries 서버에서 다른 서버로 자료를 이동해야 하는 상황의 예는 한 지역에서 새 판매 대리점이 문을 열고, 클라이언트 주소에 따라 하나 또는 두 개의 다른 판매 대리점에서 일부 클라이언트가 새 판매 대리점으로 이전될 수 있는 경우입니다. 아마도 판매 대리점이 문을 닫거나 더 이상 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 Universal Database for iSeries 조회 관리 기능
- 테이프 장치로 또는 테이프 장치에서 복사

- DDM으로 파일 명령 복사
- 네트워크 파일 명령
- iSeries 서버 저장 및 복원 명령

사용자 작성 어플리케이션 프로그램 작성:

분산 작업 단위(DUW) 연결 관리로 컴파일된 프로그램은 리모트 데이터베이스 및 로컬 데이터베이스로 연결하여 한 데이터베이스에서 FETCH하여 다른 데이터베이스로 INSERT함으로써 자료를 이동시킬 수 있습니다.

복수 행 FETCH 및 복수 행 INSERT를 사용하여 레코드 블록을 한 번에 처리할 수 있습니다. 확약 제어를 사용하면 자료 이동 중 지점마다 체크 포인트를 수행하여 실패할 경우 복사를 처음부터 다시 시작하지 않게 할 수 있습니다.

대화식 SQL을 사용하여 데이터베이스 조회:

SQL SELECT문 및 대화식 SQL을 사용하여 로컬 서버에서 표를 작성하거나 갱신하는 데 필요한 자료에 대해 다른 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 사용하여 조회 결과를 로컬 서버의 데이터베이스 파일에 배치할 수 있습니다. 대화식 SQL 프로세스에 대해 확약 제어 레벨이 지정된 경우, 이는 Application Server(AS)에 적용되며 로컬 서버의 데이터베이스 파일은 확약 제어 레벨 *NONE 아래에 존재합니다.

대화식 SQL을 사용하면 다음을 수행할 수 있습니다.

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

KC105 판매 대리점이 부품 번호 1234567의 전체 재고를 KC110으로 전송하는 상황을 고려해 보십시오. KC110은 KC105로부터 확보한 부품에 대한 KC105 데이터베이스를 조회합니다. 이 재고 조회의 결과는 KC110 서버에 이미 존재하는 데이터베이스 파일로 리턴됩니다. 이것은 이 타스크를 완료하기 위해 사용자가 사용할 수 있는 프로세스입니다.

대화식 SQL 화면을 가져오려면 STRSQL(SQL 시작) 명령을 사용하십시오. 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 Enter 키를 누르십시오. 다음 화면이 나타납니다.

파일 변경

선택사항을 입력하고 Enter 키를 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 지정된 경우 다음 예에서와 같이 대화식 SQL 처리를 시작할 수 있습니다.

SQL 명령문 입력

SQL 명령문을 입력하고 Enter 키를 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	

관련 개념

SQL 프로그래밍

관련 참조

SQL 참조

DB2 Universal Database for iSeries 조회 관리 기능을 사용하여 리모트 서버 조회:

DB2 Universal Database for iSeries 조회 관리 기능은 리모트 서버를 조회하고 출력 파일의 결과를 로컬 서버로 리턴하는 데 대화식 SQL과 거의 같은 지원을 제공합니다.

대화식 SQL과 조회 관리 기능은 모두 파일 또는 표에 대해 자료 조작 작업(INSERT, DELETE, SELECT 등)을 수행할 수 있으며 이 때 표(또는 파일)가 컬렉션에 이미 존재할 필요는 없습니다(라이브러리에 존재할 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 조회 관리 기능을 시작하기 전에 OVRDBF(데이터베이스 파일로 대체) 명령을 사용하여 다른 멤버를 지정하는 경우를 제외하고는 첫 번째 파일에 넣어집니다.

관련 개념

조회 관리 프로그래밍 PDF

관련 참조

OVRDBF(데이터베이스 파일로 대체) 명령

테이프 또는 테이프에서 파일 복사:

테이프 또는 테이프에서 파일을 복사하는 데 사용할 수 있는 명령은 여러 가지가 있습니다.

iSeries 서버에서 CPYTOTAP(테이프 복사) 명령을 사용하여 표 또는 파일을 테이프에 복사할 수 있습니다.

테이프의 자료는 CPYFRMTAP(테이프에서 복사) 명령을 사용하여 다른 서버에 로드할 수 있습니다. 이 명령의 사용에 대한 자세한 방법은 기억장치 솔루션 주제를 참조하십시오.

CPYF(파일 복사) 명령을 사용하여 테이프의 자료를 iSeries용 DB2 Universal Database로 로드할 수도 있습니다. 이 방법은 DB2 Universal Database for z/OS 또는 DB2 Universal Database Server for VM(SQL/DS™)에서 언로드된 자료를 로드할 때 특히 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 *NULLFLAGS 옵션이 지정된 CPYF 명령은 널(null) 플래그를 인식하고 테이프의 인접 필드에 있는 자료를 무시하며 iSeries용 DB2 UDB에 널(null) 필드를 작성할 수 있습니다. IBM 주 프레임으로부터 자료를 반입하기 위한 또 하나의 유용한 FMTOPT 매개변수 값은 *CVTFLOAT입니다. 이는 S/390 형식으로 테이프에 저장된 부동 소수점 자료를 iSeries용 DB2 UDB에서 사용되는 IEEE 형식으로 변환될 수 있도록 합니다.

관련 개념

기억장치 솔루션

관련 참조

CPYTOTAP(테이프에 복사) 명령

CPYFRMTAP(테이프에서 복사) 명령

CPYF(파일 복사) 명령

파일 복사 명령을 사용하여 iSeries 서버 간에 자료 이동:

하나의 iSeries 서버로부터 다른 서버로 자료를 이동시키는 또 하나의 방법은 DDM으로 파일 복사 명령을 사용하여 자료를 복사하는 것입니다.

CPYF(파일 복사), CPYSRCF(소스 파일 복사) 및 CPYFRMQRYP(조회 파일에서 복사) 명령을 사용하여 소스 및 어플리케이션 서버(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 닫을 경우, 분산 관계형 데이터베이스 관리자는 리모트 서버에서 로컬 지역 서버로 고객 및 재고 표를 복사할 수 있습니다. 관리자는 해당 표에 액세스하고 복사하려면 어플리케이션 서버(AS)에 적절한 권한이 있는 사용자 프로파일이 있어야 하며 복사되는 각 표나 파일에 대해 어플리케이션 리퀘스터(AR)에 DDM 파일을 작성해야 합니다. 다음의 예는 데이터베이스 관리자가 KC105라는 리모트 위치명을 가지는 서버로부터 KC000라는 지역 센터 서버로 SPIFFY라는 컬렉션에 INVENT라는 명령표를 복사하는데 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. iSeries 서버는 같은 작업에 대해 어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS)가 될 수 없으므로 위의 명령을 실행하는 데 SBMRMTCMD(리모트 명령 제출) 명령을 사용할 수 없습니다.

DDM으로 이 명령을 사용할 때 다음의 항목을 고려하십시오.

- CPYF(파일 복사) 명령 및 CPYSRCF(소스 파일 복사) 명령의 FROMFILE 및 TOFILE 매개변수에는 DDM 파일을 지정할 수 없습니다.

주: CYFRMQRYP(조회 파일에서 복사) 및 CPYFRMTAP(테이프에서 복사) 명령의 경우, TOFILE 매개변수에만 DDM 파일명을 지정할 수 있으며, CPYTOTAP(테이프 복사) 명령의 경우 FROMFILE 매개변수에만 DDM 파일명을 지정할 수 있습니다.

- 삭제 기능(delete-capable) 파일이 비삭제기능 파일로 복사될 때 사용자는 COMPRESS(*YES)를 지정하여야 하며, 그렇지 않으면 오류 메시지가 송신되고 작업이 종료됩니다.

- DDM 파일의 리모트 파일명이 멤버명을 지정할 경우, CPYF(파일 복사) 명령에서 이 파일에 대해 지정된 멤버명은 DDM 파일의 리모트 파일명에 있는 멤버명과 같아야 합니다. 또한 OVRDBF(데이터베이스 파일로 대체) 명령은 DDM 파일의 리모트 파일명에 있는 멤버명과 다른 멤버명을 지정할 수 없습니다.
- DDM 파일이 멤버명을 지정하지 않은 경우와 OVRDBF(데이터베이스 파일로 대체) 명령이 이 파일에 대해 멤버명을 지정하는 경우, CPYF(파일 복사) 명령은 OVRDBF 명령에 지정된 멤버명을 사용합니다.
- TOFILE 매개변수가 존재하지 않는 파일을 참조하는 DDM 파일일 경우 CPYF는 이 파일을 작성합니다. CPYF(파일 복사) 명령으로 작성된 리모트 파일에 대해서는 다음과 같은 특별한 고려사항을 유념하십시오.
 - 목표 DDM 작업에 대한 사용자 프로파일은 어플리케이션 서버(AS)에서 CRTPF(실제 파일 작성) 명령을 수행할 수 있는 권한이 있어야 합니다.
 - iSeries 목표에서 TOFILE 매개변수는 파일 관리 주제에 설명된 속성을 제외하고는 FROMFILE 매개변수의 모든 속성을 가집니다.
- TCP/IP 사용 시, CRTDDMF(분산 자료 관리 파일 작성) 명령의 RMTLOCNAME 매개변수의 두 번째 요소는 *IP가 되어야 합니다.

관련 개념

분산 자료 관리

관련 참조

CPYF(파일 복사) 명령

CPYSRCF(소스 파일 복사) 명령

CPYFRMQRYP(조회 파일에서 복사) 명령

CPYFRMTAP(테이프에서 복사) 명령

CPYTOTAP(테이프에 복사) 명령

CRTPF(실제 파일 작성) 명령

CRTDDMF(분산 자료 관리 파일 작성) 명령

OVDBF(데이터베이스 파일로 대체) 명령

SBMRMTCMD(리모트 명령 제출) 명령

네트워크 파일 명령을 사용하여 네트워크를 통해 자료 전송:

시스템 네트워크 구조(SNA) 분배 서비스(SNADS)를 지원하는 네트워크 프로토콜을 통해 자료를 전송할 수 있습니다. 분산 관계형 데이터베이스 처리에 사용되는 APPC 및 APPN 프로토콜 외에도 SNADS는 BSCEL(Binary Synchronous Equivalence Link) 및 SNUF(SNA Upline Facility) 프로토콜과 함께 사용될 수 있습니다.

SNADS가 지원하는 iSeries 서버는 SNDNETF(네트워크 파일 송신) 명령을 사용하여 다른 서버로 자료를 송신하고 RCVNETF(네트워크 파일 수신) 및 WRKNETF(네트워크 파일에 대한 작업) 명령을 사용하여 다른 서버에서 네트워크 파일을 수신할 수 있습니다.

관련 참조

RCVNETF(네트워크 파일 수신) 명령

SNDNETF(네트워크 파일 송신) 명령

WRKNETF(네트워크 파일에 대한 작업) 명령

서버 저장 및 복원 명령을 사용하여 표 이동:

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

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

- SAVLIB(라이브러리 저장) 명령은 하나 이상의 컬렉션 또는 라이브러리를 저장합니다.
- SAVOBJ(오브젝트 저장) 명령은 하나 이상의 오브젝트(데이터베이스 표 및 뷰 포함)를 저장합니다.
- SAVCHGOBJ(변경된 오브젝트 저장) 명령은 컬렉션 또는 라이브러리가 마지막 저장된 이후 또는 지정된 날짜로부터 변경된 모든 오브젝트를 저장합니다.
- RSTLIB(라이브러리 복원) 명령은 컬렉션 또는 라이브러리를 복원합니다.
- RSTOBJ(오브젝트 복원) 명령은 하나 이상의 오브젝트(데이터베이스 표 및 뷰 포함)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 SAVLIB(라이브러리 저장) 명령을 사용하여 컬렉션을 저장하거나 서버 A에서 SAVOBJ(오브젝트 저장) 명령을 사용하여 표를 저장하십시오.
2. 자료를 SNADS를 사용하여 분배될 수 있는 저장 파일에 저장하는지 또는 테이프에 저장하는지를 지정하십시오.
3. 저장 파일을 서버 B에 분배하거나 테이프를 서버 B로 송신하십시오.
4. 서버 B에서 RSTLIB(라이브러리 복원) 명령을 사용하여 컬렉션을 복원하거나 서버 B에서 RSTOBJ(오브젝트 복원) 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 있는 사용자는 이 서버에서 원래 소유자의 프로파일을 작성하고 복원된 오브젝트 소유권을 변경하거나, 로컬 및 리모트 사용자들 모두에 대하여 이 오브젝트에 대한 새로운 권한을 지정해야 합니다.

관련 개념

백업 및 회복

관련 참조

라이브러리 복원(RSTLIB) 명령

RSTOBJ(오브젝트 복원) 명령

SAVCHGOBJ(변경된 오브젝트 저장) 명령

SAVLIB(라이브러리 저장) 명령

SAVOBJ(오브젝트 저장) 명령

비iSeries 서버에서 iSeries 서버로 데이터베이스 이동

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

다른 IBM 서버에서 자료 이동:



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

여기에 나열된 방법을 사용하여 자료를 이동할 수 있습니다.

- 고급 언어(HLL) 프로그램은 또다른 서버에서 자료를 추출하기 위해 작성될 수 있습니다. 서버의 통신 프로그램은 이 서버로 자료를 로드하는데 사용될 수 있습니다.
- 다른 분산 관계형 데이터베이스 구조(DRDA) 구현을 지원하는 서버의 경우, SQL 기능을 사용하여 자료를 이동시킬 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.
- 다른 서버에 있는 표 및 파일에서 자료를 추출하여 통신 회선을 통하거나 테이프를 iSeries 서버로 송신할 수 있습니다.
 - DB2 Universal Database for z/OS 데이터베이스에서 데이터베이스 관리자와 함께 제공되는 DSNTIAUL 이라는 샘플 프로그램을 사용하여 파일 또는 표에서 자료를 추출할 수 있습니다.
 - DB2 Universal Database Server for VM(SQL/DS) 데이터베이스에서 데이터베이스 관리자의 데이터베이스 서비스 유틸리티 부분은 자료를 추출하는 데 사용할 수 있습니다.
 - DB2 Universal Database for z/OS 및 DB2 Universal Database Server for VM databases, Data Extract(DXT™)는 둘 다 자료를 추출하는 데 사용할 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.
- 표준 테이프 관리 기술을 사용하여 DB2 Universal Database for z/OS 또는 DB2 Universal Database Server for VM 데이터베이스에서 테이프를 복사할 수 있습니다. iSeries 서버는 CPYFRMTAP(테이프에서 복사) 명령을 사용하여 테이프에서 자료를 로드합니다. 그러나 CPYF(파일 복사) 명령은 IBM 메인프레임 컴퓨터에서 자료를 마이그레이트하는 특별한 지원을 제공합니다. OVRTAPF(테이프 파일로 대체) 명령을 사용하여 테이프 자료에 대해 CPYF를 사용할 수 있습니다. OVRTAPF 명령을 사용하면 iSeries 서버 이외의 다른 서버에서 자료를 가져올 때 필요할 수도 있는 특별한 테이프 특정 매개변수를 지정할 수 있습니다.

특별한 CPYF 지원을 사용하면 널(null) 가능한 자료 및 부동 소수점 자료를 가져올 수 있습니다. 널 가능한 자료는 단일 바이트 플래그가 각 널 가능한 필드와 연관될 수 있는 방법으로 주 프레임에서 언로드될 수 있습니다. FMTOPT 매개변수에 대해 *NULLFLAGS 옵션을 지정할 경우 CPYF(파일 복사) 명령은 널 플래그를 인식하고 테이프에 있는 인접 필드의 자료를 무시하며 iSeries용 DB2 Universal Database에서 필드를 널로 만들 수 있습니다. IBM 주 프레임에서 자료를 반입하는 데 유용한 다른 FMTOPT 매개변수 값은 *CVTFLOAT입니다. 이는 S/390 형식으로 테이프에 저장된 부동 소수점 자료를 iSeries용 DB2 UDB에서 사용되는 IEEE 형식으로 변환될 수 있도록 합니다.

iSeries 서버에서 테이프 장치 사용에 대한 자세한 방법은 기억장치 솔루션 주제를 참조하십시오.

- 통신 회선을 통해 송신된 자료는 iSeries 서버에서 SNADS 지원을 통해 처리될 수 있습니다. SNADS 지원은 분산 관계형 데이터베이스 처리에 사용되는 APPN(Advanced Peer-to-Peer Networking) 또는 APPC(Advanced Program-to-Program Communication) 프로토콜 외에 BSCCL 및 SNUF 프로토콜에 대해서도 네트워크 파일을 전송합니다.
 - TSO XMIT 기능을 사용하여 MVS™ 시스템에서 iSeries 서버로 자료를 송신할 수 있습니다. 서버는 WRKNETF(네트워크 파일에 대한 작업) 또는 RCVNETF(네트워크 파일 수신) 명령을 사용하여 네트워크 파일을 수신합니다.
 - VM 시스템에서, 자료는 SENDFILE 기능을 사용하여 서버로 송신될 수 있습니다. 서버는 WRKNETF(네트워크 파일에 대한 작업) 또는 RCVNETF(네트워크 파일 수신) 명령을 사용하여 네트워크 파일을 수신합니다.
- Microsoft® Windows에서 클라이언트 자료는 별도로 주문되는 IBM 제품인 iSeries Access를 사용하여 iSeries 서버로 송신될 수 있습니다.
- 다양한 워크스테이션 클라이언트에서 Linux, UNIX 및 Windows용 DB2 Universal Database IMPORT 및 EXPORT 유틸리티를 사용하여 iSeries 서버로 자료를 복사하고 또 복사해올 수 있습니다. IMPORT는 자료를 기존 포로만 가져올 수 있습니다. IMPORT 및 EXPORT 유틸리티에 대한 예는 Advanced Functions 및 Administration on DB2 Universal Database for iSeries, SG24-4249-03  레드북을 참조하십시오. 이 레드북에는 IMPORT 및 EXPORT 유틸리티와 함께 사용할 수 있는 파일 유형 및 자료 형식에 대한 정보도 있습니다.
- 또한, 비동기식 통신과 같이 SNADS를 지원하지 못하는 통신 회선을 통해 자료를 송신할 수도 있습니다. i5/OS 라이선스 프로그램의 일부인 FTS(파일전송 지원) 유틸리티는 자료를 송신하고 수신하는 데 사용될 수 있습니다. 통신 및 통신 파일에 대한 작업의 자세한 정보는 ICF 프로그래밍  을 참조하십시오.

관련 개념

분산 자료 관리

기억장치 솔루션

관련 참조

CPYFRMTAP(테이프에서 복사) 명령

CPYF(파일 복사) 명령

OVRDBF(데이터베이스 파일로 대체) 명령

RCVNETF(네트워크 파일 수신) 명령

20 페이지의 『추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계』

분산 관계형 데이터베이스는 로컬 데이터베이스에서 전용으로 사용하기 위해 개발한 어플리케이션과는 다른 요구사항을 갖고 있습니다.

WRKNETF(네트워크 파일에 대한 작업) 명령

비IBM 서버에서 자료 이동:

다른 서버로부터 파일 또는 표를 테이프에 복사하고 이들 파일을 iSeries 서버에 로드할 수 있습니다.

이를 수행하려면 CPYFRMIMPF(가져오기 파일에서 복사) 명령을 사용하십시오.

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

로컬 및 광역 네트워크 둘 다에 대한 피어 투 피어 연결 기능은 TCP/IP에 의해 제공됩니다. iSeries TCP/IP Connectivity Utilities/400 라이선스 프로그램의 FTP(파일 전송 프로토콜) 기능은 사용자가 리모트 서버의 기능에 따라 많은 유형의 파일을 수신할 수 있게 합니다.

OSIFS/400(OSI File Services/400) 라이선스 프로그램은 OSI(개방 서버 상호연결) 네트워크에 대해 파일 관리 및 전송 서비스를 제공합니다. 전제조건 라이선스 프로그램인 OSI Communications Subsystem/400이 있는 OSIFS/400은 iSeries 서버를 OSI 파일 전송, 액세스 및 관리(FTAM) 표준에 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. 자세한 정보는 *OSI 통신 서브시스템 프로그래밍 개념 안내서*를 참조하십시오.

관련 개념

TCP/IP 설정

관련 참조

CPYFRMIMPF(가져오기 파일에서 복사) 명령

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 콜렉션, 표 및 기타 관계형 데이터베이스 오브젝트에 대한 적절한 권한을 가져

야 합니다. 이는 분산 데이터베이스 사용자가 네트워크에서 사용하는 데이터베이스에 대한 유효한 사용자 프로파일을 가져야 한다는 의미입니다. 보안 계획은 네트워크에서 사용자와 어플리케이션 프로그램 요구를 고려해야 합니다.

분산 관계형 데이터베이스 관리자는 해결해야 할 두 가지 보안 문제점을 갖고 있습니다.

- 시스템간 보호
- 리모트 사이트에서의 사용자 식별

둘 이상의 시스템이 서로의 데이터베이스에 액세스하도록 설정된 경우, 통신 회선의 다른 쪽이 의도된 위치이고 침입자가 아닌지 확인하는 것이 중요합니다. 리모트 관계형 데이터베이스에 대한 DRDA 액세스의 경우, iSeries 서버에서 APPC(Advanced Peer-to-Peer Networking) 및 APPN(Advanced Peer-to-Peer Networking) 통신 구성 기능을 사용하면 이 네트워크 레벨 보안을 수행할 수 있는 옵션이 제공됩니다.

분산 관계형 데이터베이스 관리자에 대한 두 번째 중요성은 자료 보안이 해당 자료를 저장하는 시스템에 의해 유지보수되는 것입니다. 분산 관계형 데이터베이스에서 사용자는 데이터베이스가 로컬이든 리모트이든 시스템의 보안 레벨에 따라 데이터베이스에 액세스할 수 있도록 올바른 권한을 부여받아야 합니다. 분산 관계형 데이터베이스 네트워크 사용자는 어플리케이션 서버(AS)에서 실행하는 모든 작업에 대해 AS의 사용자 ID로 올바르게 식별되어야 합니다. APPC/APPN 및 TCP/IP 통신 프로토콜을 모두 사용하는 분산 관계형 데이터베이스 구조(DRDA) 지원은 연결 요구와 함께 사용자 ID 및 암호 송신을 제공합니다.

이 주제에서는 통신에 관련된 보안 주제 및 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.

관련 참조

24 페이지의 『분산 관계형 데이터베이스에 대한 보안 고려사항』

분산 관계형 데이터베이스 계획에 분산 자료의 보안에 대해 실행해야 하는 결정 사항이 일부 포함되어 있습니다.

분산 관계형 데이터베이스 보안의 요소

분산 관계형 데이터베이스 관리자는 네트워크에서 어플리케이션 리퀘스터(AR)의 자료 액세스를 불필요하게 제한하지 않으면서 네트워크에 있는 어플리케이션 서버의 자원을 보호해야 합니다.

AR은 권한이 있는 사용자만 분산 관계형 데이터베이스 프로그램에 액세스할 수 있도록 해당 오브젝트 및 관계형 데이터베이스를 보호합니다. 이는 사용자를 식별하고 각 사용자(또는 사용자 그룹)가 오브젝트에 대해 무엇을 수행하도록 허용되는지 지정하는 정상 iSeries 서버 오브젝트 권한 부여를 사용하여 수행됩니다. 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 사용자만이 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 네트워크에서 iSeries 서버에 대한 보안 계획의 몇 가지 양상은 다음을 포함합니다.

- 기밀 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
- 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(Advanced Program-to-Program Communication) 대화에서 시스템이 보안 레벨 10을 사용할 때 iSeries 서버는 비보안 시스템으로서 네트워크에 연결됩니다. 서버는 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 있는지 여부를 제어합니다. 예를 들어, AR에서 보안 레벨이 10으로 설정되고 AS에서 보안 레벨이 10 이상인 경우, 시스템 중 하나에서 보안 요소를 변경하지 않으면 세션이 설정되지 않고 적절한 정보가 송신되지 않을 수 있습니다.

DRDA 액세스에 대한 암호

리모트 사용자에게 데이터베이스 액세스 권한을 부여하는 가장 일반적인 방법은 연결 시간에 사용자 ID 및 암호를 제공하는 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 데이터베이스에 대한 분산 관계형 데이터베이스 구조(DRDA) 액세스에서 일단 대화가 설정되면 암호를 다시 입력할 필요가 없습니다. 리모트 작업 단위(RUW) 연결 관리 메소드로 실행할 때 RELEASE, DISCONNECT 또는 CONNECT문을 사용하여 연결을 종료하면, 어플리케이션 리퀘스터(AR) 작업 속성 및 사용자가 연결된 AS의 유형에 따라 첫 번째 어플리케이션 서버(AS)와의 대화가 제거되거나 제거되지 않을 수 있습니다(특정 규칙에 대해서는 DDM 대화 제어를 참조하십시오). 첫 번째 AS에 연결된 대화가 제거되지 않

으면, 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.

관련 개념

보안

관련 참조

APPC, APPN 및 HPR

124 페이지의 『DDM 대화 제어』

이 주제에서 연결이란 용어는 SQL 연결의 개념을 나타냅니다. SQL 연결은 명시성 또는 내재성의 SQL CONNECT를 행하는 시간부터 논리 SQL 연결이 SQL DISCONNECT와 같은 수단에 의해 종료되거나 RELEASE 다음에 COMMIT이 이어질 때까지 지속됩니다.

APPC 네트워크의 보안 요소

분산 관계형 데이터베이스 구조(DRDA)를 사용할 때 DRDA 환경에 있는 각 서버의 자료 자원을 보호해야 합니다.

DRDA 환경에 있는 각 서버의 자료 자원을 보호하기 위해 다음 매개변수로 제어되는 세 가지 그룹의 보안 요소를 사용할 수 있습니다.

- 시스템 관련 보안이나 세션에 대해서는 각 iSeries 서버에 *LOCPWD* 매개변수를 사용하여 소스와 목표 시스템 간에 APPC 세션이 처음 설정될 때 소스와 목표 시스템 간에 시스템 유효성 확인 암호를 교환하는 것을 나타냅니다. 두 시스템 모두 세션에 시작되기 전에 같은 암호를 교환해야 합니다. (System/36™에서 이 암호는 위치 암호라고 불립니다.) APPC 네트워크에서 CRTDEVAPPC(장치 설명 작성)(APPC) 명령에 있는 *LOCPWD* 매개변수가 이 암호를 지정합니다. 장치는 APPN을 사용하여 자동으로 작성되고 리모트 위치 리스트의 위치 암호는 신원을 확인하기 위해 이들 두 위치에서 사용되는 암호를 지정합니다. 유형 (*APPNRMТ)의 리모트 위치 리스트를 작성하려면 CRTCFGL(구성 리스트 작성) 명령을 사용하십시오.
- 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 CRTDEVAPPC(장치 설명 작성)(APPC) 명령의 *SECURELOC* 매개변수는 로컬 서버가 리모트 서버의 보안 검증 기능을 허용하는지 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 프로파일

자세한 정보는 APPC 네트워크의 DDM 소스 시스템 보안 주제를 참조하십시오.

- 오브젝트 관련 보안의 경우, CHGNETA(네트워크 속성 변경) 명령에 DDMACC 매개변수를 사용하여 iSeries 서버에 있는 파일을 다른 서버가 전부 액세스할 수 있는지 여부와 액세스할 수 있는 경우 들어오는 요구를 어떤 보안 레벨에서 검사할 것인지를 나타냅니다. 이 오브젝트 관련 매개변수에 대한 자세한 정보는 DDM 네트워크 속성(DDMACC 매개변수) 주제에서 제공됩니다.
 - DDMACC 매개변수에 *REJECT를 지정하면 목표 iSeries 서버가 수신하는 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 보안에 대한 DDM 서버 액세스 제어 종료 프로그램 주제를 참조하십시오.)
 - 파일이 DRDA를 사용하여 목표 서버에 작성될 때, 지정된 라이브러리명이 이 파일을 포함합니다. DRDA 요구시 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.

관련 개념

추가 보안에 대한 DDM 서버 액세스 제어 종료 프로그램

62 페이지의 『APPC 네트워크의 DRDA 어플리케이션 서버 보안』

목표 서버가 iSeries 서버일 때, 리모트 파일 액세스 요구가 허용되는지 여부를 판별하기 위해 여러 가지 요소를 함께 사용합니다.

관련 참조

CHGNETA(네트워크 속성 변경) 명령

CRTCFGL(구성 리스트 작성) 명령

CRTDEVAPPC(장치 설명 작성)(APPC) 명령

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 CRTCFGL(구성 리스트 작성) 명령으로 작성되며, 여기에는 모든 리모트 위치, 해당 위치 암호 및 리모트 위치 보안 여부의 리스트가 포함됩니다. 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.

CHGCFGL(구성 리스트 변경) 명령을 사용하여 리모트 구성 리스트를 변경할 수 있지만 해당 위치에 대한 모든 장치가 모두 단절변환 상태가 된 후에야 변경됩니다.

DSPCFGL(구성 리스트 표시) 명령을 사용할 때는 암호가 있다는 표시가 나타나지 않습니다. CHGCFGL(구성 리스트 변경) 명령은 암호가 입력된 경우 필드에 *PASSWORD를 넣어 암호가 존재한다는 것을 표시합니다. 암호를 표시하는 방법이 없습니다. 위치 보안을 설정하는 데 문제점이 있는 경우 두 시스템에 암호를 다시 입력하여 암호가 서로 일치하는지 확인해야 합니다.

관련 참조

APPC, APPN 및 HPR

CHGCFGL(구성 리스트 변경) 명령

CRTCFGL(구성 리스트 작성) 명령

DSPCFGL(구성 리스트 표시) 명령

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 암호화된 상태로 송신될 경우에만 사용자 ID와 암호가 모두 통신용으로 송신됩니다. 그렇지 않으면 오류가 보고됩니다. i5/OS의 DRDA는 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 프로파일을 대화에 사용할 수 있는지는 지정된 서비스

템에 대한 ADDCMNE(통신 항목 추가) 명령 또는 CHGCMNE(통신 항목 변경) 명령의 DFTUSR 매개변수에 지정된 값에 따라 결정됩니다. DFTUSR 매개변수의 *NONE 값은 어플리케이션 서버(AS)가 목표에 있는 디폴트 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 *VFYENCPWD일 경우, 사용자 프로파일 및 연관된 암호는 암호의 값을 기밀로 유지하도록 암호를 암호화한 후 리모트 서버로 송신되므로, 사용자는 DRDA를 사용하려면 두 서버 모두에 같은 사용자 프로파일 이름과 암호를 가져야 합니다.

주: SECURELOC(*VFYENCPWD)는 대부분의 정보가 리모트 서버에 의해 확인되므로 이들 세 가지 옵션 중 가장 안전합니다. 그러나 이 옵션을 사용하려면 사용자가 여러 서버에서 같은 암호를 유지보수해야 합니다. 이는 사용자가 한 서버에서 암호를 변경하면서 동시에 다른 서버를 갱신하지 않을 경우 문제가 생길 수 있습니다.

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.

관련 참조

ADDCMNE(통신 항목 추가) 명령

CHGCMNE(통신 항목 변경) 명령

APPC 네트워크의 DRDA 어플리케이션 서버 보안:

목표 서버가 iSeries 서버일 때, 리모트 파일 액세스 요구가 허용되는지 여부를 판별하기 위해 여러 가지 요소를 함께 사용합니다.

사용자 관련 보안 요소

사용자 관련 보안 요소는 목표 서버의 SECURELOC 매개변수, 소스 서버가 송신한 사용자 ID(허용된 경우), 소스 사용자가 송신한 사용자 ID의 암호 및 목표 서버의 사용자 프로파일 또는 디폴트 사용자 프로파일을 포함합니다.

오브젝트 관련 보안 요소

오브젝트 관련 보안 요소는 DDMACC 매개변수 및 선택적으로 정상적인 오브젝트 권한 제어를 보충하기 위해 사용자가 제공하는 사용자 종료 프로그램을 포함합니다.

목표 보안의 사용자 관련 요소

분산 관계형 데이터베이스 작업을 처리하려면 어플리케이션 서버(AS)에 유효한 사용자 프로파일이 존재해야 합니다. 사용자는 iSeries 서버에서 통신 작업을 처리하는 서브시스템에 대하여 디폴트 사용자 프로파일을 지정할 수 있습니다.

디폴트 사용자 프로파일명은 AS에서 ADDCMNE(통신 항목 추가) 명령의 DFTUSR 매개변수에 지정됩니다. ADDCMNE 명령은 통신 작업에 사용되는 서브시스템 설명에 통신 항목을 추가합니다.

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

- SECURELOC 또는 AS의 보안 위치에 대해 *YES가 지정되었으면 AS는 어플리케이션 리퀘스터(AR)를 보안 위치로 간주합니다. 해당 요구에 대해 AR로부터 사용자 ID 및 이미 확인된 인디케이터가 예상됩니다. 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.

다음 표는 iSeries 서버에서 SNA SECURITY(PGM)를 제어하는 요소의 가능한 모든 조합을 보여줍니다. 모든 열의 "Y"는 요소가 존재하거나 조건이 충족됨을 표시합니다. PWD 열의 "M"은 보안 관리자가 사용자 암호를 검색하고 암호 보호가 사용 중이면 보호(암호화된) 암호를 송신하는 것을 표시합니다. 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 지원하고 암호가 i5/OS 또는 OS/400 버전 2 릴리스 2 이상에서 작성된 경우에만 사용될 수 있습니다.

표 4. 분산 관계형 데이터베이스에 대한 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

표 4. 분산 관계형 데이터베이스에 대한 리모트 액세스 (계속)

행	UID	PWD ¹	AVI	SEC(Y)	DFT	유효	액세스
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

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

유효 사용자 ID와 암호는 유효합니다

UID 사용

공급된 사용자 ID로 연결

DFT 사용

디폴트 사용자 ID로 연결

거부 연결되지 않음

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 않게 하려면 분산 관계형 데이터베이스 오브젝트에 액세스해야 하는 모든 AR 사용자를 위해 AS에 사용자 프로파일을 작성하십시오. 그러나, 디폴트 사용자 프로파일을 사용하기로 결정하는 경우, 사용자가 정식 권한없이 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 SBS(D(SAMPLE) DEV(*ALL) DFTUSER(QUSER)
```

TCP/IP 네트워크의 보안 요소

원시 TCP/IP를 통한 DDM 및 DRDA는 APPC와 연관된 통신 장치, 모드, 보안 위치 속성 및 대화 보안 레벨과 같은 i5/OS 통신 보안 서비스 및 개념을 사용하지 않습니다. 그러므로, TCP/IP에 대한 보안 설정은 상당한 차이가 있습니다.

TCP/IP 네트워크의 어플리케이션 리퀘스터 보안:

연결 시나리오에 따라 다른 레벨의 인증을 사용해야 합니다. 따라서 관리자는 각 RDB 디렉토리 항목에서 우선 인증 방법 필드를 설정하여 어플리케이션 서버(AS)에 연결할 때 어플리케이션 리퀘스터(AR)에 필요한 최하위 보안 인증 방법을 설정할 수 있습니다.

관리자는 또한 하위 보안 인증 방법을 허용하도록 선택하여 서버와 협상할 인증 방법에 대한 결정을 허용할 수도 있습니다. 이 경우 우선 인증 방법이 계속 시도되지만 AS가 이 우선 인증 방법을 승인할 수 없으면 서버 보안 설정 및 암호 지원의 가용성과 같은 다른 요인에 따라 하위 방법을 사용할 수 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) 측에서 두 가지 방법 중 하나를 사용하여 DRDA TCP/IP 연결 요구에 대해 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 다음 예에서와 같이 SQL CONNECT문의 USER/USING 형식을 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.

다음 예는 CLI를 사용하는 프로그램에서 사용자 ID 및 암호가 DRDA 어플리케이션 리퀘스터(AR)에 대해 호스트 변수로 제공되는 방법입니다.

```
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. 리스트는 디폴트로 비어 있지만 ADDSVRAUTE(서버 인증 항목 추가) 명령을 사용하여 항목을 추가할 수 있습니다. 사용자가 TCP/IP를 통한 DRDA 연결을 시도할 때, iSeries용 DB2 Universal Database 클라이언트(AR)는 클라이언트 작업을 실행하는 사용자 프로파일에 대한 서버 권한 리스트를 검사합니다. CONNECT문의 RDB명과 권한 항목의 SERVER 이름(대문자이어야 함)이 일치할 경우 해당 항목의 연관된 USRID 매개변수가 연결 사용자 ID에 사용됩니다. PASSWORD 매개변수가 항목에 저장되어 있는 경우, 연결 요구 시 이 암호도 역시 송신됩니다.

서버 권한 항목은 DDM 파일 I/O 조작을 위해 TCP/IP를 통해 암호를 송신하는 데도 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.

ADDSVRAUTE(서버 인증 항목 추가) 명령을 사용하여 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')
```

다음 예는 RDB 디렉토리 항목을 사용할 때 ADDSVRAUTE(서버 인증 항목 추가) 명령의 구문을 보여줍니다.

```
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. RDB 디렉토리를 사용하도록 작성하지 않은 DDM 파일을 사용하는 경우는 예외이며 이 때에는 SERVER 매개변수에 QDDMSERVER를 지정해야 합니다. RDB명을 지정할 때 반드시 대문자로 지정해야 합니다. USRID 매개변수는 서버 작업을 실행할 사용자 프로파일을 지정합니다. PASSWORD 매개변수는 사용자 프로파일에 대한 암호를 지정합니다.

USRPRF 매개변수를 생략하면 이는 디폴트로 ADDSVRAUTE(서버 인증 항목 추가) 명령이 실행되는 사용자 프로파일로 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(서버 인증 항목 표시) 명령을 사용하여 서버 인증 리스트에 추가된 인증 항목을 판별할 수 있습니다. 사용자 작성 프로그램의 QSYRTVSE(서버 인증 항목 검색)(QsyRetrieveServerEntries) API를 사용할 수도 있습니다.

RMVSVRAUTE(서버 인증 항목 제거) 명령을 사용하여 서버 권한 항목을 제거할 수 있습니다. CHGSVRAUTE(서버 인증 항목 변경) 명령을 사용하여 서버 권한 항목을 변경할 수 있습니다.

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

Kerberos 소스 구성

분산 관계형 데이터베이스 구조(DRDA) 및 분산 자료 관리(DDM)는 두 시스템이 모두 Kerberos에 대해 구성되어 있으면 Kerberos 인증을 이용할 수 있습니다.

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. 유효한 TGT를 가지고

있으면 암호가 직접적으로 필요하지 않으므로 서버 인증 항목을 사용할 필요가 없게 됩니다. 그러나 작업의 사용자 프로파일이 유효한 TGT를 가지고 있지 않으면 서버 인증 항목으로부터 사용자 ID 및 암호를 검색하여 필요한 TGT 및 서비스 티켓을 생성할 수 있습니다.

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 맵핑할 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 그림은 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 Enter 키를 누르십시오.

F3=종료  F12=취소
  
```

다음은 리모트 위치명과 일치하는 정의역명을 보여주는 해당 krb5.conf 파일 내용의 부분입니다(주: DSPF(파일 표시) 명령을 사용하여 구성 파일 내용을 표시합니다).

```
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.

관련 개념

EIM(Enterprise Identity Mapping)

관련 태스크

43 페이지의 『DRDA에 대한 TCP/IP 서버 설정』

TCP/IP 프로토콜을 사용할 분산 관계형 데이터베이스 구조(DRDA) 어플리케이션 서버(AS)가 있는 경우, DDM TCP/IP 서버를 설정해야 합니다.

네트워크 인증 서비스

관련 참조

ADDSVRAUTE(서버 인증 항목 추가) 명령
DPSVRAUTE(서버 인증 항목 표시) 명령
QSYRTVSE(서버 인증 항목 검색)(QsyRetrieveServerEntries) API
RMVSVRAUTE(서버 인증 항목 제거) 명령
CHGSVRAUTE(서버 인증 항목 변경) 명령
DSPF(파일 표시) 명령

TCP/IP 네트워크에서 어플리케이션 서버 보안:

TCP/IP 서버는 clear-text 암호가 있는 사용자 ID의 디폴트 보안을 가지고 있습니다. 이는 서버가 설치될 때 인바운드 TCP/IP 연결 요구에 최소한 서버 작업이 실행될 사용자 ID에 수반되는 clear-text 암호가 포함되어야 한다는 것을 의미합니다.

보안은 CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령을 사용하거나 iSeries Navigator의 네트워크 → 서버 → TCP/IP → DDM 서버 등록정보에서 변경할 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 필요하지 않지만 송신되는 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.

상위 보안 레벨의 경우 암호화된 암호가 필요하거나 PWDRQD(*ENCRYPTED) 및 Kerberos 또는 PWDRQD(*KERBEROS)를 사용할 수 있습니다. Kerberos를 사용할 경우, EIM(Enterprise Identity Mapping)을 사용하여 사용자 프로파일을 Kerberos 프린시펄로 맵핑해야 합니다.

다음 예는 사용자 ID에 암호화된 암호가 수반되어야 한다는 것을 지정하기 위해 CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령을 사용하는 방법을 보여줍니다. 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. 이는 DRDA 표준 알고리즘이며 가장 최근에 릴리스된 IBM DRDA 어플리케이션 리퀘스터에서 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.

관련 개념

EIM(Enterprise Identity Mapping)

관련 참조

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

DDM 및 DRDA에 대한 연결 보안 프로토콜:

TCP/IP를 통한 분산 자료 관리(DDM) 또는 분산 관계형 데이터베이스 구조(DRDA)의 현재 iSeries용 DB2 UDB 구현은 여러가지 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. TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령은 암호화하기 전의 아웃바운드 자료 스트림과 암호 해독 후 인바운드 자료 스트림을 기록합니다.

관련 개념

202 페이지의 『통신 추적』

분산 관계형 데이터베이스 구조(DRDA)를 사용하여 분산 관계형 데이터베이스에 액세스할 때 CPF3Exx 범위 또는 CPF91xx 범위의 메시지를 얻은 경우 통신 추적을 실행해야 합니다.

관련 참조

TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령

DDM 및 DRDA에 대한 보안 소켓 층:

iSeries 분산 관계형 데이터베이스 구조(DRDA)용 DB2 Universal Database 클라이언트는 보안 소켓 층(SSL)을 지원하지 않습니다.

그러나 인터넷 프로토콜 보안 구조(IPSec)를 통해 유사한 기능을 사용할 수 있습니다.

DDM TCP/IP 서버는 SSL 자료 암호화 프로토콜을 지원합니다. 이 프로토콜을 사용하여 레코드 레벨 액세스에 대해 SSL을 지원하는 iSeries Toolbox for Java™ 및 iSeries Access 제품군 OLE DB Provider와 같은 클라이언트 및 SSL을 지원할 수 있는 독립형 소프트웨어 벤더가 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 CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령에 PWDRQD(*ENCRYPTED)를 지정하면 모든 유효한 암호를 SSL과 함께 사용할 수 있습니다. 이는 서버가 암호를 포함하여 전체 자료 스트림이 암호화되었다고 인식하기 때문에 가능합니다.

관련 개념

『DDM 및 DRDA에 대한 인터넷 프로토콜 보안 구조』

인터넷 프로토콜 보안 구조(IPSec)는 암호화 보안 서비스를 제공하는 네트워크층의 보안 프로토콜입니다. 이들 서비스는 인터넷 또는 인트라넷을 통한 기밀 자료 전달을 지원합니다.

SSL(보안 소켓 층)

관련 참조

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

필요한 프로그램:

SSL 지원을 설정하고 설치해야 합니다.

관련 개념

Windows용 iSeries Access: 프로그래밍

iSeries 서버 요구사항:

iSeries 서버가 SSL을 통해 통신하려면 V4R4 이상의 오퍼레이팅 시스템을 실행해야 하며 다음 어플리케이션이 설치되어야 합니다.

- i5용 TCP/IP 연결 유틸리티, 5722-TC1(기본 TCP/IP 지원)
- i5/OS용 IBM HTTP Server, 5722-DG1(디지털 인증 관리자 액세스용)
- 디지털 인증 관리자, 5722-SS1 - Boss 옵션 34

DDM 및 DRDA에 대한 인터넷 프로토콜 보안 구조:

인터넷 프로토콜 보안 구조(IPSec)는 암호화 보안 서비스를 제공하는 네트워크층의 보안 프로토콜입니다. 이들 서비스는 인터넷 또는 인트라넷을 통한 기밀 자료 전달을 지원합니다.

iSeries에서, IPSec 즉 VPN(가상 사설망) 지원의 구성요소를 사용하면 어플리케이션(예: DRDA 또는 DDM)에 관계없이 두 IP 주소 또는 포트 조합 사이의 모든 자료를 암호화할 수 있습니다. 사용자는 IPSec에 사용되는 주소와 포트를 구성할 수 있습니다. IBM은 DRDA 액세스 또는 DDM 액세스를 위한 IPSec에 대해 447 포트를 사용할 것을 제안합니다.

서버에서 CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령에 암호를 PWDRQD(*ENCRYPTED) 지정함으로써 생기는 요구사항은 일반적으로 IPSec와 함께 유효한 암호를 사용해도 충족되지 않습니다. 이는 어플리케이션(DRDA 또는 DDM)이 IPSec가 사용되고 있는지를 판별할 수 없기 때문입니다. 따라서, 사용자는 IPSec로 PWDRQD(*ENCRYPTED)를 사용해서는 안됩니다.

관련 개념

69 페이지의 『DDM 및 DRDA에 대한 보안 소켓 층』

iSeries 분산 관계형 데이터베이스 구조(DRDA)용 DB2 Universal Database 클라이언트는 보안 소켓 층(SSL)을 지원하지 않습니다.

VPN(가상 사설망)

『clear text로 송신되는 특정 암호에 대한 고려사항』

iSeries가 연결 암호의 암호화를 지원하지는 하지만 RDB 디렉토리 항목을 설정할 때 지정할 수 있는 연결 보안 옵션 중 하나는 *USRIDPWD입니다.

관련 참조

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

clear text로 송신되는 특정 암호에 대한 고려사항:

iSeries가 연결 암호의 암호화를 지원하지는 하지만 RDB 디렉토리 항목을 설정할 때 지정할 수 있는 연결 보안 옵션 중 하나는 *USRIDPWD입니다.

자세한 정보는 관계형 데이터베이스 디렉토리에 대한 작업의 관계형 데이터베이스 디렉토리 항목 추가 명령 및 관계형 데이터베이스 디렉토리 항목 변경 명령을 참조하십시오.

연결이 작성된 서버가 *USRIDPWD 보안 옵션을 허용할 경우 연결 암호는 암호화되지 않고 흐를 수 있습니다. V5R3에서 SQL SET ENCRYPTION PASSWORD문과 ENCRYPT 기능을 사용하면 암호가 암호화되지 않고 네트워크를 흐를 수 있습니다. 현재 자료 스트림을 암호화하는 솔루션은 두 가지가 있습니다. 한 가지 방법은 IPSec을 사용하는 것입니다. 다른 방법으로는 SSL을 지원하는 AR을 사용하는 경우 이 프로토콜을 사용하여 iSeries AS로 보내고 받는 자료를 암호화할 수 있습니다.

관련 개념

70 페이지의 『DDM 및 DRDA에 대한 인터넷 프로토콜 보안 구조』

인터넷 프로토콜 보안 구조(IPSec)는 암호화 보안 서비스를 제공하는 네트워크층의 보안 프로토콜입니다. 이들 서비스는 인터넷 또는 인트라넷을 통한 기밀 자료 전달을 지원합니다.

관련 참조

33 페이지의 『관계형 데이터베이스 디렉토리에 대한 작업』

관계형 데이터베이스 디렉토리에 대해 작업하려면 다음 명령을 사용하십시오.

SET ENCRYPTION PASSWORD문

DDM/DRDA에 대한 포트 및 포트 제한사항:

iSeries 서버 관리자는 분산 자료 관리(DDM) 통신의 보안에 대한 새로운 선택사항을 통해 사용 중인 포트를 블로킹하여 특정 통신 모드를 제한할 수 있습니다. 이 주제에서는 다음의 몇 가지 고려사항에 대해 설명합니다.

DDM 또는 DRDA TCP/IP 서버는 447 포트(잘 알려진 DDM 포트) 및 446 포트(잘 알려진 DRDA 포트) 및 448 포트(잘 알려진 SSL 포트)에서 청취합니다. iSeries용 DB2 Universal Database의 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.

- clear text 자료 스트림의 경우 446
- IPSec 암호화된 자료 스트림의 경우 447(제한됨)
- SSL 암호화된 자료 스트림의 경우 448(필수)

CFGTCP(TCP/IP 구성) 명령을 사용하여 서버에서 하나 이상의 포트를 사용하지 못하게 할 수 있습니다. 이렇게 하려면 이 명령의 TCP/IP 포트 제한사항에 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 IPSec에서만 사용하도록 구성된 경우 446을 블록킹하면 원시 TCP/IP를 통한 DDM 및 DRDA 액세스에 암호화된 자료 스트림만 사용할 수 있게 됩니다. 사용자는 447과 448을 둘 다 SSL 전용으로만 제한하도록 블록 표시할 수도 있습니다. 성능 또는 기타 이유(SSL 사용 가능 클라이언트의 현재 제한된 가용성과 같이)에 대해서는 이러한 예를 따르는 것이 비실용적일 수 있지만 이들 예는 가능한 구성을 보여주기 위해 제공됩니다.

관련 참조

CFGTCP(TCP/IP 구성) 명령

DRDA 서버 액세스 제어 종료 프로그램

분산 관계형 데이터베이스 구조(DRDA) 서버의 보안 피처는 APPC 및 TCP/IP와 함께 사용할 수 있도록 하기 위해 CHGNETA(네트워크 속성 변경) 명령의 DDMACC 매개변수의 사용을 DRDA로 확장합니다.

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.

이 보안 기능을 선택하지 않을 경우에는 보통 아무것도 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 블록킹하거나 필터링하기 위해 종료 프로그램을 사용하려면 새 DRDA 종료 프로그램을 작성하거나 기존 종료 프로그램을 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).

분산 데이터 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)로 명명되는 기능 코드는 DRDA 연결 요구가 처리되는 중임을 나타냅니다(예: DRDA 서버 액세스 제어 종료 프로그램에서 FUNC 매개변수 참조). APP(어플리케이션) 입력 매개변수는 DRDA 연결 요구 호출에 대해 '*DDM ' 대신 '*DRDA '로 설정됩니다.

DRDA에 대한 종료 프로그램을 코딩할 때 매개변수 구조의 다음 필드가 유용할 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.
- 예: DRDA 서버 액세스 제어 종료 프로그램의 호출자는 이 종료 프로그램의 SRVNAME 매개변수를 설정하거나 설정하지 않을 수 있습니다. 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 TYPEDEFN은 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 이들 코드의 처음 세 문자의 부분적인 리스트입니다(사용자는 비IBM 코드를 종료 프로그램에 사용하기 전에 먼저 이를 확인해야 합니다).

QSQ IBM iSeries용 DB2 Universal Database

DSN IBM DB2 Universal Database for z/OS

SQL Linux, Unix 및 Windows용 IBM DB2 Universal Database(이전에는 DDCS라고 함)

ARI IBM DB2 Universal Database 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

JCC IBM DB2 Universal Driver for SQLJ 및 JDBC

이 필드의 나머지는 vvrrm로 구성되며, 여기서 vv는 버전이고 rr은 릴리스이며 m은 수정 레벨입니다.

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 됩니다. 이 메시지는 사용자에게 관계형 데이터베이스에 대한 권한이 없음을 나타냅니다. 클라이언트 플랫폼에 따라 사용자에게 오류를 서로 다르게 보고할 수 있다는 점을 주지하십시오.

제한사항:

- 기능 검사가 사용자 종료 프로그램에서 발생되면, 프로그램은 같은 응답 메시지를 리턴하고 연결 시도에 실패합니다. 종료 프로그램은 iSeries용 DB2 UDB에 대해 어떠한 확약 가능한 갱신도 수행해서는 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 처리 부분에서는 *ELIGIBLE 값을 갖는 ACTGRP 매개변수가 있는 RCLACTGRP(활성 그룹 재생) 명령이 사용됩니다. 결과적으로 RCLACTGRP 명령으로 제거된 활성 그룹에 대한 사전시작 서버 작업의 잔존 연계를 사용하려고 시도하면 MCH3402 예외(프로그램이 더 이상 존재하지 않는 오브젝트 전부 또는 일부를 참조하려고 시도함)가 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).

관련 개념

76 페이지의 『DRDA에 대한 오브젝트 관련 보안』

iSeries 서버가 어플리케이션 서버(AS)인 경우, 해당 관계형 데이터베이스 표에 대한 액세스를 제어하기 위해 보안을 강제할 수 있는 오브젝트 관련 레벨은 두 가지입니다.

분산 자료 관리

관련 참조

RCLACTGRP(활성 그룹 재생) 명령

CHGNETA(네트워크 속성 변경) 명령

예: DRDA 서버 액세스 제어 종료 프로그램

이 종료 프로그램은 사용자가 ALIEN일 때를 제외하고 모든 DRDA 조작과 모든 DRDA 연결을 허용하는 PL/I 종료 프로그램의 예를 보여줍니다.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```

/*****
/*
/* 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 LENGTH PIC '99999',        /* 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;
```

DRDA에 대한 오브젝트 관련 보안

iSeries 서버가 어플리케이션 서버(AS)인 경우, 해당 관계형 데이터베이스 표에 대한 액세스를 제어하기 위해 보안을 강제할 수 있는 오브젝트 관련 레벨은 두 가지입니다.

CHGNETA(네트워크 속성 변경) 명령에 DDMACC 매개변수를 사용하면 이 서버의 표가 다른 시스템에 의해 모두 액세스될 수 있는지 여부 및 액세스될 수 있는 경우 수신 DRDA 요구가 어떤 보안 레벨에서 점검되는지를 표시할 수 있습니다.

- DDMACC 매개변수에 *REJECT를 지정하면 AS가 수신하는 모든 분산 관계형 데이터베이스 요구가 거부됩니다. 그러나 이 시스템(어플리케이션 리퀘스터(AR))은 허용되는 다른 시스템의 표에 액세스하기 위해 여전히 SQL 요구를 사용할 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. AS 작업이 실행되는 사용자 프로파일에 대한 정보는 APPC 네트워크의 DDM 보안 요소 주제를 참조하십시오.

TCP/IP 연결의 경우, 서버 작업은 초기에 QUSER하에서 실행하기 시작합니다. 사용자 ID의 유효성을 확인한 후 교환이 발생하여 해당 작업이 연결 요구에 지정된 사용자 프로파일에서 실행되도록 합니다. 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 특정의 명령을 사용하여 iSeries 서버에서 특정 파일에 액세스할 수 있는지 여부를 제어하는 데 사용될 수 있습니다.

DRDA 작업의 경우, 선택적 사용자 제공 사용자 종료 프로그램(액세스 제어 프로그램)의 이름이 DDMACC 매개변수에 지정되어 있으면 시스템은 마치 *OBJAUT가 지정된 것처럼 이 항목을 하나의 예외로 처리합니다. 사용자 작성 종료 프로그램이 DRDA 작업에서 나타내는 유일한 효과는 연결 요구를 거부하는 것입니다.

DDMACC 매개변수는 초기에 *OBJAUT로 설정되며 CHGNETA(네트워크 속성 변경) 명령을 사용하여 이전에 기술된 값 중 하나로 변경될 수 있으며 현재 값은 DSPNETA(네트워크 속성 표시) 명령으로 표시될 수 있습니다. RTVNETA(네트워크 속성 검색) 명령을 사용하여 CL 프로그램에 있는 값을 얻을 수도 있습니다.

DDMACC 매개변수 값이 변경되면 변경이 즉시 적용되더라도 이 시스템(AS)에서 시작된 새 분산 관계형 데이터베이스 작업에만 영향을 줍니다. 변경되기 전에 이 AS에서 실행하던 작업은 원래의 값을 계속 사용합니다.

관련 개념

통신 관리 PDF

72 페이지의 『DRDA 서버 액세스 제어 종료 프로그램』

분산 관계형 데이터베이스 구조(DRDA) 서버의 보안 피치는 APPC 및 TCP/IP와 함께 사용할 수 있도록 하기 위해 CHGNETA(네트워크 속성 변경) 명령의 DDMACC 매개변수의 사용을 DRDA로 확장합니다.

관련 참조

CHGNETA(네트워크 속성 변경) 명령

DSPNETA(네트워크 속성 표시) 명령

RTVNETA(네트워크 속성 검색) 명령

분산 관계형 데이터베이스 오브젝트에 대한 권한

관계형 데이터베이스 오브젝트에 대한 사용자의 권한을 부여하거나 취소하기 위해 SQL GRANT 및 REVOKE 문을 사용하거나 제어 언어(CL)인 GRTOBJAUT(오브젝트 권한 부여) 및 RVKOBJAUT(오브젝트 권한 취소) 명령을 사용할 수 있습니다.

SQL GRANT 및 REVOKE 명령문만이 패키지, 표 및 뷰에서 조작합니다. 어떤 경우에는 GRTOBJAUT 및 RVKOBJAUT를 사용하여 사용자에게 명령 및 프로그램과 같은 다른 오브젝트에 대한 권한을 부여할 필요가 있습니다.

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

대화식 SQL문의 경우 권한은 해당 명령문을 처리하는 사용자의 권한에 대해 점검됩니다. 허용된 권한은 대화식 SQL문에 사용되지 않습니다.

분산 관계형 데이터베이스 어플리케이션을 실행하는 사용자는 어플리케이션 서버(AS)에서 SQL 패키지를 실행할 수 있는 권한이 필요합니다. 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 부여 받은 경우, 패키지에 대한 처리권을 가집니다. CRTSQLPKG(SQL 패키지 작성) 명령을 사용하여 패키지를 작성한 경우 사용자에게 이 패키지에 대한 처리 권한을 부여해야 합니다. SQL 프로그램에서 또는 대화식 SQL 을 사용하여 이 명령문을 발행할 수 있습니다. 샘플 명령문은 다음과 같습니다.

```
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 부여 받은 모든 사용자에게 주는 EXECUTE 특권을 제거할 수 있습니다.

같은 사용자에게 한 번 이상 같은 특권을 부여하였다면, 이 사용자에게 대하여 해당 특권을 취소함으로써 부여 한 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. 샘플 명령문은 다음과 같습니다.

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

GRTOBJAUT(오브젝트 권한 부여) 명령을 사용하여 SQL 패키지에 대한 권한을 부여하거나 RVKOBJAUT(오브젝트 권한 취소) 명령을 사용하여 SQL 패키지에 대한 권한을 취소할 수도 있습니다.

관련 개념

SQL 오브젝트에 대한 보안

관련 참조

SQL 패키지 작성(CRTSQLPKG) 명령

GRTOBJAUT(오브젝트 권한 부여) 명령

RVKOBJAUT(오브젝트 권한 취소) 명령

94 페이지의 『분산 관계형 데이터베이스 명령문』

SQL 언어와 함께 들어 있는 명령문은 특히 분산 관계형 데이터베이스를 지원합니다.

분산 관계형 데이터베이스에 대한 허용된 권한 하에서 실행되는 프로그램

분산 관계형 데이터베이스 프로그램은 허용된 권한 하에서 실행될 수 있는데, 이는 이 프로그램이 실행되는 동안 프로그램에서 사용되는 오브젝트에 대해 프로그램 소유자의 권한을 허용한다는 것을 의미합니다. 프로그램 작성시 *SQL 사전컴파일러 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. CRTSQLxxx 명령에 OPTION(*SQL)을 지정하여 iSeries 서버에 작성한 SQL 패키지는 또한 패키지에 있는 모든 정적 SQL문에 대해 패키지 소유자의 권한을 허용합니다.

분산 관계형 데이터베이스 관리자는 DSPPGMADP(허용하는 프로그램 표시) 명령을 사용하여 어플리케이션 서버의 보안 노출을 검사할 수 있습니다. DSPPGMADP 명령은 아래와 같이 지정된 사용자 프로파일을 사용하는 프로그램 및 SQL 패키지를 표시합니다. 사용자는 명령의 결과를 프린터 또는 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 Enter 키를 누르십시오.

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

```

관련 참조

DSPPGMADP(일시적 허용하는 프로그램 표시) 명령

분산 관계형 데이터베이스의 보호 전략

어플리케이션 서버(AS)에 있는 중요한 자료를 권한이 없는 액세스로부터 보호하려면 iSeries 분산 관계형 데이터베이스의 네트워크 보안을 계획해야 합니다. 그러나 관계형 데이터베이스의 분산형 특성으로 인해, 보안 계획은 네트워크에서 자료 가용성이 불필요하게 제한되지 않도록 보장해야 합니다.

분산 관계형 데이터베이스 관리자가 결정해야 할 사항 중 하나는 네트워크에 있는 각 시스템에 대한 적절한 시스템 보안 레벨입니다. 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.

분산 관계형 데이터베이스 관리자는 또한 네트워크의 어플리케이션 리퀘스터(AR)와 어플리케이션 서버 간에 통신이 설정되는 방법도 고려해야 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. 액세스할 필요가 있는 모든 잠재적 사용자를 식별하는 AS에서 사용자 프로파일 또는 그룹 프로파일을 작성할 수 있으므로 데이터베이스 관리자는 분산 관계형 데이터베이스 작업을 관리하는 통신 서버시스템에 대해 디폴트 사용자 프로파일을 작성하는 것을 고려할 필요가 없습니다.

반대로 네트워크에 있는 다른 시스템에 대해 AS로서의 활동의 거의 하지 않으며 민감하거나 중요한 자료를 포함하지 않는 iSeries 서버는 분산 관계형 데이터베이스 작업을 관리하는 통신 서버시스템에 대해 디폴트 사용자 프로파일을 사용할 수 있습니다. 이것은 같은 어플리케이션이 이 데이터베이스에서 작업을 처리하는데 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/IP를 사용하는 시스템에 사용될 수 있습니다. 단일 사용자 ID를 설정하고 그 아래에서 서버 작업을 실행할 수 있습니다. 모든 AR에서 ADDSVRAUTE(서버 인증 항목 추가) 명령을 사용하여 연결할 모든 사용자에 대해 사용자 ID가 사용되도록 지정할 수 있습니다. 서버 권한 항목은 AS에서 CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령에 있는 PWDRQD 매개변수의 설정값에 따라 지정된 암호를 갖거나 암호에 *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 그룹 권한 또는 권한 부여 리스트를 사용하여 분산 관계형 데이터베이스 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.

관련 참조

ADDSVRAUTE(서버 인증 항목 추가) 명령

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

어플리케이션 개발

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 프로그램이 여러 다른 서버에 연결할 수 있으므로 프로그래머는 서버 간의 자료 변환에 좀 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 데이터베이스에 대한 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 처리를 위한 SQL 어플리케이션 개발 프로세스와 유사합니다. 차이점은 분산 처리용 어플리케이션은 연결하고 있는 관계형 데이터베이스의 이름을 지정해야 한다는 것입니다. 이름 지정은 프로그램을 사전컴파일할 때나 어플리케이션 내에서 수행될 수 있습니다.

로컬 및 분산 어플리케이션 모두에 같은 SQL 오브젝트를 사용하지만 하나의 오브젝트인 SQL 패키지는 분산 관계형 데이터베이스 지원에만 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. CRTSQLPKG(SQL 패키지 작성) 명령은 기존 분산 SQL 프로그램에 대한 SQL 패키지를 작성합니다.

SQL문이 있는 프로그램을 사전컴파일하려면 DB2 UDB 조회 관리자 및 SQL 개발 킷 라이선스 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 라이선스 프로그램을 사용하면 대화식 SQL을 사용하여 분산 관계형 데이터베이스에 액세스할 수 있습니다. 이 기능은 프로그램을 사전컴파일하거나 컴파일하지 않고 SQL문을 테스트할 수 있기 때문에 프로그램을 디버깅할 때 도움이 됩니다.

관련 정보

SQL 패키지 작성(CRTSQLPKG) 명령

SQL 프로그래밍

분산 관계형 데이터베이스 어플리케이션에 대한 프로그래밍 고려사항

iSeries 서버의 분산 관계형 데이터베이스 어플리케이션에 대한 프로그래밍 고려사항은 두 가지 주요 범주로 나뉘어집니다. 즉, 로컬 서버에서 지원되는 기능에 대한 범주와 다른 서버로 연결하기 위해 갖는 결과에 대한 범주입니다.

관련 참조

20 페이지의 『추가 정보: 분산 관계형 데이터베이스 어플리케이션 설계』

분산 관계형 데이터베이스는 로컬 데이터베이스에서 전용으로 사용하기 위해 개발한 어플리케이션과는 다른 요구사항을 갖고 있습니다.

분산 관계형 데이터베이스 오브젝트 명명

SQL 오브젝트는 iSeries 서버 오브젝트로 작성되고 유지보수됩니다.

iSeries용 DB2 Universal Database 프로그래밍에서 두 가지 명명 규칙인 시스템(*SYS) 및 SQL(*SQL) 중 하나를 사용할 수 있습니다. 사용하는 명명 규칙은 파일 및 표 이름을 규정하는 방법에 영향을 줍니다. 또한 대화식 SQL 표시장치에 사용되는 용어 및 보안에도 영향을 줍니다. 분산 관계형 데이터베이스 어플리케이션은 명명 규칙을 사용하여 다른 iSeries 서버의 오브젝트에 액세스할 수 있습니다. 그러나 프로그램이 iSeries 서버용 비DB2 UDB에 액세스할 경우 SQL 이름만 사용될 수 있습니다. 이는 STRSQL(SQL 시작) 명령의 NAMING 매개변수, CRTSQLxxx 명령 중 하나의 OPTIONS 매개변수 또는 CLI/JDBC에 대한 명명 연결 등록 정보를 사용하여 지정될 수 있습니다.

관련 참조

명명 규칙

시스템(*SYS) 명명 규칙:

시스템 명명 규칙을 사용할 때 파일은 라이브러리/파일 형식의 라이브러리명으로 규정됩니다.

이 명명 규칙으로 작성된 표는 작성되어 있는 라이브러리의 공용 권한을 가정합니다. 표 이름이 명시적으로 규정되지 않고 CRTSQLxxx 또는 CRTSQLPKG 명령의 DFTRDBCOL 매개변수에 디폴트 컬렉션명이 사용된 경우 정적 SQL문에 이 디폴트 컬렉션명을 사용합니다. 파일 이름이 명시적으로 규정되지 않고 디폴트 컬렉션명이 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 컬렉션이나 라이브러리에서 작성됩니다.

SQL(*SQL) 명명 규칙:

SQL 명명 규칙을 사용할 때 표는 *collection.table* 형식의 컬렉션 명으로 규정됩니다.

표 이름이 명시적으로 규정되지 않고 CRTSQLxxx 또는 CRTSQLPKG(SQL 패키지 작성) 명령의 디폴트 관계형 데이터베이스 컬렉션(DFTRDBCOL) 매개변수에 디폴트 컬렉션명이 지정된 경우, 이 디폴트 컬렉션명을 사용합니다. 표 이름이 명시적으로 규정되지 않고 디폴트 컬렉션 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.

관련 정보

SQL 패키지 작성(CRTSQLPKG) 명령

디폴트 컬렉션명:

프로그램을 사전컴파일할 때 CRTSQLxxx 명령의 DFTRDBCOL 매개변수에 이 이름을 제공하여 SQL 프로그램이 사용할 디폴트 컬렉션명을 지정할 수 있습니다.

DFTRDBCOL 매개변수는 *SYS 명명 규칙이 사용된 경우에는 규정되지 않은 파일의 라이브러리, 또는 *SQL 명명 규칙이 사용된 경우에는 규정되지 않은 표의 컬렉션으로 프로그램에 컬렉션 명을 제공합니다. 프로그램을 사전컴파일할 때 디폴트 컬렉션명을 지정하지 않으면 각 명명 규칙에 대해 규정되지 않은 이름의 규칙이 적용됩니다. 디폴트 관계형 데이터베이스 컬렉션 명만 정적 SQL문에 적용됩니다.

CRTSQLPKG(SQL 패키지 작성) 명령에 DFTRDBCOL 매개변수를 사용하여 패키지의 디폴트 컬렉션을 변경할 수도 있습니다. SQL 프로그램이 컴파일된 후 새 SQL 패키지를 작성하여 디폴트 컬렉션을 변경할 수 있습니다.

관련 태스크

111 페이지의 『CRTSQLPKG(SQL 패키지 작성) 명령 사용』

CRTSQLPKG(SQL 패키지 작성) 명령을 사용하여 컴파일된 분산 관계형 데이터베이스 프로그램에서 SQL 패키지를 작성할 수 있습니다. 또한 이 명령을 사용하여 이전에 작성한 SQL 패키지를 대체할 수 있습니다.

관련 정보

SQL 패키지 작성(CRTSQLPKG) 명령

분산 관계형 데이터베이스로 연결

분산 관계형 데이터베이스 애플리케이션이 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.

삽입된 SQL 애플리케이션의 경우, CONNECT의 유형은 CRTSQLxxx 명령의 RDBCNNMTH 매개변수로 판별됩니다. CLI 및 Java 애플리케이션은 해당 언어의 특성 때문에 항상 분산 작업 단위를 사용합니다.

관련 참조

94 페이지의 『분산 관계형 데이터베이스 명령문』

SQL 언어와 함께 들어 있는 명령문은 특히 분산 관계형 데이터베이스를 지원합니다.

리모트 작업 단위:

리모트 작업 단위 기능은 SQL문의 리모트 준비 및 처리를 지원합니다.

컴퓨터 서버 A의 활성 그룹은 컴퓨터 서버 B에 있는 애플리케이션 서버로 연결할 수 있습니다. 그런 다음, 하나 이상의 작업 단위에서 이 활성 그룹은 B에 있는 오브젝트를 참조하는 모든 정적 또는 동적 SQL문을 처리할 수 있습니다. B에서 작업 단위를 종료한 후 활성 그룹은 컴퓨터 서버 C의 애플리케이션 서버로 연결할 수 있습니다.

대부분의 SQL문은 리모트로 준비되고 처리될 수 있으며 다음과 같은 제한사항을 가집니다.

- 단일 SQL문에서 참조한 모든 오브젝트는 같은 애플리케이션 서버에 의해 관리되어야 합니다.
- 작업 단위의 모든 SQL문은 같은 애플리케이션 서버에 의해 실행되어야 합니다.

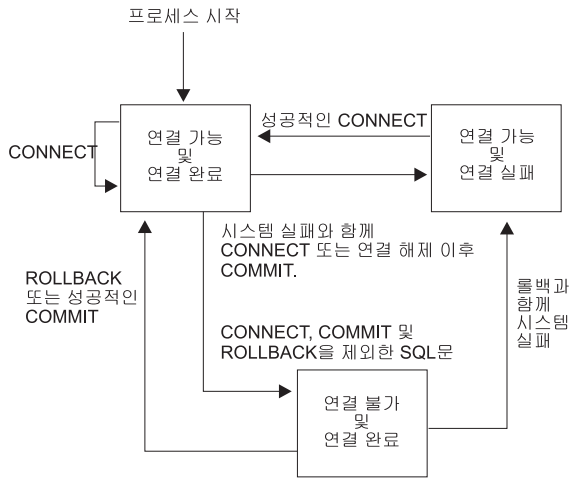
활성 그룹 상태:

활성 그룹의 상태는 항상 다음 세 가지 상태 중 하나입니다.

세 가지 상태는 다음과 같습니다.

- 연결 가능 및 연결된 상태
- 연결 불가능 및 연결된 상태
- 연결 가능 및 연결되지 않은 상태

다음 그림은 상태 전이를 보여줍니다.



RBAL1506-0

그림 9. 리모트 작업 단위 활성화 그룹 연결 상태 전이

활성 그룹의 초기 상태는 연결 가능과 연결된 상태입니다. 활성 그룹이 연결되는 어플리케이션 서버는 CRTSQL_{xxx} 및 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.

CONNECT는 CONNECT, COMMIT, DISCONNECT, SET CONNECTION, RELEASE 또는 ROLLBACK 이외의 다른 SQL문이 선행될 경우 성공적으로 실행될 수 없습니다(COMMIT(*NONE)으로 실행하는 경우는 제외). 오류를 피하려면 CONNECT 명령문을 실행하기 전에 확약 또는 롤백 작업을 실행하십시오.

분산 작업 단위:

어플리케이션 지향의 분산 작업 단위 기능은 또한 SQL문의 리모트 준비 및 실행을 리모트 작업 단위와 같은 방식으로 제공하기도 합니다.

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 단위의는 달리 많은 어플리케이션 서버가 같은 작업 단위에 참여할 수 있습니다. 확약이나 롤백 작업이 작업 단위를 종료합니다.

활성 그룹 상태:

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 그림은 상태 전이를 보여줍니다.

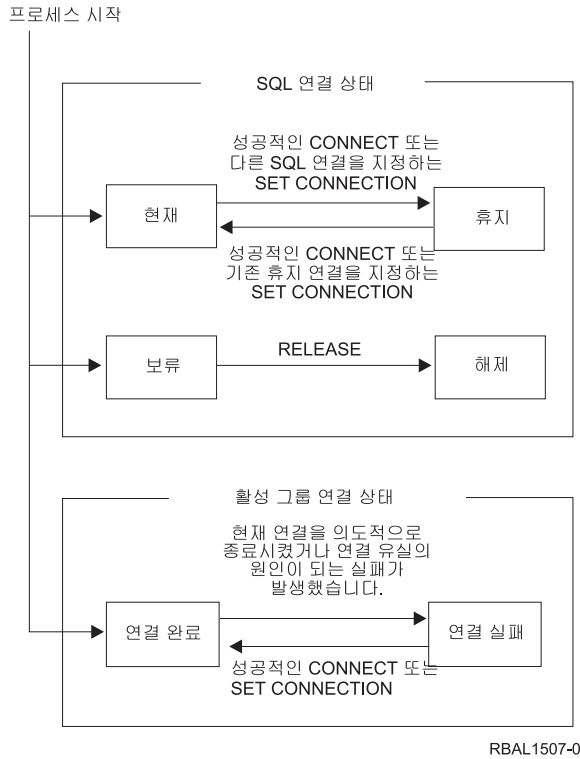


그림 10. 어플리케이션이 지시하는 분산 작업 단위 연결 및 활성 그룹 연결 상태 전이

연결 상태:

이 주제는 서로 다른 연결 상태 및 이들 연결 상태를 변경하는 방법에 대해 설명합니다.

어플리케이션이 CONNECT문을 처리하고 서버명이 어플리케이션 리퀘스터에 알려져 있으며 활성 그룹의 기존 연결 세트에 들어 있지 않은 경우,

- 현재 연결은 정지 상태와 보류 상태가 됩니다.
- 서버 이름이 연결 세트에 추가되고 새 연결이 현재 및 보류 상태가 됩니다.

서버 이름이 이미 활성 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. 활성 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. 활성 그룹이 예약 작업을 실행하면 활성 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.

- 활성 그룹은 같은 어플리케이션 서버에 동시에 하나 이상 연결될 수 없습니다.
- 활성 그룹이 SET CONNECTION 명령문을 실행하면 지정된 위치명은 활성 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 이 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 이 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 예약 작업을 실행하고 연결이 해제 상태에 있거나 어플리케이션 프로세스가 DISCONNECT 명령문을 성공적으로 실행할 때 연결은 의도적으로 종료됩니다.

연결이 종료될 때:

연결이 종료되면 연결을 통해 활성 그룹이 확보한 모든 자원과 연결을 작성하고 유지보수하는 데 사용된 모든 자원이 더 이상 할당되지 않습니다.

예를 들어, 활성 그룹이 RELEASE 명령문을 실행하면 다음 예약 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.

RUW 및 DUW 연결 관리로 실행:

리모트 작업 단위(RUW) 연결 관리로 컴파일된 프로그램을 분산 작업 단위(DUW) 연결 관리로 컴파일된 프로그램으로 호출할 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 활동 중인 연결의 종료를 시도합니다.

이 CONNECT는 활동 연결에 사용된 대화가 보호된 대화를 사용할 경우 실패합니다. 게다가 보호된 대화는 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 RBD 매개변수에 지정된 관계형 데이터베이스입니다. 분산 프로그램이 아닌 경우, 내재적 연결은 로컬 관계형 데이터베이스입니다.

SQL이 활성화되지 않을 때 SQL은 디폴트 활성 그룹의 모든 활동 연결을 종료합니다. 어플리케이션 리퀘스터가 프로세스의 첫 번째 활성 SQL 프로그램이 종료되었으며 다음 조건이 모두 충족되는 것을 감지할 때 SQL은 활성화되지 않습니다.

- 지연 중인 SQL 변경이 없는 경우.
- 보호된 대화를 사용하는 연결이 없는 경우.
- SET TRANSACTION 명령문이 활동 중이 아닌 경우.
- CLOSQLCSR(*ENDJOB)로 사전컴파일된 프로그램이 실행되지 않은 경우.

지연 중인 변경, 보호된 대화 그리고 활동 중인 SET TRANSACTION 명령문이 있으면 SQL은 종료된 상태가 됩니다. CLOSQLCSR(*ENDJOB)로 사전컴파일된 프로그램이 실행된 경우 SQL은 작업이 종료될 때까지 디폴트 활성 그룹에 대해 활동 상태로 남아 있습니다.

- 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 확약 제어가 사용 중이 아니면 활성 그룹 레벨 확약 제어가 사용 중이거나 작업 레벨 확약 정의가 작업 단위 경계에 있습니다.

작업 레벨 확약 제어가 사용 중이고 작업 단위 경계에 있지 않으면 SQL은 종료된 상태가 됩니다.

- SQL이 종료된 상태이고 작업 레벨 확약 정의가 확약되었거나 롤백인 경우.
- 작업의 종료 시.

다음 예제 프로그램은 분배되지 않습니다(연결이 필요하지 않습니다). 지역 수리 정보를 보고서에 수집하는 Spiffy사의 지역 사무소에서 실행되는 프로그램입니다.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```
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 판매 대리점에서 동일한 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)에 대한 내재적 연결. 명령문은 AS에서 실행됩니다.

C 작업 단위의 종료. COMMIT가 성공하면 어플리케이션 리퀘스터(AR)는 연결 가능 및 연결 상태로 됩니다.

D SQL 프로그램 종료 시 내재적 단절.

명시적 연결 관리:

어플리케이션 리퀘스터(AR)를 식별된 어플리케이션 서버(AS)로 명시적으로 연결하려면 CONNECT문을 사용합니다. 이 SQL문은 어플리케이션 프로그램에 삽입하거나 대화식 SQL을 사용하여 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. 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 RESET절을 사용한 CONNECT문이 성공적으로 실행되면 다음과 같은 이벤트가 발생합니다.
 - 어플리케이션 프로세스가 COMMIT HOLD 또는 ROLLBACK HOLD SQL문을 사용하여 연결 가능 상태로 되었거나 COMMIT(*NONE)를 실행 중인 경우, 열려진 모든 커서가 닫히고 준비된 모든 명령문이 삭제되며 보류된 자원이 이전 AS로부터 모두 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 RESET절을 사용한 CONNECT문이 성공적으로 실행되면 다음과 같은 이벤트가 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.

SQLCA에 있는 SQLERRD 필드 및 SQL 진단 영역의 연결 정보에 대한 자세한 정보는 SQL 프로그래밍 주제를 참조하십시오.

- AR이 연결 가능 상태가 아니거나 *server-name*이 로컬 관계형 데이터베이스 디렉토리에 나열되어 있지 않기 때문에 TO 또는 RESET절이 있는 CONNECT문이 성공하지 못할 경우 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 SQL 진단 영역의 해당 정보 및 SQLERRD(4) 및 SQLERRD(5)의 내용이 설명된 이전 단락에서 설명됩니다.

어플리케이션 프로세스가 실행하는 첫 번째 SQL문이 CONNECT 명령문이 되는 것이 좋습니다. 그러나 프로그램에 CONNECT문을 삽입했을 때 프로그램이 둘 이상의 AS에 연결할 경우 AS 이름을 동적으로 변경하고자 할 수 있습니다. 다중 서버에서 어플리케이션을 실행하려는 경우 다음과 같이 호스트 변수가 있는 CONNECT문을 지정하여 프로그램에 관계형 데이터베이스명이 전달되게 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 프로그램의 두 가지 형식의 CONNECT문(1 및 2)을 보여줍니다.

```
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;
```

위의 예(2)는 CONNECT문의 USER/USING 형식을 사용하는 방법을 보여줍니다. CONNECT 명령문의 해당 양식을 프로그램에 삽입할 때 호스트 변수로 사용자 ID와 암호를 지정해야 합니다. TCP/IP를 사용하는 경우, 적절한 매개변수와 함께 ADDSVRAUTE(서버 인증 항목 추가) 명령을 사용하여 사용자 ID 및 암호를 저장했으면 연결 시간에 보안 오브젝트로부터 이 사용자 ID 및 암호를 추출할 수 있습니다.

다음 예는 대화식 SQL에서 두 가지 CONNECT문의 형식을 보여줍니다. 암호는 작은 따옴표 안에 표시되어야 하는 점을 주의하십시오.

```
SQL 명령문을 입력하고 Enter 키를 누르십시오.
현재 연결은 관계형 데이터베이스(RDB) KC105입니다.
CONNECT TO KC000 _____

COMMIT _____
====> CONNECT TO MPLS03 USER JOE USING 'X47K' _____
_____
_____
```

관련 개념

32 페이지의 『관계형 데이터베이스 디렉토리 사용』

i5/OS 라이선스 프로그램은 관계형 데이터베이스 디렉토리를 사용하여 iSeries 서버에서 실행하는 어플리

케이션이 액세스할 수 있는 관계형 데이터베이스명을 정의하고, 연결할 때 SNA 또는 IP를 사용하는지를 지정하며, 이들 관계형 데이터베이스명을 해당 네트워크 매개변수와 연관시킵니다.

관련 참조

124 페이지의 『DDM 대화 제어』

이 주제에서 연결이란 용어는 SQL 연결의 개념을 나타냅니다. SQL 연결은 명시성 또는 내재성의 SQL CONNECT를 행하는 시간부터 논리 SQL 연결이 SQL DISCONNECT와 같은 수단에 의해 종료되거나 RELEASE 다음에 COMMIT이 이어질 때까지 지속됩니다.

분산 관계형 데이터베이스 특정 SQL 및 SQL CALL

분산 iSeries용 DB2 Universal Database 어플리케이션의 사전컴파일 프로세스 중 i5/OS 라이선스 프로그램은 어플리케이션 서버(AS)에서 실행될 SQL 패키지를 빌드할 수 있습니다.

컴파일 이후에 분산 SQL 프로그램과 패키지는 어플리케이션 수신자와 어플리케이션 서버로 사용되는 서버와 호환되어야 합니다. 분산 관계형 데이터베이스 프로그램 준비 주제는 사전컴파일 프로세스에 대한 변경사항과 SQL 패키지의 추가에 대한 자세한 정보를 제공합니다.

이 주제에서는 분산 관계형 데이터베이스 지원에서 사용되는 SQL문의 개요와 다른 서버와의 공존에 대해 고려할 몇 가지 사항을 제공합니다.

관련 개념

103 페이지의 『분산 관계형 데이터베이스 프로그램 준비』

SQL을 사용하여 프로그램을 작성할 때 SQL문을 호스트 프로그램에 삽입할 수 있습니다.

관련 정보

SQL 참조

SQL 프로그래밍

분산 관계형 데이터베이스 명령문:

SQL 언어와 함께 들어 있는 명령문은 특히 분산 관계형 데이터베이스를 지원합니다.

이들 명령문은 다음을 포함합니다.

- 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.

관련 개념

84 페이지의 『분산 관계형 데이터베이스로 연결』

분산 관계형 데이터베이스 어플리케이션이 분산하는 것은 다른 서버의 관계형 데이터베이스에 연결하는 기능입니다.

77 페이지의 『분산 관계형 데이터베이스 오브젝트에 대한 권한』

관계형 데이터베이스 오브젝트에 대한 사용자의 권한을 부여하거나 취소하기 위해 SQL GRANT 및 REVOKE문을 사용하거나 제어 언어(CL)인 GRTOBJAUT(오브젝트 권한 부여) 및 RVKOBJAUT(오브젝트 권한 취소) 명령을 사용할 수 있습니다.

111 페이지의 『SQL 패키지에 대한 작업』

SQL 패키지는 분산 관계형 데이터베이스 어플리케이션이 특정하게 사용하는 SQL 오브젝트입니다. 이 패키지는 어플리케이션 서버(AS)의 자료에 액세스하는 각 SQL문에 대한 제어 구조를 포함합니다.

SQL CALL문(저장 프로시저어):

SQL CALL 명령문이 실제로 분산 관계형 데이터베이스에 특정된 것은 아니지만 기본값이 분산 어플리케이션 논리와 처리에 있기 때문에 이에 대한 설명이 이 곳에 포함되었습니다.

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 프로시저어를 작성하는 데 대한 자세한 정보는 SQL 참조 주제에 있는 SET RESULT SETS 및 CREATE PROCEDURE문의 설명을 참조하십시오.

CALL문은 OSF(Open Software Foundation) 분산 컴퓨팅 환경(DCE)에서 리모트 프로시저어 호출(RPC) 메카니즘이 수행하는 것과 상당히 유사한 기능을 DRDA 환경에 제공합니다. 사실상, 리모트 관계형 데이터베이스에 있는 프로그램에 대한 SQL CALL은 실제 리모트 프로시저어 호출입니다. 이 유형의 RPC에는 인터페이스의 컴파일 필요하지 않고 스태브 프로그램 작성도 필요하지 않다는 장점이 있습니다.

다음과 같은 이유로 사용자는 가끔 저장된 프로시저어라고 하는 SQL CALL을 사용합니다.

- 주어진 기능을 수행하기 위한 어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 간의 메시지 흐름 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 이들 작업은 서버의 확약 제어로 보호되지 않습니다.

- 리모트 서버의 서버 API에 액세스하기 위해.

저장된 프로시저어와 어플리케이션 프로그램은 동일하거나 다른 활성 그룹에서 실행될 수 있습니다. AR의 어플리케이션 프로그램과 AS의 저장된 프로시저어 사이의 일관성을 유지하려면 지정된 ACTGRP(*CALLER)로 저장된 프로시저어를 컴파일할 것을 권장합니다. 저장 프로시저어가 결과 세트를 리턴하도록 설계되어 있다면,

*NEW 활성 그룹에서 실행하도록 작성하면 안됩니다. 그렇게 할 경우, 프로시저가 호출자에게 리턴되고 활성 그룹이 제거될 때 결과 세트와 연관된 커서가 불완전하게 닫혀질 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 ADDRPLYE(응답 리스트 항목 추가) 명령을 사용하고 저장된 프로시저의 CHGJOB(작업 변경) 명령의 INQMSGRPY 매개변수에 *SYSRPLY를 지정하십시오.

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.

관련 참조

108 페이지의 『테스트 및 디버깅』

분산 SQL 프로그램의 테스트와 디버깅은 로컬 SQL 프로그램의 테스트와 디버깅과 유사하지만 프로세스의 특정 측면은 다릅니다.

관련 정보

SQL 참조

ADDRPLYE(응답 리스트 항목 추가) 명령

CHGJOB(작업 변경) 명령

iSeries용 DB2 Universal Database CALL 고려사항:

일부 플랫폼에서 호출되며 C로 작성된 저장된 프로시저는 argc 및 argv를 매개변수로 사용할 수 없습니다. (즉, main() 유형이 될 수 없습니다.) 이는 argc 및 argv를 사용해야 하는 iSeries 저장된 프로시저와는 다릅니다.

DB2 UDB 플랫폼의 저장된 프로시저 예는 #SQLLIB#SAMPLES(또는 /sqllib/samples) 서브디렉토리를 참조하십시오. C 서브디렉토리에서 outsrv.sqc와 outcli.sqc를 검색하십시오.

iSeries 서버가 호출하는 UDB 저장된 프로시저의 경우, 프로시저명이 대문자로 되어 있는지 확인해야 합니다. iSeries 서버는 현재 프로시저명을 대문자로 사용합니다. 이는 UDB 서버에 있는 프로시저가 프로시저명이 같지만 소문자로 되어 있는 경우 이를 찾을 수 없다는 것을 의미합니다. iSeries 서버의 저장된 프로시저의 경우 프로시저명은 대문자입니다.

iSeries 서버의 저장된 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 Universal Database 공존:

SQL 언어를 사용하여 분산 관계형 데이터베이스에 대한 프로그램을 작성 및 유지보수할 때 분산 관계형 데이터베이스 네트워크에 있는 다른 서버를 고려해야 합니다.

작성하거나 유지보수 중인 프로그램은 다음과 호환 가능해야 합니다.

- 기타 iSeries 서버
- 이전 iSeries 서버 릴리스
- iSeries 서버가 아닌 서버

분산 SQL 프로그램의 SQL문은 어플리케이션 서버(AS)에서 실행된다는 점을 기억하십시오. 프로그램이 어플리케이션 리퀘스터(AR)에서 실행되더라도 SQL문은 AS에서 실행될 SQL 패키지에 들어 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. 이는 iSeries용 DB2 Universal Database의 분산 관계형 데이터베이스 지원을 사용하여 프로그램을 다시 컴파일하고 AS에서 필요한 SQL 패키지를 작성합니다.

사용자는 iSeries 서버가 아닌 어플리케이션 서버에서 실행되는 iSeries용 DB2 UDB 프로그램을 작성할 수 있으며 이들 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 시스템이 명명한 활성 그룹에서 실행하는 프로그램이 포함되는 일부의 경우 시스템은 프로그램이 종료될 때 자동 확약 조작을 수행합니다.

이 작동은 i5/OS 오퍼레이팅 시스템에서 COMMIT 및 ROLLBACK이 명령행 또는 CL 프로그램에서 명령으로 사용될 수 있기 때문에 다른 시스템의 경우와 차이가 있습니다. 그러나, 앞의 시나리오는 이 상황을 계획하지 않는 한, 다음 SQL 프로그램이 실행될 때 예상치 못한 결과를 가져올 수 있습니다. 예를 들어, 다음에 대화식 SQL(STSQL 명령)을 실행할 경우 대화식 세션은 미확약 작업이 있는 이전 어플리케이션 서버(AS)로 연결되어 있는 상태로 시작됩니다. 다른 예를 들면, 이전 시나리오를 따를 경우 내재적 연결을 수행하는 두 번째 SQL 프로그램을 시작하고 마지막으로 사용된 AS에서 이에 대한 패키지를 찾아 실행하는 시도가 작성됩니다. 이 AS는 사용자가 의도한 AS가 아닐 수도 있습니다. 이 상황을 피하려면 어플리케이션 프로그램을 종료하기 전에 마지막 작업 단위를 항상 확약하거나 롤백하십시오.

저장 프로시저, 사용자 정의 기능 및 확약 제어

대화식 SQL과 같은 어플리케이션이 DRDA 연결에서 확약 제어가 활성화되지 않고(COMMIT(*NONE)) 실행되는 경우 호출된 저장 프로시저 또는 사용자 정의 기능(UDF)이 iSeries 서버에서 확약 제어를 시작할 수 있습니다.

그러면 클라이언트와 서버간 약속 제어가 일치하지 않아서 어플리케이션이 종료할 때 미약약 갱신이 발생할 수 있습니다.

이 상황은 피해야 합니다. 그러나 구현할 경우 한 가지 해결 방법은 약속 제어에서 실행되는 저장 프로시저어 또는 UDF가 모든 데이터베이스 갱신을 명시적으로 약속하는 것입니다. 그렇지 않으면 서버가 단절 프로세스 중 지연 중인 갱신을 감지하여 지연 중인 작업을 자동으로 약속합니다.

코드화 문자 세트 ID

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 정의하여 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 서로 다른 1차 자국어 버전을 갖는 서로 다른 국가의 시스템 간에 자료를 공유합니다.
- iSeries 서버와 비iSeries 서버가 서로 다른 1차 자국어 버전을 갖는 서로 다른 국가의 시스템 간에 자료를 공유합니다.

태그는 코드화 그래픽 문자에 의미를 할당하는 주요한 수단입니다. 태그는 자료 오브젝트와 연관된 자료 구조에 있거나(명시적 태그 표시) 작업 또는 시스템 자체와 같은 오브젝트에서 계승될 수 있습니다(내재적 태그 표시).

iSeries용 DB2 Universal Database는 CCSID를 사용하여 문자 열에 태그를 붙입니다. CCSID는 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. 어플리케이션을 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

00001 - 28671

IBM이 등록한 CCSID

28672 - 65533

예약

65534 낮은 단계의 CCSID 참조

65535 변환이 실행되지 않음

CDRA CCSID의 전체 리스트는 문자 자료 표시 구조 - 레벨 1 레지스트리를 참조하십시오.

다음 일러스트레이션은 일부 CCSID입니다.

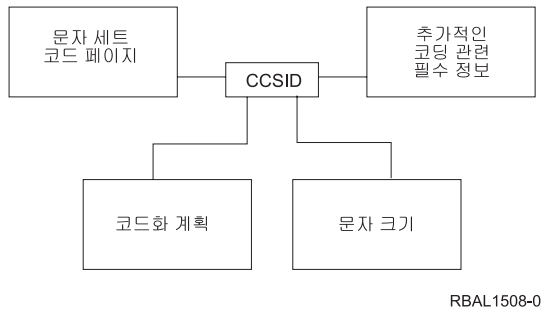


그림 11. 코드화 문자 세트 ID(CCSID)

관련 개념

SQL 프로그래밍

9 페이지의 『DRDA 및 CDRA 지원』

분산 관계형 데이터베이스의 흥미로운 가능성 중 한 가지는 데이터베이스가 여러 유형의 컴퓨터에 확산될 수 있을 뿐 아니라 이들 컴퓨터 역시 여러 국가 또는 영역에 걸쳐 있을 수 있다는 점입니다. iSeries 서버와 같은 동일 서버는 서버에서 사용되는 언어에 따라 자료를 다르게 코드화할 수 있습니다.

관련 정보

i5/OS 국제화

SQL 참조

iSeries 서버 지원:

iSeries 서버의 작업에 대한 디폴트 CCSID는 CHGJOB(작업 변경) 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. 서버가 상이한 시스템과 함께 분산 관계형 데이터베이스에 있는 경우 CCSID 65535를 사용하지 못할 수 있습니다.

어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 간에 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.

작업이 시작된 후 CHGJOB(작업 변경) 명령을 사용하여 작업 CCSID를 변경할 수 있습니다. 이렇게 하려면

1. 작업에 대한 작업 화면을 표시하려면 WRKJOB(작업에 대한 작업) 명령을 입력하십시오.
2. 옵션 2(작업 정의 속성 표시)를 표시하십시오. 현재 CCSID 값을 찾아 작업을 원래의 CCSID 값으로 재설정할 수 있습니다.

3. 새 CCSID 값을 사용하여 CHGJOB(작업 변경) 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 파일에 대한 CRTSRCPF(소스 실제 파일 작성) 또는 CRTPF(실제 파일 작성) 명령에서 CCSID가 명시적으로 지정되지 않은 경우 소스 파일은 작업 CCSID로 태그 표시됩니다. 외부적으로 설명된 데이터베이스 파일과 표는 CCSID가 자료 서술 스펙(DDS), 대화식 자료 정의 유틸리티(IDDU) 또는 CREATE TABLE SQL 문에 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 가정합니다. 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. 태그가 표시된 표의 CCSID를 변경하려면 CHGPF(실제 파일 변경) 명령을 사용하십시오. 열 레벨로 태그가 표시된 표를 변경하려면, 표를 다시 작성한 후 CPYF(파일 복사) 명령에 FMT(*MAP)를 사용하여 새 표에 자료를 복사해야 합니다. 표에서 하나 이상의 뷰가 정의된 경우 이 표를 변경하려면 다음 단계를 수행해야 합니다.

1. 뷰와 표를 액세스 경로와 함께 저장하십시오.
2. 뷰를 삭제하십시오.
3. 표를 변경하십시오.
4. 작성된 표에서 뷰와 액세스 경로를 복원하십시오.

태그가 표시되지 않았거나 CCSID 65535로 내재적으로 태그 표시된 iSeries용 DB2 Universal Database로 마이그레이트된 소스 파일 및 외부 서술 파일은 설치된 오퍼레이팅 시스템의 언어에 따라 디폴트 CCSID로 태그 표시됩니다. 따라서 iSeries용 DB2 UDB에 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: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```
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 CCSID 37에 설정된 QCCSID 시스템 값을 가집니다. 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 이 자료를 CCSID 37로 변환하여 로컬 사용을 위해 :PARTAVAIL에 넣습니다.

관련 개념

249 페이지의 『사용자 FAQ』

이 주제에서는 iSeries 서버의 네트워크(유사 환경)에서 분산 관계형 데이터베이스에 대한 iSeries 지원을 중점적으로 설명합니다.

관련 참조

CHGJOB(작업 변경) 명령

CHGPF(실제 파일 변경) 명령

CPYF(파일 복사) 명령

CRTPF(실제 파일 작성) 명령

CRTSRCPF(소스 실제 파일 작성) 명령

WRKJOB(작업(Job)에 대한 작업) 명령

기타 DRDA 자료 변환

때때로 리모트 시스템에서 처리를 수행할 때 프로그램은 한 시스템에서 자료를 변환하여 다른 시스템에서 사용할 수 있도록 해야 하는 경우가 있습니다. iSeries 서버의 분산 관계형 데이터베이스 구조(DRDA) 지원은 DRDA 지원을 사용하는 기타 시스템 간에 자료를 자동으로 변환합니다.

iSeries용 DB2 Universal Database 어플리케이션 리퀘스터(AR)가 어플리케이션 서버(AS)로 연결될 때 AR은 해당 유형을 식별하는 정보를 송신합니다. 마찬가지로 AS는 해당 프로세서 유형(예: S/390 호스트 또는 iSeries 서버)을 식별하는 정보를 서버로 다시 송신합니다. 그런 다음, 두 시스템은 연결에 정의된 대로 두 시스템 사이의 자료를 자동으로 변환합니다. 따라서 두 시스템의 구조적인 차이점에 대해 프로그램할 필요가 없습니다.

DRDA 지원을 사용하는 IBM 시스템 간의 자료 변환은 다음과 같은 자료 유형을 포함합니다.

- 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 지원을 사용하는 프로그램에서 일부 분산 관계형 데이터베이스 작업을 수행하는 데 도움이 되는 iSeries 분산 자료 관리(DDM) 지원을 사용할 수 있습니다.

예를 들어 많은 레코드를 가져오려면 SQL FETCH문보다는 DDM 및 CPYF(파일 복사) 명령을 사용하는 것이 더 빠를 수 있습니다. 또한 DDM은 분산 관계형 데이터베이스 애플리케이션에 사용하기 위해 컴파일하는 동안 가져온 리모트 시스템 자료의 외부 파일 설명을 얻는 데 사용할 수 있습니다. 이를 수행하려면 26 페이지의 『초기 설정』에 설명된 대로 DDM을 사용해야 합니다.

다음 예는 애플리케이션 서버(AS)와 애플리케이션 리퀘스터(AR)에서 같은 작업을 사용할 수 있도록 관계형 데이터베이스 디렉토리 항목을 추가하고 DDM 파일을 작성하는 방법을 보여줍니다.

주: 대화를 공유하려면 두 연결을 모두 보호하거나 두 연결 모두 보호하지 말아야 합니다.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

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) RDBCNMTH(*RUW)

          PROC :PARTS1;
          OPEN SPIFFY/UPDATE;
          .
          .
          .
          CLOSE SPIFFY/UPDATE;
```



```

.
.
.
EXEC SQL
    SELECT * INTO :PARTAVAIL
            FROM INVENTORY
            WHERE ITEM = :PARTNO;

EXEC SQL
    COMMIT;

.
.
.
END PARTS1;

```

관련 개념

분산 자료 관리

관련 참조

CPYF(파일 복사) 명령

분산 관계형 데이터베이스 프로그램 준비

SQL을 사용하여 프로그램을 작성할 때 SQL문을 호스트 프로그램에 삽입할 수 있습니다.

호스트 프로그램은 호스트 언어(iSeries PL/I, ILE C/400, COBOL/400, ILE COBOL/400, FORTRAN/400, RPG/400 또는 ILE RPG/400 프로그래밍 언어)로 작성되고 SQL문을 포함하는 프로그램입니다. 호스트 프로그램에서는 호스트 변수라고 하는 변수를 사용합니다. 이 변수는 호스트 프로그램으로 식별되는 SQL문에서 사용되는 변수입니다. RPG에서는 이 변수를 필드명이라고 하고, FORTRAN, PL/I 및 C에서는 변수라고 하며, COBOL에서는 이를 자료 항목이라고 합니다.

분산 iSeries용 DB2 Universal Database 프로그램을 분산되지 않은 iSeries용 DB2 UDB 프로그램을 코딩하는 것과 유사한 방법으로 코딩할 수 있습니다. 호스트 언어를 사용하여 호스트 변수가 있는 SQL문을 삽입합니다. 또한, 분산되지 않은 iSeries용 DB2 UDB 프로그램처럼 특정 프로세스를 사용하여 iSeries용 DB2 UDB 프로그램을 준비합니다.

그러나 분산 iSeries용 DB2 UDB 프로그램은 자료에 액세스하기 위해 SQL 패키지를 어플리케이션 서버(AS)에 작성해야 합니다.

이 주제는 분산 iSeries용 DB2 UDB 프로그램에 대한 차이점을 대략적으로 설명하면서 해당 프로세스의 단계를 설명합니다.

SQL문이 있는 프로그램 컴파일

사용자는 삽입된 SQL문이 들어 있는 어플리케이션 프로그램을 실행하기 전에 사전컴파일하고 컴파일해야 합니다. 프로그램이 SQL 사전컴파일러에 의해 실행될 수 있도록 사전 컴파일.

SQL 사전컴파일러는 어플리케이션 프로그램 소스의 각 명령문을 스캔하고 다음을 수행합니다.

- 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 명령문의 구문이 올바르지 않을 경우 사전컴파일러는 SQL문 및 호스트 변수 선언을 올바로 식별하지 않을 수 있습니다.

SQL 사전컴파일 프로세스는 리스트와 임시 소스 파일 멤버를 생성합니다. 또한 사전컴파일러 명령의 OPTION과 RDB 매개변수에 지정된 내용에 따라 SQL 패키지를 생성할 수 있습니다.

관련 참조

106 페이지의 『어플리케이션 프로그램 컴파일』

iSeries용 DB2 Universal Database 사전컴파일러는 *NOGEN 사전컴파일러 옵션이 지정된 경우를 제외하고는 프로그램을 사전컴파일한 후 자동으로 호스트 언어 컴파일러를 호출합니다.

리스팅:

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 사전컴파일에 사용되거나 OUTPUT(*PRINT) 매개변수가 ILE 사전컴파일로 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 쓰여집니다(CRTSQLRPGI를 사용하여 작성한 프로그램의 경우 QTEMP 라이브러리의 QSQLTEMP1).

사전컴파일러로 변경된 소스 코드에서, SQL문은 SQL 인터페이스 모듈 QSQROUTE, 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.

관련 개념

111 페이지의 『SQL 패키지에 대한 작업』

SQL 패키지는 분산 관계형 데이터베이스 어플리케이션이 특정하게 사용하는 SQL 오브젝트입니다. 이 패키지는 어플리케이션 서버(AS)의 자료에 액세스하는 각 SQL문에 대한 제어 구조를 포함합니다.

관련 참조

106 페이지의 『어플리케이션 프로그램 컴파일』

iSeries용 DB2 Universal Database 사전컴파일러는 *NOGEN 사전컴파일러 옵션이 지정된 경우를 제외하고는 프로그램을 사전컴파일한 후 자동으로 호스트 언어 컴파일러를 호출합니다.

107 페이지의 『어플리케이션 바인드』

어플리케이션 프로그램을 실행하기 전에 프로그램, 참조하는 표 및 뷰 사이의 관계를 설정해야 합니다. 이 프로세스를 바인딩이라고 합니다.

사전컴파일러 명령:

DB2 UDB 조회 관리자 및 SQL 개발 킷 라이선스 프로그램에는 각 호스트 언어마다 하나씩 7개의 사전컴파일러 명령이 들어 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.

CRTSQL_{xxx} 명령의 OBJ 매개변수(ILE 사전컴파일의 경우) 또는 PGM 매개변수(비ILE 사전컴파일의 경우)로 프로그램 소스 명령문을 포함하는 멤버명만 지정하고 프로그램을 사전컴파일할 수 있습니다. SQL은 모든 사전컴파일러 매개변수에 대해 디폴트값을 지정합니다(그러나 사용자가 명시적으로 지정하는 값으로 대체될 수 있습니다).

다음 리스트는 분산 관계형 데이터베이스를 지원하는 데 사용되는 모든 CRTSQL_{xxx} 명령에 공통적으로 사용되는 매개변수를 간략히 설명합니다.

RDB

SQL 패키지 옵션을 작성할 관계형 데이터베이스 이름을 지정합니다. *NONE을 지정하면 프로그램 또는 모듈이 분산 오브젝트가 아니므로 CRTSQLPKG(SQL 패키지 작성) 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)
```

관련 참조

SQL 패키지 작성(CRTSQLPKG) 명령

어플리케이션 프로그램 컴파일

iSeries용 DB2 Universal Database 사전컴파일러는 *NOGEN 사전컴파일러 옵션이 지정된 경우를 제외하고는 프로그램을 사전컴파일한 후 자동으로 호스트 언어 컴파일러를 호출합니다.

컴파일러 명령은 프로그램명, 소스 파일 이름, 소스 멤버 이름이 작성된 사전컴파일러, 텍스트 및 사용자 프로 파일 지정이 실행됩니다. 다른 매개변수는 호스트 언어에 따라 컴파일러로 전달되기도 합니다.

관련 개념

103 페이지의 『SQL문이 있는 프로그램 컴파일』

사용자는 삽입된 SQL문이 들어 있는 어플리케이션 프로그램을 실행하기 전에 사전컴파일하고 컴파일해야 합니다. 프로그램이 SQL 사전컴파일러에 의해 실행될 수 있도록 사전 컴파일.

삽입된 SQL 프로그래밍

관련 참조

105 페이지의 『SQL 패키지 작성』

SQL 패키지를 호출한 오브젝트는 CRTSQLxxx 명령을 컴파일할 때 사전컴파일 프로세스의 부분으로 작성될 수 있습니다.

어플리케이션 바인드

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 작업의 경우, 액세스 계획은 SQL 패키지에 저장되고 서버는 이를 SQL 패키지와 함께 관리합니다.

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 않거나 데이터베이스에 대해 성능을 개선할 수 있는 변경사항(예: 색인 추가)이 발생한 것을 감지하면 새 액세스 계획이 자동으로 작성됩니다. 어플리케이션 서버(AS)가 iSeries 서버가 아니면 CRTSQLPKG(SQL 패키지 작성) 명령을 사용하여 바인드를 다시 수행해야 합니다. 바인딩은 세 가지 사항을 실행합니다.

- 데이터베이스에 있는 설명을 사용하여 SQL문의 유효성을 다시 확인합니다.

바인드가 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. 바인드 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 바인드 프로세스는 프로그램에 액세스 계획을 빌드하고 저장합니다.

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 SQL문을 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.

관련 개념

111 페이지의 『SQL 패키지에 대한 작업』

SQL 패키지는 분산 관계형 데이터베이스 어플리케이션이 특정하게 사용하는 SQL 오브젝트입니다. 이 패키지는 어플리케이션 서버(AS)의 자료에 액세스하는 각 SQL문에 대한 제어 구조를 포함합니다.

관련 참조

105 페이지의 『SQL 패키지 작성』

SQL 패키지를 호출한 오브젝트는 CRTSQLxxx 명령을 컴파일할 때 사전컴파일 프로세스의 부분으로 작성될 수 있습니다.

SQL 패키지 작성(CRTSQLPKG) 명령

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 후 이 프로그램을 분산 관계형 데이터베이스 네트워크의 최종 테스트에 사용할 수 있습니다. 리모트 연결을 통해 어플리케이션을 테스트할 때 이 어플리케이션을 어플리케이션 서버(AS)가 될 서버에서 로컬로 테스트할 것을 고려하여 테스트가 분산 환경으로 이동될 때 해당 프로그램만 이동시키면 되도록 하십시오.

분산 SQL 프로그램의 디버깅은 로컬 SQL 프로그램 디버깅과 같은 기술을 사용합니다. STRDBG(디버그 시작) 명령을 사용하여 디버거를 시작하고 어플리케이션을 디버그 모드에 넣을 수 있습니다. 중단점을 추가하고, 명령문을 추적하거나 변수의 내용을 표시할 수 있습니다.

그러나, 분산 SQL 프로그램을 디버그하려면 UPDPDPROD 매개변수에 대해 *YES의 값을 지정해야 합니다. 이는 i5/OS 분산 관계형 데이터베이스 지원이 QSYS 라이브러리에 있는 파일을 사용하고 QSYS가 기간계 라이브러리기 때문입니다. 이렇게 하면 기간계 라이브러리에 있는 자료를 어플리케이션 리퀘스터(AR)에서 변경할 수 있습니다. AR에서 STRDBG(디버그 시작) 명령을 발행하면 단지 AR 작업이 디버그 모드로 되므로 AS에서의 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 리스트 및 분산 관계형 데이터베이스의 오류 메시지 리스트가 172 페이지의 『문제 해결』에 제공됩니다.

또한 디버그 모드에 있는 결과로서 서버가 SQL문의 처리 효율성을 최대화시키는 방법에 대한 정보용 메시지가 발행됩니다. 최대화는 AS에서 발생하므로 AR 작업 기록부에는 이러한 유형의 메시지가 나타나지 않습니다. 이 정보를 얻으려면 AS 작업을 디버그 모드로 해야 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 대한 세부사항은 QRWOPTIONS 자료 영역 사용법을 참조하십시오. 자료 영역은 디버그 시작 뿐 아니라 작업 추적을 시작하고, 작업 기록부를 요구하며 작업 출력 표시 및 기타 다른 작업을 수행하는 데도 사용할 수 있습니다. iSeries AR에서 QRWOPTIONS 설정을 수행할 수도 있으며 옵션을 iSeries 서버 서버에 새도우되게 할 수도 있습니다.

AR 및 AS 모두 iSeries 서버이며 AAPC로 연결되어 있으면 SBMRMTCMD(리모트 명령 제출) 명령을 사용하여 AS 작업에서 디버그 모드를 시작할 수 있습니다. DDM 파일 설정 주제에 설명된 대로 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 이미 존재하지 않을 경우 AS 작업을 시작한 후 이 작업에서 디버그 모드를 시작합니다. 관계형 데이터베이스 활동을 모니터링하기 위한 방법 중 하나를 사용하여 AS 작업 기록부를 조사하고 해당 작업을 찾으십시오.

AS 작업을 디버그 모드로 설정하는 다음 방법은 AR과 iSeries용 DB2 Universal Database AS로 작동되지만 약간의 제한이 있습니다. 이는 어플리케이션이 설정을 수행하기 위한 연결을 한 후 멈출 수 있는지에 따라 결정됩니다. 이는 또한 사용자가 추적 또는 디버그하려는 것이 연결이 성립된 후에 발생한다고 가정합니다.

- AS로 사인 온하고 AS 작업을 찾으십시오.
- 다음과 같은 대화식 작업(AS 작업을 찾기 위해 사용 중인 작업)에서 STRSRVJOB(서비스 작업 시작) 명령을 발행하십시오.

```
STRSRVJOB (job-number/user-ID/job-name)
```

STRSRVJOB 명령에 대한 작업 이름은 AS 작업의 이름입니다. 이 명령을 발행하면 AS 작업에 영향을 주는 대화식 작업에서 특정 명령을 발행할 수 있습니다. 이들 명령 중 하나는 STRDBG(디버그 시작) 명령입니다.

- 대화식 작업에서 UPDPROD 매개변수에 *YES 값을 사용하여 STRDBG 명령을 발행하십시오. AS 작업을 디버그 모드로 설정하여 AS 작업 기록부에 디버그 메시지를 생성합니다.

이 디버그 세션을 종료하려면 대화식 작업을 사인 오프하여 종료하거나 ENDDBG(디버그 종료) 명령 다음에 ENDSRVJOB(서비스 작업 종료) 명령을 사용하십시오.

SQL문을 실행하기 전에 먼저 AS 작업을 디버그 모드로 해야 하므로 AS에서 디버그를 설정할 시간을 가질 수 있도록 어플리케이션을 변경해야 하는 경우도 있습니다. AS 작업은 어플리케이션이 AS에 연결되면 시작됩니다. AS에 연결된 후 AS에서 디버그가 시작될 때까지 대기 상태에 들어가도록 어플리케이션을 코딩할 수 있습니다.

하나의 대기 작업만 있고 다른 클라이언트의 방해가 없는 경우처럼 작업이 발생하기 전에 TCP/IP에 사용될 사전시작 작업을 예상할 수 있으면 지연할 필요가 없습니다.

관련 개념

114 페이지의 『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.

관련 태스크

44 페이지의 『DDM 파일 설정』

iSeries 서버에서 DRDA 지원 구현은 통신을 위해 분산 자료 관리(DDM) 대화를 사용합니다. 따라서 사용자는 분산 관계형 데이터베이스 처리에 관련하여 DDM을 사용할 수 있습니다.

관련 참조

ENDDBG(디버그 종료) 명령

ENDSRVJOB(서비스 작업 종료) 명령

디버그 시작(STRDBG) 명령

서비스 작업 시작(STRSRVJOB) 명령

95 페이지의 『SQL CALL문(저장 프로시저)』

SQL CALL 명령문이 실제로 분산 관계형 데이터베이스에 특정된 것은 아니지만 기본적으로 분산 어플리케이션 논리와 처리에 있기 때문에 이에 대한 설명이 이 곳에 포함되었습니다.

SBMRMTCMD(리포트 명령 제출) 명령

프로그램 참조:

프로그램을 작성할 때 i5/OS 라이선스 프로그램은 참조되는 콜렉션, 표, 뷰, SQL 패키지 및 색인에 대한 모든 정보를 SQL 프로그램에 있는 SQL문에 저장합니다.

DSPPGMREF(프로그램 참조 표시) 명령을 사용하여 프로그램에 있는 모든 오브젝트 참조를 표시할 수 있습니다. SQL 명명 규칙을 사용하면 라이브러리명이 다음 방법 중 하나로 저장됩니다.

- SQL 이름이 완전히 규정되면 콜렉션 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 콜렉션 명이 이름 규정자로 저장됩니다.

서버 명명 규칙을 사용하면 라이브러리명이 다음 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 콜렉션 명이 이름 규정자로 저장됩니다.

관련 참조

프로그램 참조 표시(DSPPGMREF) 명령

SQL 패키지에 대한 작업

SQL 패키지는 분산 관계형 데이터베이스 어플리케이션이 특정하게 사용하는 SQL 오브젝트입니다. 이 패키지는 어플리케이션 서버(AS)의 자료에 액세스하는 각 SQL문에 대한 제어 구조를 포함합니다.

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.
- CRTSQLPKG(SQL 패키지 작성) 명령 사용

관련 참조

94 페이지의 『분산 관계형 데이터베이스 명령문』

SQL 언어와 함께 들어 있는 명령문은 특히 분산 관계형 데이터베이스를 지원합니다.

105 페이지의 『SQL 패키지 작성』

SQL 패키지를 호출한 오브젝트는 CRTSQLxxx 명령을 컴파일할 때 사전컴파일 프로세스의 부분으로 작성될 수 있습니다.

107 페이지의 『어플리케이션 바인드』

어플리케이션 프로그램을 실행하기 전에 프로그램, 참조하는 포 및 뷰 사이의 관계를 설정해야 합니다. 이 프로세스를 바인딩이라고 합니다.

CRTSQLPKG(SQL 패키지 작성) 명령 사용

CRTSQLPKG(SQL 패키지 작성) 명령을 사용하여 컴파일된 분산 관계형 데이터베이스 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.

어플리케이션 서버(AS)에서 SQL 패키지를 작성하는 데는 DB2 UDB 조회 관리자 및 SQL 개발 킷 라이선스 프로그램이 필요하지 않습니다.

관련 참조

83 페이지의 『디폴트 콜렉션명』

프로그램을 사전컴파일할 때 CRTSQLxxx 명령의 DFTRDBCOL 매개변수에 이 이름을 제공하여 SQL 프로그램이 사용할 디폴트 콜렉션명을 지정할 수 있습니다.

SQL 패키지 작성(CRTSQLPKG) 명령

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. DLTSQLPKG(SQL 패키지 삭제) 명령 또는 DROP PACKAGE SQL문을 입력하여 패키지를 삭제할 수도 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) 명령을 사용하여 SQL 패키지 삭제:

DLTSQLPKG(SQL 패키지 삭제) 명령을 사용하여 하나 이상의 SQL 패키지를 삭제할 수 있습니다. 삭제 중인 SQL 패키지가 있는 iSeries 서버에서 DLTSQLPKG 명령을 입력해야 합니다.

사용자는 SQL 패키지에 대한 *OBJEXIST 권한과 패키지가 있는 콜렉션에 대해 적어도 *EXECUTE 권한을 가져야 합니다.

패키지를 제거하는 여러 가지 SQL 메소드가 있습니다.

- DB2 UDB 조회 관리자 및 SQL 개발 킷 라이선스 프로그램이 설치된 경우, 대화식 SQL을 사용하여 어플리케이션 서버(AS)에 연결한 후 SQL DROP PACKAGE문을 사용하여 패키지를 드롭(drop)하십시오.
- 연결하는 SQL 프로그램을 실행한 다음 패키지를 제거하십시오.
- 조회 관리를 사용하여 연결하고 패키지를 제거하십시오.

다음 명령은 SPIFFY 콜렉션에 있는 SQL 패키지 PARTS1을 삭제합니다.

```
DLTSQLPKG SQLPKG(SPIFFY/PARTS1)
```

리모트 iSeries 서버에 있는 SQL 패키지를 삭제하려면 SBMRMTCMD(리모트 명령 제출) 명령을 사용하여 리모트 서버에서 DLTSQLPKG(SQL 패키지 삭제) 명령을 실행하십시오. 또한 리모트 서버를 사인 온하는 데 표시장치 pass-through를 사용하여 SQL 패키지를 삭제할 수 있습니다. 리모트 서버가 iSeries 서버가 아닐 경우 리모트 워크스테이션 프로그램을 사용하여 이 서버로 pass-through한 후 이 서버로 SQL 패키지 삭제 명령을 로컬로 제출하십시오.

관련 참조

DLTSQLPKG(SQL 패키지 삭제) 명령

SBMRMTCMD(리포트 명령 제출) 명령

패키지 권한 수정:

i5/OS에 작성된 프로그램에 대해서는 해당 패키지를 사용할 수 있는 권한이 있는 사용자를 변경할 수 있습니다.

이는 SQL의 GRANT 및 REVOKE문을 사용하여 수행될 수 있습니다.

- GRANT ALL PRIVILEGES ON TABLE table-name TO user(user에 PUBLIC 사용 가능)
- GRANT EXECUTE ON PACKAGE package-name(일반적으로 i5/OS 프로그램명) TO user(user에 PUBLIC 사용 가능)

명령행에서 GRTOBJAUT 및 RVKOBJAUT 명령을 입력하여 수행될 수도 있습니다.

SQL DROP PACKAGE문 사용:

DROP PACKAGE문은 분산 관계형 데이터베이스에 대한 PACKAGE 매개변수를 포함합니다. DROP PACKAGE문을 프로그램에 삽입하거나 대화식 SQL을 사용하여 DROP PACKAGE문을 발행할 수 있습니다.

DROP PACKAGE문을 발행하면 해당 SQL 패키지 및 설명이 어플리케이션 서버(AS)에서 삭제됩니다. 이는 로컬 서버에 입력된 DLTSQPKG(SQL 패키지 삭제) 명령과 동일한 결과를 가집니다. SQL 패키지에 종속적인 다른 오브젝트는 이 명령문 실행 결과 삭제되지 않습니다.

패키지를 삭제하려면 사용자는 SQL 패키지에 대해 다음 권한을 가져야 합니다.

- 참조된 컬렉션에 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.

관련 참조

SQL DROP문

관리

사용자는 분산 관계형 데이터베이스의 관리자로서, 여러 서버에서 실행하는 작업에 대해 책임이 있습니다.

로컬 시스템에서 어플리케이션 리퀘스터(AR)로서 시작되는 작업은 iSeries 서버에서 다른 작업을 모니터링하는 것과 같은 방법으로 모니터링할 수 있습니다.

로컬 시스템에서 어플리케이션 리퀘스터(AS)로서 수행 중인 작업 단위를 추적할 때 같은 도구를 사용하지만 다른 종류의 정보를 찾습니다.

이 주제에서는 네트워크에서 수행되고 있는 분산 관계형 데이터베이스 작업을 관리할 수 있는 방법에 대해 설명합니다. 여기에서 논의되는 대부분의 명령과 프로세스 및 기타 자원은 분산 관계형 데이터베이스 사용을 위한 것만이 아니며 모두 iSeries 서버의 조작을 위해 제공된 툴입니다. 여기에서 논의되는 모든 관리 명령, 프로세스 및 자원은 모든 iSeries용 DB2 Universal Database 기능과 함께 i5/OS 라이선스 프로그램에 포함됩니다. iSeries 서버의 작업 관리 기능은 여러 서버의 작업을 추적할 수 있는 효과적인 방법을 제공합니다.

관련 태스크

44 페이지의 『DDM 파일 설정』

iSeries 서버에서 DRDA 지원 구현은 통신을 위해 분산 자료 관리(DDM) 대화를 사용합니다. 따라서 사용자는 분산 관계형 데이터베이스 처리에 관련하여 DDM을 사용할 수 있습니다.

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.

분산 관계형 데이터베이스의 작업에 대한 작업

WRKJOB(작업에 대한 작업) 명령은 작업에 대한 작업 메뉴를 표시합니다. 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
- 요약 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(스플 파일에 대한 작업)를 사용하여 같은 정보를 찾을 수 있습니다. 이는 스플 파일에 대한 작업 화면을 표시하며 여기서 옵션 5를 사용하여 QPJOBLOG라는 파일을 표시할 수 있습니다(리스트에 있는 경우). WRKJOB(작업에 대한 작업) 명령은 작업에 대한 작업 메뉴를 표시합니다.

관련 참조

WRKJOB(작업(Job)에 대한 작업) 명령

175 페이지의 『어플리케이션이 예상 시간 내에 완료되지 않음』

요구를 완료하는 데 예상된 시간보다 오래 걸리는 경우 먼저 어플리케이션 리퀘스터(AR)를 검사합니다.

분산 관계형 데이터베이스의 사용자 작업에 대한 작업

작업에서 사용되고 있는 사용자 프로파일(사용자명)을 알고 있는 경우 WRKUSRJOB(사용자 작업에 대한 작업) 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 Enter 키를 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
---  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 설명한 작업에 대한 작업 화면을 제공합니다.

서버가 TCP/IP를 사용하는 경우 DDM TCP/IP 서버 작업의 상태를 보고자 할 때 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 유용하게 사용할 수 있습니다. 다음 명령을 실행하십시오.

```
WRKUSRJOB QUSER *ACTIVE
```

QRWT 문자로 시작되는 작업이 표시될 때까지 PgDn 키를 누르십시오. 서버가 활동 중이면 QRWTLSTN이라는 하나의 작업과 QRWTSRVR이라는 하나 이상의 작업(사전시작 DRDA 작업이 서버에서 실행되는 경우)이 표시되어야 합니다. QRWTSRVR 작업은 사전시작 작업입니다. RWTLSN 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 이전에 QRWTSRVR이 정상적으로 실행되는 서브시스템은 QSYSWRK입니다. V5R1 이후 QRWTSRVR은 QUSRWRK에서 실행됩니다.

관련 참조

WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 명령

175 페이지의 『어플리케이션이 예상 시간 내에 완료되지 않음』

요구를 완료하는 데 예상된 시간보다 오래 걸리는 경우 먼저 어플리케이션 리퀘스터(AR)를 검사합니다.

분산 관계형 데이터베이스의 활동 작업에 대한 작업

여러 사용자에게 대해 실행 중인 작업을 모니터링하려는 경우 또는 작업을 찾는 중이며 작업 이름이나 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 일상적인 활동 작업에 대한 작업 화면을 보여줍니다.

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 Enter 키를 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		DEQW
—	QCMN	QSYS	SBS	.0		DEQW
—	QINTER	QSYS	SBS	.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 Enter 키를 누르십시오.
 2=변경 3=보류 4=종료 5=작업 대상 6=해제 7=메세지 표시
 8=스플 파일에 대한 작업 13=단절 ...

-----경과된-----

옵션	서브시스템/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. 여러 사용자에 대해 실행 중인 작업을 모니터링하려는 경우 또는 작업을 찾는 중이며 작업 이름이나 사용자 ID를 알지 못할 경우 WRKACTJOB(활동 작업에 대한 작업) 명령을 사용하십시오.

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. 마우스 오른쪽 버튼을 클릭하고 등록 정보를 선택하십시오. 해당 정보가 표시된 등록 정보 창이 열립니다.

관련 참조

활동 작업에 대한 작업(WRKACTJOB) 명령

167 페이지의 『서버를 통해 분산 관계형 데이터베이스 성능 향상』

충분한 서버 성능을 발휘하려면 서버 자원 간의 적절한 균형이 필요합니다. 임의의 자원을 남용하는 것은 성능에 좋지 않은 영향을 미칩니다. 이 주제는 서버의 성능을 관찰하는 데 도움을 줄 수 있는 서버 명령에 대해 설명합니다.

175 페이지의 『어플리케이션이 예상 시간 내에 완료되지 않음』

요구를 완료하는 데 예상된 시간보다 오래 걸리는 경우 먼저 어플리케이션 리퀘스터(AR)를 검사합니다.

분산 관계형 데이터베이스의 확약 정의에 대한 작업

서버에서 확약 정의에 대해 작업할 경우 WRKCMDFN(확약 정의에 대한 작업) 명령을 사용하십시오.

확약 정의는 STRCMTCTL(확약 제어 시작) 명령으로 확약 제어가 시작될 때 확약 제어에 대한 정보를 저장하는 데 사용됩니다. 이러한 확약 정의는 활동 작업과 연관되거나 연관되지 않을 수 있습니다. 활동 작업과 연관되지 않은 확약 제어는 종료되었지만 해당 논리적 작업 단위 중 하나 이상이 아직 완료되지 않았습니다.

WRKCMDFN(확약 정의에 대한 작업) 명령을 사용하여 확약 정의의 작업명, 상태 또는 논리적 작업 단위 ID에 기초하여 확약 정의에 대한 작업을 수행할 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. 이러한 확약 정의가 들어 있는 작업은 APPC로 보호되는 대화를 통해 통신합니다. 한 서버에 확약 정의를 표시하여 LUWID를 찾은 다음 이를 WRKCMDFN(확약 정의에 대한 작업) 명령에 대한 입력으로 사용하여 이에 해당하는 확약 정의를 찾을 수 있습니다.

WRKCMDFN(확약 정의에 대한 작업) 명령을 사용하여 결정되지 않은 작업에 있는 로컬 자원을 사용 가능하게 할 수 있지만 확약 정의가 PRP(Prepared) 또는 LAP>Last Agent Pending) 상태인 경우에만 해당됩니다. 확약 정의를 강제로 확약 또는 롤백할 수 있으며 따라서 보류된 자원을 해제할 수 있습니다. 그러나 제어는 개시자가 확약 정의에서 수행된 조치를 인식할 때까지 원래 확약을 발행한 프로그램으로 리턴되지 않습니다.

다른 서버에 대해 재동기화가 완료되지 않을 것으로 판별된 경우 WRKCMDFN(확약 정의에 대한 작업) 명령을 사용하여 동기화를 종료할 수도 있습니다.

관련 개념

트랜잭션 및 약속 제어 문제점 해결

관련 참조

STRCMTCTL(약속 제어 시작) 명령

WRKCMDFN(약속 정의에 대한 작업) 명령

분산 관계형 데이터베이스의 작업 기록부를 사용하여 요구 정보 추적

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 종료되고 작업 기록부가 아직 인쇄되지 않은 경우 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하여 작업을 찾은 후 옵션 8(스푼 파일 표시)을 선택하십시오. QPJOB LOC라는 스푼 파일을 찾고 옵션 5(작업 기록부 표시)를 선택하십시오. WRKJOB(작업에 대한 작업) 명령과 작업 화면에 대한 작업에서 다른 옵션을 사용하여 작업 기록부를 표시할 수도 있습니다.

일괄처리 또는 대화식 작업이 아직 활동 중이거나 작업 대기행렬에 있으며 아직 시작하지 않은 경우 WRKUSRJOB 명령을 사용하여 해당 작업을 찾으십시오. WRKACTJOB(활동 작업에 대한 작업) 명령은 활동 작업의 작업 기록부를 표시하는 데 사용되며 작업 대기행렬에 있는 작업은 표시하지 않습니다. 옵션 5(작업(job)에 대한 작업)를 선택한 다음, 옵션 10(작업 기록부 표시)을 선택하십시오.

사용자 자신의 대화식 작업의 작업 기록부를 표시하려면 다음 중 하나를 수행하십시오.

- DSPJOBLOG(작업 기록부 표시) 명령을 입력하십시오.
- WRKJOB(작업에 대한 작업) 명령을 입력하고 작업에 대한 작업 화면에서 옵션 10(작업 기록부 표시)을 선택하십시오.
- 작업 기록부에 나타나는 메시지를 표시하려면 명령 입력 화면에서 F10 키(상세 메시지 표시)를 누르십시오.

DSPJOBLOG(작업 기록부 표시) 명령을 사용하면 작업 기록부 화면이 표시됩니다. 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.

관련 태스크

198 페이지의 『작업 기록부 인쇄』

iSeries 서버 상의 모든 작업에는 그 작업에 대해 입력된 요구와 관련된 정보를 수록한 작업 기록부가 있습니다. 사용자가 어플리케이션 리퀘스터(AR)에서 문제점이 있는 경우 작업 기록부의 정보가 문제점 진단에 도움이 될 수 있습니다.

관련 참조

DSPJOBLOG(작업 기록부 표시) 명령

활동 작업에 대한 작업(WRKACTJOB) 명령

WRKJOB(작업(Job)에 대한 작업) 명령

WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 명령

182 페이지의 『분산 관계형 데이터베이스 메시지』

Application Server(AS) 또는 어플리케이션 리퀘스터(AS)에 오류 메시지가 발생할 경우 서버 메시지는 작업 기록부에 로그되어 실패의 이유를 표시합니다.

분산 관계형 데이터베이스 작업 찾기

어플리케이션 리퀘스터(AR)에서 분산 관계형 데이터베이스 작업에 대한 정보를 찾을 때 사용되는 사용자 프로파일을 알고 있으면 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하여 해당 작업을 찾을 수 있습니다.

어플리케이션 서버(AS)에서 이 명령을 사용할 수도 있지만 AS에서의 사용자 프로파일은 AR에서 사용되는 사용자 프로파일과 다를 수 있다는 점을 주지해야 합니다. TCP/IP 서버에 대하여 작업명을 규정하는 사용자 프로파일은 항상 QUSER이 되며 이것의 작업명은 항상 ORWTSRVR이 됩니다. DSPJOB(기록부 표시) 명령을 사용하면 전체 서버 작업명을 찾는 데 도움이 될 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 작업이 분산 관계형 데이터베이스 작업인지를 보여주고, 그렇다면 어떤 리모트 서버로 작업이 연결되는지 보여줍니다. 작업 기록부를 살펴보면 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

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

AS에 있으면서 작업명을 알지 못하지만 사용자명을 알고 있는 경우, WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하십시오. 사용자를 지정하지 않을 경우, 이 명령은 사용 중인 사용자 프로파일에 있는 작업 리스트를 리턴합니다. TCP/IP의 경우 작업명의 사용자 프로파일은 항상 QUSER가 됩니다. 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 기록부에 대해 서버가 설정된 방법에 따라 활동 작업만 나타날 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 Enter 키를 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			1	2	

활동 중인 AS 작업을 찾고 있으며 사용자명을 알지 못하는 경우, WRKACTJOB(활동 작업에 대한 작업) 명령은 서버에서 활동 중인 서브시스템에 대한 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 Enter 키를 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		DEQW	
4	QCMN	QSYS	SBS	.0		WDEQ	
—	KC0001	KCCLERK	EVK	.0 *		EVTW	
5			6				

- 4 AS 작업을 처리하도록 설정된 서브시스템을 탐색하십시오. 이 예에서, AS 작업용 서브시스템은 QCMN입니다. TCP/IP 서버 작업에 대한 서브시스템은 V5R2 이전에는 QSYSWRK이며 V5R1 이후에는 QUSRWRK입니다.
- 5 APPC AS 작업에 대하여, 작업명은 AS 사용을 위해 작성되어 있는 장치의 장치명입니다.

- 6 나열된 작업 유형은 일반적으로 EVK로, 프로그램 시작 요구에 의해 시작됩니다. TCP/IP AS 작업의 경우 작업 유형은 PJ입니다(DRDA 사전시작 작업이 서버에서 활동 중이지 않은 경우는 제외합니다. 활동 중이지 않은 경우 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 리퀘스터에서 AS에 액세스 중인 작업에 대한 분산 데이터베이스 작업 기록부는 맨 위 쪽에 다음과 같은 명령문을 포함합니다.

CPI3E01

(시스템명)에 의해 액세스된 로컬 관계형 데이터베이스.

AS에서 작업 중인 작업을 찾은 후 AR이 iSeries 서버인 경우 이를 다시 AR에 대해 추적할 수도 있습니다. 다음의 메시지들 중 하나가 사용자의 작업 로그에서 나타나고 사용자가 수신한 메시지에 커서가 위치합니다.

CPI9152

어플리케이션 리퀘스터(AR)가 시작한 목표 DDM 작업

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.

관련 태스크

198 페이지의 『작업 기록부 인쇄』

iSeries 서버 상의 모든 작업에는 그 작업에 대해 입력된 요구와 관련된 정보를 수록한 작업 기록부가 있습니다. 사용자가 어플리케이션 리퀘스터(AR)에서 문제점이 있는 경우 작업 기록부의 정보가 문제점 진단에 도움이 될 수 있습니다.

관련 참조

DSPLOG(기록부 표시) 명령

활동 작업에 대한 작업(WRKACTJOB) 명령

WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 명령

175 페이지의 『어플리케이션이 예상 시간 내에 완료되지 않음』

요구를 완료하는 데 예상된 시간보다 오래 걸리는 경우 먼저 어플리케이션 리퀘스터(AR)를 검사합니다.

리모트 iSeries 서버 조작

사용자는 분산 관계형 데이터베이스의 관리자로서 리모트 iSeries 서버를 조작해야 하는 경우가 있습니다.

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 조장 중인지 확인하는 가장 간단한 방법은 리모트 위치가 해당 서버의 전원을 켜서 분산 관계형 데이터베이스 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.

서버는 실제 시간이나 이전에 스케줄된 시간에 이를 실행할 여러 가지의 방법을 제공합니다. 리모트 서버가 조 작 중일 때 특정 작업을 좀더 자주 수행해야 하는 수가 있습니다. 이를 수행할 수 있는 세 가지 1차적인 방법은 표시장치 pass-through, SBMRMTCMD(리모트 명령 제출) 명령 또는 저장 프로시듀어를 사용하는 것입니다.

SBMRMTCMD(리모트 명령 제출) 명령은 분산 자료 관리(DDM) 지원을 사용하여 CL 명령을 제출하여 어플리케이션 서버(AS)에서 실행합니다. 사용자는 가장 먼저 DDM 파일을 작성해야 합니다. DDM 파일의 리모트 위치 정보는 사용되어야 할 통신 회선을 판별하는 데 사용됩니다. 이와 같이 DDM 파일은 제출된 명령을 수신하기 위한 AS를 식별합니다. DDM 파일과 연관된 리모트 파일은 DDM 파일이 AS에서 실행할 명령을 제출하기 위해 사용될 때 관련되지 않습니다.

SBMRMTCMD(리모트 명령 제출) 명령은 일괄처리 환경 및 QCAEXEC 시스템 프로그램 둘 다에서 실행될 수 있는 모든 CL 명령을 제출할 수 있습니다. 즉, 이 명령은 ALLOW 속성에 대해 지정되는 *BPGM 및 *EXEC의 값을 가집니다. DSPCMD(명령 표시) 명령을 사용하여 ALLOW 속성을 표시할 수 있습니다.

SBMRMTCMD(리모트 명령 제출) 명령의 1차 목적은 어플리케이션 리퀘스터(AR) 사용자 또는 프로그램이 AS에 위치한 오브젝트에서 파일 관리 조작 및 파일 권한 활동을 수행할 수 있도록 하는 것입니다. 이 명령의 2차 목적은 사용자가 비파일 조작(예: 메시지 대기행렬 작성)을 수행하거나 사용자가 작성한 명령을 AS에서 실행되도록 제출할 수 있도록 하는 것입니다. CMD 매개변수를 사용하여 AS에서 실행될 명령을 나타내는 최대 2000자의 문자 스트링을 지정할 수 있습니다.

사용자는 제출되고 있는 CL 명령 및 이 명령이 조작될 오브젝트에 대해 AS에서 적절한 권한을 가지고 있어야 합니다. AR 사용자가 이를 수행하는 데 올바른 권한을 가지고 있는 경우(AS 사용자 프로파일에 판별된 대로) SBMRMTCMD(리모트 명령 제출) 명령을 사용하여 리모트 파일에서 수행할 수 있는 조치는 다음과 같습니다.

- 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 작업에 이 명령을 사용하는 것은 iSeries 서버에서 다른 메소드만큼 효율적이지 못합니다. 예를 들어, 이 명령을 사용하여 리모트 파일의 파일 설명 또는 파일 속성을 표시하거나 파일 또는 기타 오브젝트를 덤프할 수 있지만 해당 출력은 AS에 남아 있습니다. AR에서 리모트 파일 설명 및 파일 속성을 표시하는 데 더 좋은 방법은 SYSTEM(*RMT)을 지정하여 DSPFD(파일 설명 표시) 및 DSPFFD(파일 필드 설명 표시) 명령을 사용하고 리모트 파일과 연관된 DDM 파일의 이름을 지정하는 것입니다.

제출할 수 있는 CL 명령 리스트 및 이들 명령의 사용에 대한 제한사항은 분산 자료 관리 주제를 참조하십시오. 또한 DDM의 대화 공유 방법에 대한 자세한 정보는 DDM 대화 제어를 참조하십시오.

관련 개념

분산 자료 관리

시스템 종료 및 다시 시작 스케줄

관련 태스크

44 페이지의 『DDM 파일 설정』

iSeries 서버에서 DRDA 지원 구현은 통신을 위해 분산 자료 관리(DDM) 대화를 사용합니다. 따라서 사용자는 분산 관계형 데이터베이스 처리에 관련하여 DDM을 사용할 수 있습니다.

관련 참조

『DDM 대화 제어』

이 주제에서 연결이란 용어는 SQL 연결의 개념을 나타냅니다. SQL 연결은 명시성 또는 내재성의 SQL CONNECT를 행하는 시간부터 논리 SQL 연결이 SQL DISCONNECT와 같은 수단에 의해 종료되거나 RELEASE 다음에 COMMIT이 이어질 때까지 지속됩니다.

DSPCMD(명령 표시) 명령

DSPFD(파일 설명 표시) 명령

DSPFFD(파일 필드 설명 표시) 명령

SBMRMTCMD(리모트 명령 제출) 명령

IPL을 제어하는 시스템 값

DDM 대화 제어

이 주제에서 연결이란 용어는 SQL 연결의 개념을 나타냅니다. 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.

주: 대화라는 용어는 SNA APPC 전문용어로 특정한 기술적인 의미를 갖고 있습니다. 이것은 공식적인 의미에서 TCP/IP 전문용어까지 연장되지 않습니다. 그러나 TCP/IP(다른 주제에서는 ‘네트워크 연결’)에는 유사한 개념이 있습니다. 이 책에서 이 단어는 TCP/IP 네트워크 연결에도 적용되는 것으로 사용됩니다. 이 책의 다른 주제에서 이 용어는 특정 APPC의 의미를 갖지만 독자는 해당 문맥에서 그 의미를 깨달을 수 있어야 합니다.

어플리케이션 리퀘스터(AR)가 DRDA를 사용하여 어플리케이션 서버(AS)에 연결할 때 AR은 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 데이터베이스가 사용하는 DDM 대화는 AR이 AS에 연결되어 있는 동안 활동합니다.

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 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).
- 보호 설정된 대화의 경우, 확약 또는 롤백은 모든 SQL 프로그램이 종료되고 모든 DDM 관련 기능이 완료된 후 수행됩니다.
- 이제는 AR 작업이 AS에 연결되지 않습니다.

DDMCNV 작업 속성의 값에 관계없이, 대화는 작업 라우팅 단계 종료 시, 작업 종료 시 또는 작업이 RRTJOB(작업 재라우팅) 명령을 시작할 때 제거됩니다. 활동 작업에서 사용되지 않는 대화 역시 RCLDDMCNV(DDM 대화 재생) 또는 RCLRSC(자원 재생) 명령으로 제거될 수 있습니다. 또한, 통신 회선 장애와 같이 오류가 발생하여 대화가 제거될 수도 있습니다.

DDMCNV 매개변수는 CHGJOB(작업 변경) 명령에 의해 변경되고 OPTION(*DFNA)을 사용한 DSPJOB(작업 표시) 명령에 의해 표시됩니다. 또한 RTVJOBA(작업 속성 검색) 명령을 사용하여 이 매개변수의 값을 얻고 이를 CL 프로그램에 사용할 수 있습니다.

관련 참조

CHGJOB(작업 변경) 명령

SPJOB(작업 표시) 명령

91 페이지의 『명시적 연결 관리』

어플리케이션 리퀘스터(AR)를 식별된 어플리케이션 서버(AS)로 명시적으로 연결하려면 CONNECT문을 사용합니다. 이 SQL문은 어플리케이션 프로그램에 삽입하거나 대화식 SQL을 사용하여 발행할 수 있습니다.

『DDM 자원 재생』

RCLDDMCNV(분산 자료 관리 대화 재생) 명령은 작업에 대한 DDMCNV 속성이 *KEEP인 경우에도 소스 작업에서 현재 사용되고 있지 않는 모든 어플리케이션 대화를 재생합니다.

RCLRSC(자원 재생) 명령

RRTJOB(작업 경로 재지정) 명령

RTVJOBA(작업 속성 검색) 명령

DDM 자원 재생

RCLDDMCNV(분산 자료 관리 대화 재생) 명령은 작업에 대한 DDMCNV 속성이 *KEEP인 경우에도 소스 작업에서 현재 사용되고 있지 않는 모든 어플리케이션 대화를 재생합니다.

이 명령을 사용하면 열려진 파일을 모두 닫거나 RCLRSC(자원 재생) 명령으로 수행된 다른 모든 기능을 수행하지 않고도 사용되지 않는 DDM 대화를 재생할 수 있습니다.

RCLDDMCNV(분산 자료 관리 대화 재생) 명령은 이 명령이 입력된 어플리케이션 리퀘스터(AR)의 작업에 대한 DDM 대화에 적용됩니다. AR 작업에 사용되는 DDM 대화에 대해 연관되는 어플리케이션 서버(AS) 작업이 있습니다. 연관된 DDM 대화가 종료되면 AS 작업이 자동으로 종료됩니다. 종료되는 TCP/IP 대화의 경우, AS 작업은 일반적으로 사전시작 작업이며, 보통 종료되기 보다는 재순환됩니다.

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. 요약 제어를 사용하는 경우, DDM 대화를 재생하기 전에 COMMIT 또는 ROLLBACK 조작을 수행해야 하는 경우가 있습니다.

관련 참조

124 페이지의 『DDM 대화 제어』

이 주제에서 연결이란 용어는 SQL 연결의 개념을 나타냅니다. SQL 연결은 명시성 또는 내재성의 SQL CONNECT를 행하는 시간부터 논리 SQL 연결이 SQL DISCONNECT와 같은 수단에 의해 종료되거나 RELEASE 다음에 COMMIT이 이어질 때까지 지속됩니다.

RCLRSC(자원 재생) 명령

RCLDDMCNV(분산 자료 관리 대화 재생) 명령

프로그램이 사용하는 오브젝트 표시

DSPPGMREF(프로그램 참조 표시) 명령을 사용하여 프로그램 또는 SQL 패키지가 사용하는 표, 자료 영역 및 기타 프로그램을 판별할 수 있습니다. 이 정보는 SQL 패키지 및 컴파일된 컴파일된 프로그램에 대해서만 사용 가능하며 데이터베이스 출력 파일로 쓰거나 표시 및 인쇄할 수 있습니다.

프로그램이나 패키지가 작성되어 있을 때, 이 프로그램이나 패키지 안에 사용되는 특정 오브젝트에 대한 정보가 저장됩니다. 그런 다음, DSPPGMREF(프로그램 참조 표시) 명령에 이 정보를 사용할 수 있습니다. 검색되는 정보는 다음을 포함할 수 있습니다.

- 프로그램이나 패키지의 이름과 텍스트 설명
- 프로그램이나 패키지를 포함하는 라이브러리나 콜렉션의 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.

다음 표는 고급 언어 및 유틸리티가 정보를 저장하는 해당 오브젝트를 보여줍니다.

표 5. 고급 언어가 오브젝트 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

표 5. 고급 언어가 오브젝트 정보를 저장하는 방법 (계속)

언어	파일	프로그램	자료 영역	주 참조
----	----	------	-------	------

Notes:

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. 사용자 정의 명령은 명령에 지정된 파일, 프로그램 또는 자료 영역에 대한 정보도 저장할 수 있습니다. PARM 또는 ELEM 명령문의 FILE, PGM 및 DTAARA 매개변수의 설명을 참조하십시오.
2. 프로그램명은 프로그램명에 리터럴을 사용한 경우(이는 정적 호출로 예를 들어 CALL 'PGM1')에만 저장되고 프로그램명에 COBOL/400 ID를 사용한 경우(동적 호출로 예를 들어 CALL PGM1)에는 저장하지 않습니다.
3. 로컬 자료 영역(LDA)의 사용은 저장되지 않습니다.
4. SQL 패키지에 대한 정보

저장된 파일 정보는 사용의 유형에 대한 항목(수)을 포함합니다. DSPPGMREF(프로그램 참조 표시) 명령의 데이터베이스 파일 출력(OUTFILE 매개변수를 사용할 때 빌드됨)에서 이 항목은 여기에 나열된 코드 중 한 가지 표현입니다. 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

관련 참조

프로그램 참조 표시(DSPPGMREF) 명령
ELEM(요소 정의)

예: 프로그램 참조 표시

어플리케이션 리퀘스터(AR) 프로그램에서 사용되는 오브젝트를 보려면 다음 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. 이들은 여러 개의 관계형 데이터베이스에 위치할 수 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 . . . . . : QSQRROUTE
Library . . . . . : *LIBL
Object type . . . . . : *PGM
Object . . . . . : INVENT
Library . . . . . : SPIFFY
Object type . . . . . : *FILE
File name in program . . . . . :
File usage . . . . . : Input

```

어플리케이션 서버(AS) SQL 패키지에서 사용되는 오브젝트를 보려면 다음 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 삭제

저널 리시버가 들어 있는 콜렉션을 삭제하려고 시도하면 어플리케이션 서버(AS) 작업에 대한 QSYSOPR 메세지 대기행렬로 조회 메세지가 송신될 수 있습니다. AS 및 어플리케이션 리퀘스터(AR) 작업은 이 조회가 응답될 때까지 대기합니다.

메세지 대기행렬에 나타나는 메세지는 다음과 같습니다.

CPA7025

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

AR 작업이 대기 중일 때 이는 마치 어플리케이션이 정지된 것처럼 나타날 수 있습니다. AR 작업이 예상 시간보다 오래 대기 중일 때 다음을 살펴보십시오.

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

주: 어플리케이션이 이와 같이 명백한 '정지' 상태가 되면 작업을 요구하는 어플리케이션은 서버의 조회 메시지가 응답될 때까지 대기됩니다. 이는 MOV OBJ(오브젝트 이동) 명령으로 저널 리시버를 다른 라이브러리로 이동시킬 수 없기 때문입니다. 또 다른 라이브러리에 저장되거나 복원될 수 없습니다. 사용자가 할 수 있는 모든 일은 CRTJRNRCV(저널 리시버 작성) 명령을 사용하여 다른 라이브러리에 새 저널 리시버를 작성하고 CHGJRN(저널 변경) 명령을 사용하여 이를 저널에 접속시키는 것입니다. JRNRCV(*GEN) 매개변수와 CHGJRN(저널 변경) 명령을 사용하여 시스템이 작성하는 모든 새 저널 리시버는 새 라이브러리에 작성됩니다. 저널을 저장할 때 접속된 리시버가 다른 라이브러리에 있을 경우 이 저널의 저장된 버전을 복원할 때 새 저널 리시버 역시 이 다른 라이브러리에 작성됩니다.

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 먼저 ADDRPLYE(응답 리스트 항목 추가) 명령을 사용하여 CPA7025 메시지에 대한 AS 응답 리스트에 항목을 추가해야 합니다.

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

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

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

CHGJOB(작업 설명 변경) 명령을 사용하여 AS에서 이러한 상황이 더 이상 일어나지 않게 함으로써 이 작업 설명을 사용하는 모든 작업이 서버 응답 리스트를 사용하게 할 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 작성하고자 할 수 있습니다.

관련 개념

저널 관리

작업 관리

관련 참조

ADDRPLYE(응답 리스트 항목 추가) 명령

CHGJOB(작업 설명 변경) 명령

- CHGJRN(저널 변경) 명령
- CRTJRNRCV(저널 리시버 작성) 명령
- MOV OBJ(오브젝트 이동) 명령
- SBMRMTCMD(리포트 명령 제출) 명령

분산 관계형 데이터베이스의 작업 사용 통계

서버를 사용하고 있는 사용자와 이 사용자가 사용하고 있는 서버 자원을 판별할 수 있도록 iSeries 서버에서 작업 사용 계정 기능이 자료를 수집합니다. 일반적인 작업 사용 통계는 처리 장치, 프린터, 표시장치 및 데이터 베이스와 통신 기능의 사용과 같이 사용된 서버 및 자원에서 실행되는 작업의 세부사항을 제공합니다.

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

1. CRTJRNRCV(저널 리시버 작성) 명령을 사용하여 저널 리시버를 작성하십시오.
2. CRTJRN(저널 작성) 명령을 사용하여 QSYS/QACGJRN이라는 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. WRKSYSVAL(시스템 값에 대한 작업) 또는 CHGSYSVAL(시스템 값 변경) 명령을 사용하여 계정 레벨 서버 값 QACGLVL을 변경하십시오.

CHGSYSVAL(시스템 값 변경) 명령의 VALUE 매개변수는 작업 계정 저널 항목이 생성되는 때를 결정합니다. *NONE 값은 서버가 작업 사용 계정 저널에서 어떠한 항목도 생산하지 못한다는 뜻입니다. *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) 매개변수의 값을 포함하며, 이 값은 계정 코드 또는 디폴트 값 *USRPRF가 될 수 있습니다. *USRPRF를 지정하면, 작업의 사용자 프로파일에 있는 계정 코드가 사용됩니다.

CRTUSRPRF(사용자 프로파일 작성) 명령 또는 CHGUSRPRF(사용자 프로파일 변경) 명령에 계정 코드 매개변수 ACGCDE를 사용하여 사용자 프로파일에 계정 코드를 추가할 수 있습니다. CRTJOB(작업 설명 작성) 명령 또는 CHGJOB(작업 설명 변경) 명령의 ACGCDE 매개변수에 대해 원하는 계정 코드를 지정하여 특정 작업 설명에 대한 계정 코드를 변경할 수 있습니다.

작업 사용 통계 저널이 설정되면 작업 사용 통계 항목은 CHGSYSVAL(시스템 값 변경) 명령이 적용된 후 서버에 들어가는 다음 작업부터 저널 리시버에 넣어집니다.

DSPJRN(저널 표시) 명령에 OUTFILE 매개변수를 사용하여 처리할 수 있는 데이터베이스 파일로 계정 항목을 쓸 수 있습니다.

- 관련 개념
- 작업 관리

『TCP/IP 서버 관리』

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

관련 참조

25 페이지의 『분산 관계형 데이터베이스에 대한 계정』

분산 자료 사용에 대한 통계와 비용을 계산할 수 있어야 합니다.

CHGJOB(작업 설명 변경) 명령

CHGSYSVAL(작업값 변경) 명령

CHGUSRPRF(사용자 프로파일 변경) 명령

CRTJOB(작업 설명 작성) 명령

CRTJRNRCV(저널 리시버 작성) 명령

CRTJRN(저널 작성) 명령

CRTUSRPRF(사용자 프로파일 작성) 명령

DSPJRN(저널 표시) 명령

WRKSYSVAL(시스템 값에 대한 작업) 명령

TCP/IP 서버 관리

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

i5/OS 라이선스 프로그램과 함께 제공되는 DRDA/DDM TCP/IP 서버를 올바르게 작동하기 위해 기존 시스템 구성을 변경할 필요는 없습니다. 이 서버는 사용자가 i5/OS를 설치할 때 설정되고 구성됩니다. 때로는 사용자의 요구를 더욱 충족시키고 문제점을 해결하며 서버의 성능을 개선하거나 단순히 서버의 작업을 보기 위해 시스템이 서버 작업을 관리하는 방법을 변경하려 할 수도 있습니다. 그와 같이 변경하고 사용자의 처리 요구사항을 만족시키려면, 시스템의 어느 부분이든 영향을 주는 오브젝트와 해당 오브젝트를 변경하는 방법에 대해서 알아야 합니다.

이 주제에서는 상위 레벨에서 서버 작업에 대해 작업하기 위해 이해할 필요가 있는 몇 가지 작업 관리 개념 및 이러한 개념 및 오브젝트가 서버에 관련되는 방법에 대해 설명합니다. iSeries 서버를 관리하는 방법을 완전히 이해하려면 이 주제를 계속하기 전에 작업 관리 주제를 주의깊게 검토할 것을 권장합니다. 그런 다음 이 주제에서는 어떻게 TCP/IP 서버를 관리할 수 있는지 및 시스템의 나머지 부분에 대해 어떻게 서버를 일치시키는지를 보여줍니다.

관련 개념

27 페이지의 『iSeries 서버에서 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.

작업 관리

관련 태스크

43 페이지의 『DRDA에 대한 TCP/IP 서버 설정』

TCP/IP 프로토콜을 사용할 분산 관계형 데이터베이스 구조(DRDA) 어플리케이션 서버(AS)가 있는 경우, DDM TCP/IP 서버를 설정해야 합니다.

관련 참조

131 페이지의 『분산 관계형 데이터베이스의 작업 사용 통계』

서버를 사용하고 있는 사용자와 이 사용자가 사용하고 있는 서버 자원을 판별할 수 있도록 iSeries 서버에서 작업 사용 계정 기능이 자료를 수집합니다. 일반적인 작업 사용 통계는 처리 장치, 프린터, 표시장치 및 데이터베이스와 통신 기능의 사용과 같이 사용된 서버 및 자원에서 실행되는 작업의 세부사항을 제공합니다.

CHGJOB(작업 설명 변경) 명령

CHGSYSVAL(작업값 변경) 명령

CHGUSRPRF(사용자 프로파일 변경) 명령

CRTJOB(작업 설명 작성) 명령

CRTJRNRCV(저널 리시버 작성) 명령

CRTJRN(저널 작성) 명령

CRTUSRPRF(사용자 프로파일 작성) 명령

DSPJRN(저널 표시) 명령

WRKSYSVAL(시스템 값에 대한 작업) 명령

DRDA TCP/IP 서버 전문 용어

iSeries용 DB2 Universal Database에 대한 DDM 및 DRDA TCP/IP 액세스에 대해서는 둘 다 같은 서버 소프트웨어가 사용됩니다.

간결성을 위해 다음 설명에서 *DRDA/DDM* 서버 대신 *DDM* 서버 용어를 사용합니다. 그러나 때때로 문맥에서 규정자를 사용할 필요가 없는 경우 *TCP/IP* 서버, *DRDA* 서버 또는 서버로 언급될 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* 서버라고 언급합니다.

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

- 1 서버라는 용어는 DRDA 어플리케이션 환경에서 DRDA 어플리케이션 서버(또는 AS)와 교환할 수 있게 사용
- 1 됩니다. 서버는 DDM 어플리케이션 환경에서 DDM 목표 시스템과 교환할 수 있게 사용됩니다. (일부 문맥에
- 1 서는 iSeries 시스템(하드웨어)이 서버 또는 iSeries 서버로도 불리는 점을 주의하십시오.)

DDM에 대한 TCP/IP 통신 지원 개념

DRDA와 DDM에 의해 사용되는 TCP/IP 통신 지원에 특수하게 속하는 여러 개념이 있습니다.

TCP/IP를 통한 DRDA 또는 DDM 연결 설정:

TCP/IP 통신 지원을 사용하는 DDM 서버를 개시하기 위해, DRDA 어플리케이션 리퀘스터나 DDM 소스 시스템이 잘 알려진 포트 번호인 446 또는 447에 연결됩니다. DDM 서버는 448 포트에서도 청취하지만 이는 iSeries용 DB2 Universal Database 어플리케이션 리퀘스터나 DDM 클라이언트가 지원하지 않는 연결에 사용하기 위해서만입니다.

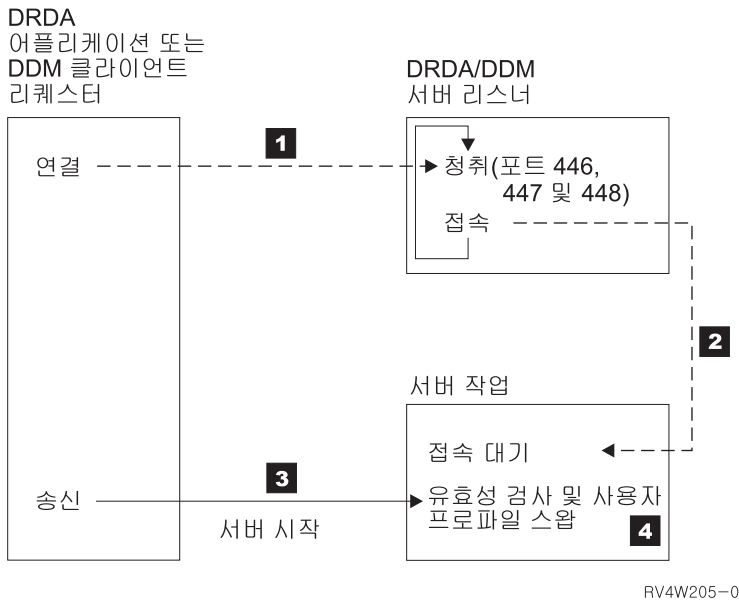


그림 12. DRDA/DDM TCP/IP 서버

1. 클라이언트의 연결 요구를 청취하고 승인하려면 DDM 리스너 프로그램을 STRTCPSVR SERVER(*DDM)(TCP/IP 서버 시작)을 사용하여 시작해야 합니다.

이 연결 요구가 승인되면 DDM 리스너는 클라이언트의 연결을 DDM 서버 작업 2에 접속시키는 내부 요구를 발행합니다. 이 서버 작업은 사전시작 작업이거나, 사용자가 QUSRSYS 또는 사용자 정의 서브시스템에서 QRWTSRVR 사전시작 작업 항목을 제거한 경우(이 경우 사전시작 작업을 사용하지 않음) 클라이언트 연결 요구가 처리될 때 제출되는 일괄처리 작업이 될 수 있습니다. 서버 작업은 클라이언트와 그 이상의 통신을 처리합니다.

초기에 발생하는 자료 교환은 서버 작업이 실행될 사용자 프로파일 3을 식별하는 요구를 포함합니다.

사용자 프로파일 및 암호(사용자 프로파일과 함께 송신된 경우)의 유효성이 검증된 후 서버 작업은 사용자 프로파일 4에 대해 정의된 속성(예: CCSID)을 사용하도록 작업을 변경할 뿐 아니라 이 사용자 프로파일로 교체됩니다.

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.

관련 참조

STRTCPSVR(TCP/IP 서버 시작) 명령

DRDA 및 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. 클라이언트가 서버 작업과 통신을 설정한 후에는 이 통신 기간 동안 클라이언트와 리스너 간에는 더 이상 연관이 없습니다.

DDM 리스너는 DRDA 어플리케이션 리퀘스터와 DDM 소스 시스템이 DDM TCP/IP 서버와 연결을 설치할 수 있도록 활동 중이어야 합니다. CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령을 사용하거나 iSeries Navigator를 통해 DRDA 리스너가 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.

관련 참조

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

STRTCPSVR(TCP/IP 서버 시작) CL 명령:

리스너를 시작하려면 *DDM 또는 *ALL의 SERVER 매개변수 값을 사용하는 STRTCPSVR(TCP/IP 서버 시작) 명령을 사용합니다.

관련 참조

STRTCPSVR(TCP/IP 서버 시작) 명령

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:

이 주제에는 STRTCPSVR(TCP/IP 서버 시작) CL 명령의 몇 가지 예가 나와 있습니다.

예: 모든 TCP/IP 서버 시작

```
STRTCPSVR SERVER(*ALL)
```

이 명령은 DDM 서버를 포함하여 모든 TCP/IP 서버를 시작합니다.

예: DDM TCP/IP 서버만 시작

```
STRTCPSVR *DDM
```

이 명령은 DDM TCP/IP 서버만 시작합니다.

ENDTCPSVR(TCP/IP 서버 종료) CL 명령:

ENDTCPSVR(TCP/IP 서버 종료) 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.

관련 참조

ENDTCPSVR(TCP/IP 서버 종료) 명령

TCP/IP 서버 종료 제한사항:

DDM 리스너가 활동하지 않을 때 이를 종료하기 위해 ENDTCPSVR(TCP/IP 서버 종료) 명령을 사용하면 진단 메시지가 발행됩니다. (ENDTCPSVR) SERVER(*ALL) 명령이 발행될 때 리스너가 활동 중이 아니면 이와 같은 진단 메시지는 송신되지 않습니다.

관련 참조

ENDTCPSVR(TCP/IP 서버 종료) 명령

예: TCP/IP 서버 종료:

이 주제에는 TCP/IP 서버 종료에 관한 몇 가지 예가 나와 있습니다.

예: 모든 TCP/IP 서버 종료

```
ENDTCPSVR *ALL
```

이 명령은 모든 활동 TCP/IP 서버를 종료합니다.

예: 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 네트워크 → 서버 → **TCP/IP** 디렉토리 경로를 따라 수행됩니다.

DRDA/DDM 서버 작업

이 주제는 DRDA/DDM 서버 작업에 대해 설명합니다.

DDM과 서브시스템 설명 및 사전시작 작업 항목:

서브시스템 설명은 서브시스템을 입력하는 작업 방법, 작업 장소, 작업량 및 서브시스템이 해당 작업을 실행하는 데 사용하는 자원에 대해 정의합니다. 다음은 QUSRWRK(또는 OS/400 V5R2 이전에는 QSYSWRK) 서브시스템 설명의 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 시작될 때 활성화되거나 STRPJ(사전시작 작업 시작) 및 ENDPJ(사전시작 작업 종료) 명령으로 제어될 수 있습니다.

관련 참조

ENDPJ(사전시작 작업 종료) 명령

STRPJ(사전시작 작업 시작) 명령

DRDA 및 DDM 사전시작 작업:

사전시작 작업(예: DSPACTPJ(활동 사전시작 작업 표시) 명령)에 관한 서버 정보는 이 정보가 TCP/IP 연결 요구의 결과로서 시작된 사전시작 작업에 해당될 수 있더라도 사전시작 작업을 시작하기 위해 작성된 요구를 나타내기 위해 프로그램 시작 요구라는 용어를 독점적으로 사용합니다.

다음 리스트에는 DDM TCP/IP 서버에 대한 초기 구성 값을 갖는 사전시작 작업 항목 속성이 들어 있습니다. 이들 속성은 CHGPJE(사전시작 작업 항목 변경) 명령으로 변경될 수 있습니다.

- 서브시스템 설명. 사전시작 작업 항목을 포함하는 서브시스템은 V5R2의 QUSRWRK입니다. 이전 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 중이지 않으며 DDM 리스너 작업이 일괄처리 작업인 경우, 즉각적인 DDM 서버 작업은 계속 QSYSWRK 서브시스템에서 실행됩니다.

- **초기 작업 수.** 초기에 구성된 대로 서브시스템이 시작될 때 시작되는 작업 수는 한 개입니다. 이 값은 사용자의 특정 환경 및 요구에 맞게 조정될 수 있습니다.
- **임계값(threshold).** 사전시작 작업 항목에 대해 사용 가능한 최소 사전시작 작업 수는 한 개입니다. 이 임계값에 도달하면 추가 사전시작 작업이 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. V5R2에서는 이러한 가능성을 최소화하기 위해 i5/OS가 변경되었습니다.
- **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 설정되고 나머지 값은 초기 설정값으로 제공된 그대로인 경우 각 사전시작 작업 항목에 대해 다음이 발생합니다.

- 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(예: DSPACTPJ(활동 사전시작 작업 표시) 명령)에 관한 서버 정보는 이 정보가 TCP/IP 연결 요구의 결과로서 시작된 사전시작 작업에 해당될 수 있더라도 사전시작 작업을 시작하기 위해 작성된 요구를 나타내기 위해 프로그램 시작 요구라는 용어를 독점적으로 사용합니다.

관련 태스크

141 페이지의 『DDM 서버 작업 서브시스템 구성』

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.

관련 참조

CHGPJE(사전시작 작업 항목 변경) 명령

DSPACTPJ(활동 사전시작 작업 표시) 명령

모니터 사전시작 작업:

DSPACTPJ(사용 중인 사전시작 작업 표시) 명령을 사용하여 사전시작 작업을 모니터링할 수 있습니다.

DSPACTPJ 명령은 다음 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

관련 참조

DSPACTPJ(활동 사전시작 작업 표시) 명령

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 사용 가능한 사전시작 작업 수를 변경해야 하는지 여부를 나타낼 수 있습니다. 프로그램 시작 요구가 사용 가능한 사전시작 작업을 대기 중임을 나타내는 정보가 표시되면 CHGPJE(사전시작 작업 항목 변경) 명령을 사용하여 사전시작 작업을 변경할 수 있습니다.

프로그램 시작 요구가 충분한 속도로 신속히 작동하고 있지 않으면 다음을 조합하여 수행할 수 있습니다.

- 임계값을 증가시키십시오.
- INLJOBS(초기 작업 수) 매개변수 값을 증가시키십시오.
- ADLJOBS(추가 작업 수) 매개변수 값을 증가시키십시오.

중요한 것은 서버 작업을 시작하도록 송신된 모든 요구에 대해 사용 가능한 사전시작 작업이 있는지 확인하는 것입니다.

관련 참조

CHGPJE(사전시작 작업 항목 변경) 명령

사전시작 작업 항목 제거:

서버가 사전시작 작업 기능을 사용하지 않게 할 경우 다음 지시사항을 따라야 합니다.

1. ENDPJ(사전시작 작업 종료) 명령을 사용하여 사전시작된 작업을 종료하십시오.

ENDPJ 명령으로 종료된 사전시작 작업은 사전시작 작업 항목에 작업 시작 *YES가 지정된 경우 다음 번에 서브시스템이 시작될 때 시작됩니다. 사전시작 작업을 끝내기만 하고 그 다음 단계를 실행하지 않으면, 특정의 서버를 시작하도록 하는 요구는 어느 것이든 장애가 발생합니다.

2. RMVPJE(사전시작 작업 항목 제거) 명령을 사용하여 서브시스템 설명에 있는 사전시작 작업 항목을 제거하십시오.

RMVPJE 명령으로 제거된 사전시작 작업 항목은 서브시스템 설명에서 영구적으로 제거됩니다. 항목이 제거된 다음 서버에 대한 새로운 요구는 성공하지만 작업 개시의 성능 오버헤드가 발생합니다.

관련 참조

ENDPJ(사전시작 작업 종료) 명령

RMVPJE(사전시작 작업 항목 제거) 명령

라우팅 항목:

i5/OS 작업은 서브시스템 설명의 라우팅 항목을 사용하여 서브시스템으로 라우트됩니다. QSYSWRK 서브시스템에서 리스너 작업에 대한 라우팅 입력은 i5/OS이 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 i5/OS을 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.

다음 리스트는 QSYSWRK 서브시스템에서 리스너 작업에 대한 초기 구성을 제공합니다.

서브시스템

QSYSWRK

작업 대기행렬

QSYSNOMAX

사용자

QUSER

라우팅 자료

QRWTLSTN

작업 이름

QRWTLSTN

클래스 QSYSCLS20

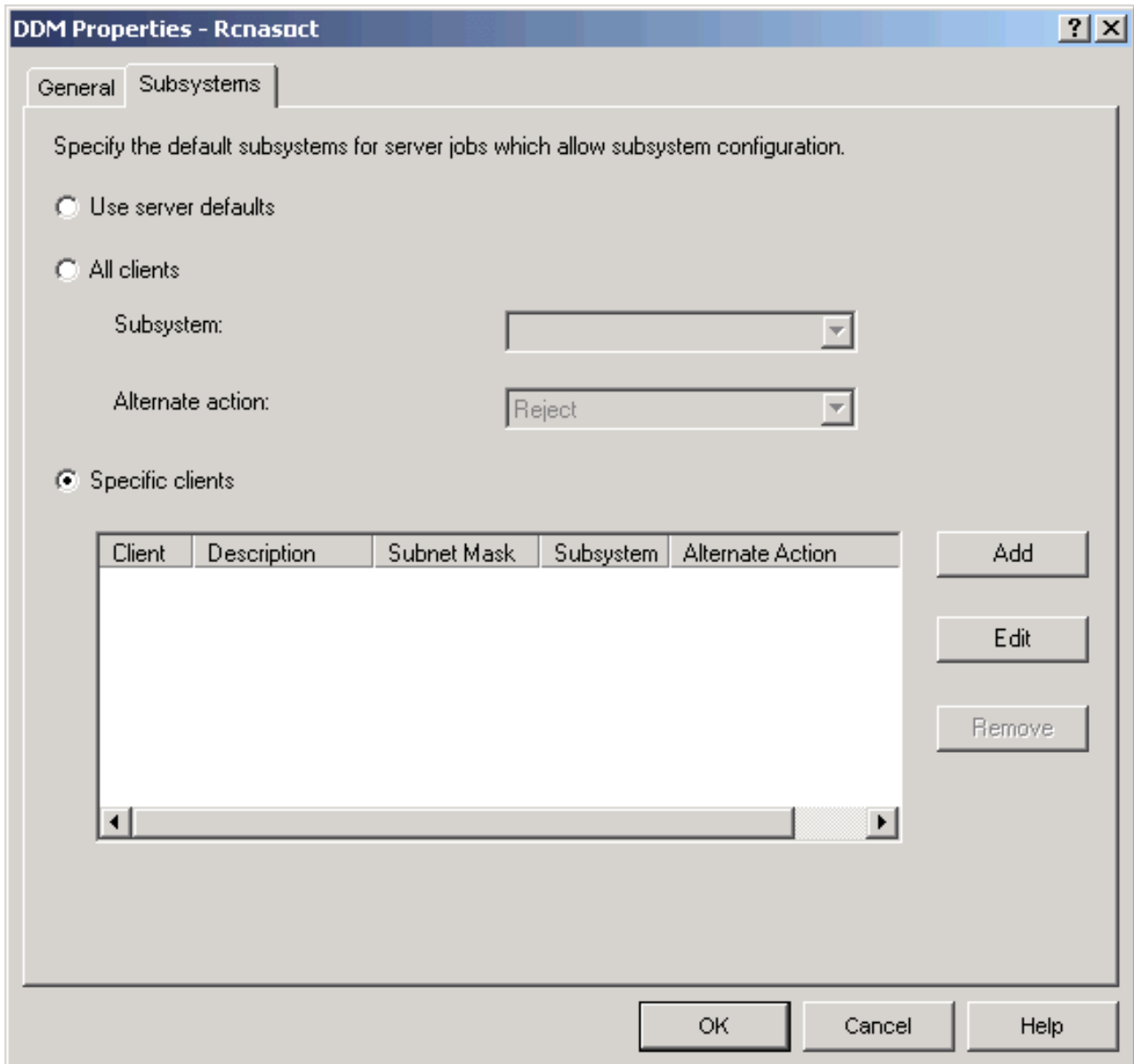
DDM 서버 작업 서브시스템 구성

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. ADDPJE(사전시작 작업 항목 추가) 명령으로 필요한 각 서브시스템에 대한 사전시작 작업 항목을 작성하십시오.
2. STRPJ(사전시작 작업 시작) 명령으로 작성한 사전시작 작업 항목을 시작하십시오.
3. iSeries Navigator에서 **Network**를 펼치십시오.
4. **Servers**를 펼치십시오.
5. **TCP/IP**를 클릭하십시오.
6. 오른쪽 패널에 표시된 서버 리스트에서 **DDM**을 마우스 오른쪽 버튼으로 클릭하고 **Properties**를 선택하십시오.
7. **Subsystems** 탭에서 특정 클라이언트와 서브시스템명을 추가합니다.

다음 예에서 관리자는 QADMIN 서브시스템으로 연결하여 실행될 수 있는 반면 네트워크의 다른 서버는 QUSRWRK로 연결하여 실행될 수 있습니다. 다른 모든 클라이언트는 거부됩니다.



관련 참조

ADDPJE(사전시작 작업 항목 추가) 명령

STRPJ(사전시작 작업 시작) 명령

137 페이지의 『DRDA 및 DDM 사전시작 작업』

사전시작 작업(예: DSPACTPJ(활동 사전시작 작업 표시) 명령)에 관한 서버 정보는 이 정보가 TCP/IP 연결 요구의 결과로서 시작된 사전시작 작업에 해당될 수 있더라도 사전시작 작업을 시작하기 위해 작성된 요구를 나타내기 위해 프로그램 시작 요구라는 용어를 독점적으로 사용합니다.

서버 작업 식별

서버에서 시작된 서버 작업을 살펴볼 때 서버 작업을 특정 어플리케이션 리퀘스터 작업이나 특정 PC 클라이언트로 관련시키는 것이 어렵다는 것을 알게 될 수 있습니다. 문제점을 조사하고 성능 자료를 수집하려면 먼저 특정 작업을 식별할 수 있어야 합니다. iSeries Navigator에서는 해당 작업을 훨씬 쉽게 할 수 있도록 이러한 태스크에 대해 지원합니다.

이 주제에서는 iSeries Navigator를 사용하지 않을 때 디버그 또는 성능 조사를 시작하기 전에 서버 작업을 식별할 수 있는 방법에 대한 정보를 제공합니다.

iSeries 작업명:

iSeries에서 사용되는 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 식별할 수 있도록 하려면 다음 방법을 사용할 수 있습니다.

첫 번째 방법은 WRKACTJOB(활동 작업에 대한 작업) 명령을 사용하는 것입니다. 두 번째 방법은 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하는 것입니다. 세 번째는 어느 클라이언트 사용자에게 어느 작업이 사용되고 있는지를 판별하는 이력 기록부를 표시하는 방법입니다.

관련 참조

활동 작업에 대한 작업(WRKACTJOB) 명령

WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 명령

WRKACTJOB 명령을 사용하여 활동 작업 표시:

WRKACTJOB(활동 작업에 대한 작업) 명령은 모든 활동 작업을 표시합니다. 서버 작업 전체가 표시될 뿐만 아니라 리스너 작업도 표시됩니다.

다음 그림은 WRKACTJOB 명령을 사용하여 샘플 상태를 보여줍니다. 서버에 관련된 작업만 그림에 나타납니다. 사용할 수 있는 사전시작 작업을 참조하려면 F14 키를 누르십시오.

그림에 표시되는 작업 유형은 다음과 같습니다.

- 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 작업: 77

옵션을 입력한 후 Enter 키를 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
—	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

WRKACTJOB(활동 작업에 대한 작업) 명령은 모든 활동 작업을 표시합니다. 서버 작업 전체가 표시될 뿐만 아니라 리스너 작업도 표시됩니다.

관련 참조

활동 작업에 대한 작업(WRKACTJOB) 명령

WRKUSRJOB 명령을 사용하여 활동 사용자 작업 표시:

WRKUSRJOB(사용자 작업에 대한 작업) 명령 USER(QUSER) STATUS(*ACTIVE)는 QUSER 아래에서 실행 중인 모든 활동 서버 작업을 표시합니다. 이는 DDM 리스너와 모든 DDM 서버 작업을 포함합니다. 이 명령은 DDM 관련 작업을 찾기 위해 살펴보아야 하는 작업을 더 적게 나열한다는 점에서 바람직할 수 있습니다.

관련 참조

WRKUSERJOB(사용자 작업(Jobs)에 대한 작업) 명령

이력 기록부 표시:

클라이언트 사용자가 서버 작업과의 연결을 성공적으로 설치할 때마다 해당 작업은 해당 클라이언트 사용자의 프로파일하에 실행하도록 교체됩니다.

특정 클라이언트 사용자와 연관된 작업을 판별하기 위해 DSPLOG(기록부 표시) 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 Enter 키를 누르십시오.

F3=종료 F10=전체 화면 F12=취소
```

다음 예는 MSGID 매개변수가 있는 DSPLOG(기록부 표시) 명령을 사용하여 불필요한 항목을 필터링하는 방법을 보여줍니다.

```
DSPLOG MSGID(CPI3E34)
```

QRWOPTIONS 자료 영역에 적절한 옵션을 설정하여 이들 레코드가 이력 기록부에 쓰여지지 않게 할 수도 있습니다.

관련 개념

213 페이지의 『QRWOPTIONS 자료 영역』

DDM 또는 DRDA TCP/IP 서버 작업이 시작될 때 이들 작업은 사용자가 진단 및 기타 옵션을 지정할 수 있는 자료 영역을 찾습니다. 해당 이름은 QRWOPTIONS이며, 유효하려면 QGPL 라이브러리에 상주해야 합니다. 48자 스트링으로 이루어집니다.

관련 참조

DSPLOG(기록부 표시) 명령

관계형 데이터베이스 디렉토리 감사

이 주제는 관계형 데이터베이스 디렉토리에 대한 프로그램 액세스를 감사하는 방법을 설명합니다.

관계형 데이터베이스 디렉토리에 대한 액세스는 다음에 나열된 조건 중 하나가 충족될 경우 보안 감사 저널에 기록됩니다.

- 시스템 QAUDLVL의 값은 *SYSMGT입니다.
- 사용자 AUDLVL의 값은 *SYSMGT입니다.

*SYSMGT 값을 사용하여 서버는 다음의 명령으로 실행된 액세스 전체에 대해 감사를 실시합니다.

- ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령
- CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 명령
- DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령
- RMVRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 명령
- WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 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 사용하는 RDB 디렉토리에 액세스하는 기존 프로그램은 더 이상 올바르게 실행되지 않을 수 있습니다. *ALLOBJ 특수 권한이 없을 경우 QADBXRDBD에 빌드된 QADBXRMTNM이라는 논리 파일에 액세스해야 합니다. 이 파일에 대한 직접 액세스를 감사하려면 CHGOBJAUD(오브젝트 감사 변경) 명령으로 감사를 설정하십시오.

관련 참조

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령

관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 명령

RMVRDBDIRE(관계형 데이터베이스 디렉토리 항목 제거) 명령

WRKRDBDIRE(관계형 데이터베이스 디렉토리 항목에 대한 작업) 명령

CHGOBJAUD(오브젝트 감사 변경) 명령

자료 가용성 및 보호

분산 관계형 데이터베이스 환경에서 자료 가용성은 네트워크의 개별 서버에 있는 자료를 보호할 뿐 아니라 사용자가 네트워크 전역에 있는 자료에 액세스할 수 있는지 확인하는 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
- 협약 제어
- 보조 기억장치 풀
- 체크섬 보호
- 이중복사 보호 및 무정전 전원 장치

각 서버에 대한 시스템 오퍼레이터는 일반적으로 해당 서버의 자료 백업 및 회복을 담당하고 있는 반면, 사용자는 네트워크 전역에서 자료의 최적 가용성을 위한 전략을 계획할 때 네트워크 중복성 및 자료 중복성의 측면도 고려해야 합니다. 특정 자료가 기업망에 대해 중요할 수록 이 자료에 액세스하기 위한 방법이 더 많이 있어야 합니다.

분산 관계형 데이터베이스에 대한 회복 지원

컴퓨터 서버에서 발생할 수 있는 장애는 서버 장애(전체 서버가 작동하고 있지 않을 때), 화재, 홍수 또는 유사 재해에 의한 사이트 손실 또는 오브젝트의 손상이나 손실 등을 포함합니다. 분산 관계형 데이터베이스의 경우, 네트워크에 있는 한 서버에 장애가 발생하면 네트워크 전역의 사용자가 이 서버에 있는 관계형 데이터베이스에 액세스하지 못하게 됩니다.

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 후 자료 보호 및 회복에 대한 계획은 분산 관계형 데이터베이스에서 특히 중요합니다.

분산 관계형 데이터베이스의 각 서버는 자체 자료를 백업하고 회복할 책임을 가지고 있습니다. 또한 네트워크 안에 각 서버는 이상 서버의 종료 후에 회복 프로시ду어를 처리합니다. 그러나 경험이 없는 조작자나 조작자가 전혀 없는 서버의 경우 백업 및 회복 프로시ду어는 분산 관계형 데이터베이스 관리자가 표시장치 pass-through 를 사용하여 수행될 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.

다음 방법을 사용하여 자료 및 프로그램을 보호할 수 있습니다.

보조 기억장치에 자료 기록

파일 작성 명령의 FRCRATIO(강제-쓰기 비율) 매개변수를 사용하여 보조 기억장치에 자료를 강제로 기록할 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

어플리케이션 서버(AS)가 어플리케이션 리퀘스터(AR)에게 사용 가능한지 확인하는 중요한 부분은 사용자의 시스템이 갑작스런 전원 유실로부터 보호되는지 확인하는 것입니다. 무정전 전원 장치는 별도로 주문 가능하며 전원 장애, 전원 중단 또는 전압 강하에 의해 서버가 유실되지 않도록 보호합니다. 일반적으로 무정전 전원 장치는 모든 워크스테이션에 전원을 공급하지는 않습니다. 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 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.

보조 기억장치 풀(ASP), 체크섬 보호 및 이중복사 보호는 디스크 관련 장애 후 최근에 입력된 자료를 회복하는 방법을 제공하는 i5/OS 디스크 회복 기능입니다. 이들 기능은 추가 서버 자원을 사용하지만 분산 관계형 데이터베이스의 서버에 대해 상위 레벨의 보호 기능을 제공합니다. 일부 서버는 다른 서버보다 어플리케이션 서버로서 더 중요할 수 있으므로 분산 관계형 데이터베이스 관리자는 네트워크에 있는 각 서버에서 이러한 디스크 자료 보호 방법을 가장 잘 사용할 수 있는 방법을 검토해야 합니다.

관련 개념

백업 및 회복

보조 기억장치 풀:

보조 기억장치 풀(ASP)은 동일한 기억장치 영역에 할당된 하나 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 데이터베이스의 체크섬 보호:

체크섬 보호는 보조 기억장치 풀(ASP)에 있는 모든 디스크에서의 자료 유실을 보호합니다.

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 없어지면 서버는 해당 세트에서 남아 있는 기능 단위의 체크섬과 자료를 사용하여 분실된 장치의 내용을 재구성합니다. 이러한 방법으로 장치 중 하나에 장애가 발생할 경우 이 장치의 내용을 회복할 수 있습니다. 재구성된 자료는 장애 발생 시 디스크에 있었던 가장 최근의 정보를 반영합니다. 체크섬 보호는 서버 성능에 크게 영향을 줄 수 있습니다. 분산 관계형 데이터베이스에서 이는 문제가 될 수 있습니다.

분산 관계형 데이터베이스에 대한 이중복사 보호:

이중복사 보호는 디스크 제어기, 디스크 I/O 프로세서 또는 버스와 같은 서로 다른 디스크 관련 하드웨어 구성 요소를 복사함으로써 서버의 가용성을 높여줍니다. 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.

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(해당 저널의 저널 리시버에만 사용됨)에 새 저널 리시버를 작성하여 CHGJRN(저널 변경) 명령으로 이를 접속시키면 다른 사용자 ASP에서 그 다음 서버가 생성한 저널 리시버를 모두 가져오므로 사용자는 성능이 개선된 것을 알 수 있습니다.

표가 작성되었을 때 이것은 컬렉션 안에 작성된 저널 SQL에 자동으로 저널됩니다. 그런 다음 이 저널 기능을 사용하여 저널, 저널 리시버 및 저널에 대한 표 저널링을 관리하는 것은 사용자의 책임입니다. 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 STRJRNPF(실제 파일 저널 시작) 명령에서 IMAGES 매개변수를 지정하여 행 이미지 전후 또는 행 이미지 바로 뒤로 저널할 것인지 여부를 제어합니다.

저널된 전체 데이터베이스 파일은 서버가 시작될 때(IPL 시간) 또는 독립형 ASP의 연결변환 중에 자동으로 해당 저널과 동기화됩니다. 서버가 비정상적으로 종료되거나 독립 ASP가 비정상적으로 단절변환된 경우 일부 데이터베이스 변경은 저널에 포함될 수 있지만 데이터베이스 자체에는 아직 반영되지 않을 수 있습니다. 데이터베이스에도 변경이 생기면 서버는 자동으로 저널에서 데이터베이스를 갱신하고 지금까지의 표를 회생시킵니다.

저널링은 데이터베이스 표가 보다 쉽고 빠르게 저장될 수 있게 합니다. 예를 들어, 전체 표를 매일 저장하는 대신 표의 변경사항을 포함하는 저널 리시버를 저장할 수 있습니다. 전체 표를 계속 정기적으로 저장할 수 있습니다. 이 메소드는 일상의 저장 작업을 실행하는 데 걸리는 시간을 줄일 수 있습니다.

DSPJRN(저널 표시) 명령을 사용하여 저널 리시버 항목을 데이터베이스 파일로 변환할 수 있습니다. 이와 같은 파일은 활동 보고서, 감사 추적, 보안 및 프로그램 디버깅에 사용될 수 있습니다.

관련 개념

저널 관리

관련 정보

CHGJRN(저널 변경) 명령

DSPJRN(저널 표시) 명령

STRJRNPF(저널 실제 파일 시작) 명령

색인 회복:

색인은 표에서 행을 읽는 순서에 대해 설명합니다. 색인이 저널에 기록되어 있으면 서버는 비정상적인 서버 종료 후 IPL을 수행하는 동안 또는 비정상 단절변환 후 독립형 ASP의 연결변환 중 색인을 리빌드하는 데 많은 시간을 소비하지 않고 색인을 회복시킬 수 있습니다.

표를 저널할 때 표 안에 행에 대한 변경사항의 이미지가 저널에 기록됩니다. 이들 행 이미지는 서버 또는 독립형 ASP가 비정상적으로 종료될 경우 표를 회복하는 데 사용됩니다. 그러나 비정상 종료 후 서버는 표에 빌드된 색인이 표에 있는 자료와 동기화되지 않는다는 것을 발견할 수 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 실패 또는 비정상 서서 실패가 발생한 경우, 변경 프로세스 중에 있는 색인은 자료와 일치하는지 확인하기 위해 리빌드될 필요가 있을 수 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 i5/OS 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. 독립 ASP의 연결변환 또는 IPL 중 사용자는 리빌드되어야 하는 색인 및 회복 옵션이 표시된 화면을 볼 수 있으며 이들 회복 옵션을 대체할 수 있습니다.

SQL 색인은 자동으로 저널되지 않습니다. STRJRNAP(저널 액세스 경로 시작) 명령을 사용하여 SQL 조작으로 작성된 모든 색인을 저널링할 수 있습니다. 서버 저장 및 복원 기능을 사용하면 SAVOBJ(오브젝트 저장) 또는 SAVLIB(라이브러리 저장) 명령에 ACCPTH(*YES)를 사용하여 표를 저장할 때 색인을 저장할 수 있습니다. 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.

관련 정보

STRJRNAP(저널 액세스 경로 시작) 명령

SAVOBJ(오브젝트 저장) 명령

라이브러리 저장(SAVLIB) 명령

색인 리빌드 시간을 줄이도록 표 설계:

표 설계는 색인 회복 시간을 줄여 줍니다.

예를 들어, 사용자는 큰 마스터 표를 이력 표와 트랜잭션 표로 나눌 수 있습니다. 그런 다음 트랜잭션 표를 새 자료를 추가하는 데 사용하고 이력 표를 조회용으로만 사용합니다. 매일 트랜잭션 자료를 이력 표에 병합한 후

다음 날의 자료를 위해 트랜잭션 파일을 비울 수 있습니다. 이 설계를 사용하면 서버가 온종일 이상적으로 종료되는 경우에 보다 작은 트랜잭션 표로의 색인을 재설정해야 할 수도 있기 때문에 색인을 재설정하는 시간을 줄일 수 있습니다. 그러나 큰 이력 표에 대한 색인이 그 날 하루 거의 읽기 전용이므로 해당 자료와 동기화될 수 있으며 리빌드할 필요가 없습니다.

색인 재설정 시간을 줄이는 데 표 설계를 사용하는 방법과 액세스 경로 저널링과 같이 서버가 공급하는 기능을 사용하는 방법 사이에 균형을 고려하십시오. 앞에서 설명한 표 설계에는 더 복잡한 어플리케이션 설계가 필요할 수 있습니다. 사용자의 상황을 평가한 후 더 복잡한 어플리케이션을 설계하는 대신 액세스 경로 저널링과 같은 서버가 제공하는 기능을 사용하도록 결정할 수 있습니다.

시스템 관리 액세스 경로 보호:

SMAPP(시스템 관리식 액세스 경로 보호)는 액세스 경로에 대해 자동 보호를 제공합니다.

SMAPP 지원을 사용하면 사용자는 STRJRNAP(저널 액세스 경로 시작) 명령과 같은 저널링 명령을 사용하지 않고도 액세스 경로 저널링의 이점을 얻을 수 있습니다. SMAPP 지원은 IPL 시간 또는 독립 ASP의 연결 변환 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.

관련 정보

STRJRNAP(저널 액세스 경로 시작) 명령

시스템 관리 액세스 경로 보호

확약 제어를 통해 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.

154 페이지의 표 6은 각각의 잠금 레벨 값에 대해 레코드 잠금 지속 시간을 나타내고 있습니다.

프로그램이 사전컴파일되거나 대화식 SQL이 시작될 때 COMMIT(*CHG), COMMIT(*CS) 또는 COMMIT(*ALL)를 요구하면, SQL이 STRCMTCTL(확약 제어 시작) 명령을 내재적으로 호출하여 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 사전컴파일 시간에 저널될 필요는 없으나 어플리케이션을 실행할 때 저널되어야 합니다.

리모트 관계형 데이터베이스가 서버에서 자료에 액세스하고 있고 확약 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.

확약 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. 확약 제어는 확약 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 세션 실행에서 지연 중인 잠겨진 행(레코드)이 있는 경우, 서버 명령 입력 화면에서 COMMIT 또는 ROLLBACK문이 발행될 수 있습니다. 그렇지 않으면, 작업이 끝났을 때 내재성 ROLLBACK 조치가 일어납니다.

WRKCMDFN(확약 정의에 대한 작업) 명령을 사용하여 확약 정의의 상태를 모니터링하고 서버 전체의 확약 제어 활동과 관련된 잠금 및 보류된 자원을 해제합니다.

표 6. 레코드 잠금 기간

SQL 명령문	COMMIT문	레코드 잠금 기간	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
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

표 6. 레코드 잠금 기간 (계속)

SQL 명령문	COMMIT문	레코드 잠금 기간	잠금 유형
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 능한 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 커 서 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 기간 (계속)

SQL 명령문	COMMIT문	레코드 잠금 기간	잠금 유형
Notes:			
<p>1. 결과표가 읽기 전용(DECLARE CURSOR 설명 참조) 및 다음 항목 중 하나에 해당되는 경우 커서는 UPDATE 또는 DELETE 기능으로 열려집니다.</p> <ul style="list-style-type: none"> • 커서가 FOR UPDATE절을 포함하여 정의됩니다. • 커서가 FOR UPDATE, FOR FETCH ONLY 또는 ORDER BY절을 포함하지 않고 정의되며 프로그램이 다음 항목 중 적어도 하나를 포함합니다. <ul style="list-style-type: none"> - 같은 커서명을 참조하는 UPDATE 커서 - 같은 커서명을 참조하는 DELETE 커서 - CRTSQL_{xxx} 명령에 지정된 ALWBLK(*READ) 또는 ALWBLK(*NONE)를 포함하는 EXECUTE 또는 EXECUTE IMMEDIATE 명령문 <p>2. 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.</p> <p>3. 행이 갱신되거나 삭제되지 않을 경우, 잠금은 *READ로 떨어집니다.</p> <p>4. 목표 표의 행에서 UPDATE 잠금과 부속 선택 표의 행에서 READ 잠금.</p> <p>5. 표나 뷰는 COMMIT(*ALL)을 만족시키기 위해서만 오로지 잠길 수 있습니다. 행 잠금은 아직까지 반복 읽기에서 실행됩니다. 사용지정된 잠금과 잠금 지속 시간은 *ALL과 같습니다.</p>			

관련 태스크

118 페이지의 『분산 관계형 데이터베이스의 예약 정의에 대한 작업』

서버에서 예약 정의에 대해 작업할 경우 WRKCMTDFN(예약 정의에 대한 작업) 명령을 사용하십시오.

관련 참조

157 페이지의 『분산 관계형 데이터베이스에 대한 저장 및 복원 처리』

자료 및 프로그램을 저장하고 복원하면 프로그램 또는 서버 장애로부터 회복할 수 있으며 서버 간 정보 교환이나 오브젝트 또는 자료의 오프라인 저장을 할 수 있습니다. 분산 관계형 데이터베이스 네트워크의 각 서버에서 확실한 백업 정책은 문제점이 발생할 때 서버를 신속히 복원하여 네트워크 사용자가 사용할 수 있도록 합니다.

관련 정보

DECLARE CURSOR

STRCMTCTL(예약 제어 시작) 명령

트랜잭션 및 예약 제어 문제점 해결

WRKCMTDFN(예약 정의에 대한 작업) 명령

SAVOBJ(오브젝트 저장) 명령

라이브러리 저장(SAVLIB) 명령

변경된 오브젝트 저장(SAVCHGOBJ) 명령

SAVSAVFDTA(저장 파일 자료 저장) 명령

시스템 저장(SAVSYS) 명령
라이브러리 복원(RSTLIB) 명령
RSTOBJ(오브젝트 복원) 명령
RSTUSRPRF(사용자 프로파일 복원) 명령
RSTAUT(권한 복원) 명령
RSTCFG(구성 복원) 명령

분산 관계형 데이터베이스에 대한 저장 및 복원 처리

자료 및 프로그램을 저장하고 복원하면 프로그램 또는 서버 장애로부터 회복할 수 있으며 서버 간 정보 교환이나 오브젝트 또는 자료의 오프라인 저장을 할 수 있습니다. 분산 관계형 데이터베이스 네트워크의 각 서버에서 확실한 백업 정책은 문제점이 발생할 때 서버를 신속히 복원하여 네트워크 사용자가 사용할 수 있도록 합니다.

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 테이프를 로드할 필요가 없으므로 무인 저장 조작을 수행할 수 있습니다. 분산 관계형 데이터베이스에서 저장 파일은 보호 방법으로서 다른 서버로 송신될 수 있습니다.

정보를 복원하면 정보는 테이프나 저장 파일에서 보조 기억장치로 기록되고 서버 사용자는 이를 액세스할 수 있습니다.

iSeries 서버는 사용자의 데이터베이스 표와 SQL 오브젝트를 저장하고 복원하기 위해 충분한 만큼의 명령 집합을 갖고 있습니다.

- SAVLIB(라이브러리 저장) 명령은 하나 이상의 컬렉션을 저장합니다.
- SAVOBJ(오브젝트 저장) 명령은 SQL 표, 뷰 및 색인과 같은 하나 이상의 오브젝트를 저장합니다.
- SAVCHGOBJ(변경된 오브젝트 저장) 명령은 컬렉션이 마지막 저장된 이후 또는 지정된 날짜로부터 변경된 모든 오브젝트를 저장합니다.
- SAVSAVFDTA(저장 파일 자료 저장) 명령은 저장 파일의 내용을 저장합니다.
- SAVSYS(시스템 저장) 명령은 오퍼레이팅 시스템, 보안 정보, 장치 구성 및 서버 값을 저장합니다.
- RSTLIB(라이브러리 복원) 명령은 컬렉션을 복원합니다.
- RSTOBJ(오브젝트 복원) 명령은 SQL 표, 뷰 및 색인과 같은 하나 이상의 오브젝트를 복원합니다.
- RSTUSRPRF(사용자 프로파일 복원), RSTAUT(권한 복원) 및 RSTCFG(구성 복원) 명령은 SAVSYS(시스템 저장) 명령으로 저장된 사용자 프로파일, 권한 및 구성을 복원합니다.

관련 태스크

118 페이지의 『분산 관계형 데이터베이스의 확약 정의에 대한 작업』

서버에서 확약 정의에 대해 작업할 경우 WRKCMTDFN(확약 정의에 대한 작업) 명령을 사용하십시오.

관련 참조

152 페이지의 『확약 제어를 통해 트랜잭션 회복』

확약 제어는 iSeries 서버에서 저널 관리 기능의 연장입니다. 서버는 단일 작업 단위(트랜잭션)로서 관계형 데이터베이스의 변경 그룹을 식별하고 처리할 수 있습니다.

관련 정보

DECLARE CURSOR

STRCMTCTL(확약 제어 시작) 명령

트랜잭션 및 확약 제어 문제점 해결

WRKCMTDFN(확약 정의에 대한 작업) 명령

SAVOBJ(오브젝트 저장) 명령

라이브러리 저장(SAVLIB) 명령

변경된 오브젝트 저장(SAVCHGOBJ) 명령

SAVSAVFDTA(저장 파일 자료 저장) 명령

시스템 저장(SAVSYS) 명령

라이브러리 복원(RSTLIB) 명령

RSTOBJ(오브젝트 복원) 명령

RSTUSRPRF(사용자 프로파일 복원) 명령

RSTAUT(권한 복원) 명령

RSTCFG(구성 복원) 명령

분산 관계형 데이터베이스 환경에서 색인 저장 및 복원:

SQL 색인을 복원하는 것은 재설정하는 것보다 빠를 수 있습니다. 여러 요소에 따라 시간이 다양하기는 하지만 데이터베이스 색인을 리빌드하는 데 걸리는 시간은 10 000행당 1분 정도입니다.

색인을 복원한 후 최신 저널 변경사항을 적용하여(저널링이 활동 중인지 여부에 따라) 표를 최신 상태로 가져와야 합니다. 이러한 추가 회복 시간을 포함하더라도 색인을 리빌드하는 것보다 색인을 복원하는 것이 더 빠르다는 것을 알게 될 수 있습니다.

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

관련 개념

백업 및 회복

분산 관계형 데이터베이스 환경에 보안 정보 저장 및 복원:

여러 가지 CL 명령을 사용하여 보안 정보를 저장하고 복원할 수 있습니다.

사용자 프로파일을 갱신하고 분산 관계형 데이터베이스 네트워크에서의 사용자 권한을 갱신하여 서버 보안 환경을 자주 변경할 경우, 사용자는 완전한 SAVSYS(시스템 저장) 명령인 전용 서버를 사용하는 오래 실행하는 프로세스를 사용하지 않고 매체 또는 저장 파일에 보안 정보를 저장할 수 있습니다. SAVSECDTA(보안 자료 저장) 명령을 사용하면 전용 서버를 사용하지 않고도 짧은 시간에 보안 자료를 저장할 수 있습니다. SAVSECDTA 명령을 사용하여 저장된 자료는 RSTUSRPRF(사용자 프로파일 복원) 또는 RSTAUT(권한 복원) 명령을 사용하여 복원할 수 있습니다.

SAVSECDTA(보안 자료 저장) 및 RSTUSRPRF(사용자 프로파일 복원) 명령이 저장하고 복원할 수 있는 보안 정보에 포함된 항목은 DRDA TCP/IP 지원이 리모트 서버 사용자 ID 및 암호 정보를 저장하고 복원하는데 사용하는 서버 권한 항목입니다.

관련 정보

시스템 저장(SAVSYS) 명령

SAVSECDTA(보안 자료 저장) 명령

RSTUSRPRF(사용자 프로파일 복원) 명령

RSTAUT(권한 복원) 명령

분산 관계형 데이터베이스 환경에서 SQL 패키지 저장 및 복원:

리모트 서버의 관계형 데이터베이스를 참조하는 어플리케이션 프로그램을 사전컴파일하고 바인드할 때 어플리케이션에 있는 SQL문을 처리하는 데 필요한 제어 구조를 포함시키기 위해 어플리케이션 서버(AS)에 SQL 패키지를 작성합니다.

SQL 패키지는 iSeries 오브젝트이므로 SAVOBJ(오브젝트 저장) 명령을 사용하여 매체 또는 저장 파일에 저장되고 RSTOBJ(오브젝트 복원) 명령을 사용하여 복원될 수 있습니다.

SQL 패키지는 저장된 컬렉션과 같은 이름을 갖는 컬렉션으로 복원되어야 하며 SQL 패키지의 이름을 변경할 수는 없습니다.

관련 정보

RSTOBJ(오브젝트 복원) 명령

SAVOBJ(오브젝트 저장) 명령

관계형 데이터베이스 디렉토리 저장 및 복원:

관계형 데이터베이스 디렉토리는 iSeries 오브젝트가 아닙니다. 대신 IPL 시간에 서버가 여는 파일로 구성됩니다.

따라서 이들 파일을 직접 저장하려면 SAVOBJ(오브젝트 저장) 명령을 사용할 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.

항목을 추가한 후 관계형 데이터베이스 디렉토리를 저장하려면 DSPRBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령에 OUTFILE 매개변수를 지정하여 이 명령의 결과를 출력 파일로 보내십시오. 출력 파일은 테이프 또는 저장 파일에 저장되고 서버로 복원될 수 있습니다. 관계형 데이터베이스 디렉토리가 손상되거나 서버를 회복해야 하는 경우 제어 언어(CL) 프로그램을 사용하여 관계형 데이터베이스 항목 자료가 있는 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 RBDIR이라는 이름의 출력 파일로 송신됩니다.

```
DSPRBDIRE OUTPUT(*OUTFILE) OUTFILE(RBDIR)
```

다음의 샘플 CL 프로그램은 출력 파일 RBDIR의 내용을 읽고 ADDRBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령을 사용하여 관계형 데이터베이스 디렉토리를 재작성합니다. 오래된 디렉토리 항목은 새 항목이 만들어지기 전에 제거된다는 점에 주의하십시오.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```

/*****/
/* - Restore RDB Entries from output file created with:          - */
/* -   DSPRBDIRE OUTPUT(*OUTFILE) OUTFILE(RBDIR)                - */
/* -   FROM A V4R2 OR LATER LEVEL OF OS/400 or i5/OS            - */
/*****/
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 &RWRLOC  *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(&RWRLOC) TEXT(&RWTEXT) )

ELSE IF (&RWTYPE = &SNA) THEN( +
 QSYS/ADDRDBDIRE RDB(&RWRDB) RMTLOCNAME(&RWRLOC) 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 RDBSAV
END

ENDPGM

앞에서 설명한 출력 파일 유형을 사용할 수 없는 경우 다른 방법으로 디렉토리를 복원할 수 있습니다. 이 방법은 저장된 서버에서 오브젝트를 추출하여 이를 다른 라이브러리에 복원한 다음 ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 명령을 사용하여 수동으로 항목을 오브젝트에 입력하는 것입니다.

SAVSYS(시스템 저장) 명령을 실행하면 관계형 데이터베이스 디렉토리를 구성하는 파일이 저장됩니다. 다음 RSTOBJ(오브젝트 복원) 명령을 사용하여 관계형 데이터베이스 디렉토리가 들어 있는 실제 파일을 저장 매체에서 라이브러리로 복원할 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 i5/OS의 제품 코드를 나타냅니다(예를 들어, 버전 5 릴리스 3에 대해 5722SS1). LABEL 매개변수에 있는 *vrmtx*는 i5/OS의 버전, 릴리스 및 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.

관련 참조

39 페이지의 『예: 관계형 데이터베이스 디렉토리 설정』

Spiffy사 네트워크는 관계형 데이터베이스 디렉토리가 분산 관계형 데이터베이스 네트워크의 서버에서 사용되는 방법을 설명하고 각 디렉토리를 설정하는 방법을 보여주는 예를 제공합니다. 이 예는 통신에 APPC(Advanced Program-to-Program Communication)를 사용하는 것으로 가정하며, 이는 TCP/IP에 반대되며 설정하기가 더 간편합니다.

관련 정보

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

DSPRDBDIRE(관계형 데이터베이스 디렉토리 항목 표시) 명령

RSTOBJ(오브젝트 복원) 명령

SAVOBJ(오브젝트 저장) 명령

시스템 저장(SAVSYS) 명령

분산 관계형 데이터베이스에 대한 네트워크 중복 고려사항

네트워크 중복은 분산 관계형 데이터베이스 네트워크의 사용자가 네트워크에 있는 관계형 데이터베이스에 액세스할 수 있는 다양한 방법을 제공합니다.

어플리케이션 리퀘스터(AR)에서 어플리케이션 서버(AS)로 통신 경로가 하나 뿐인 경우 통신 회선이 중단될 때 AR의 사용자는 AS 관계형 데이터베이스에 액세스할 수 없습니다. 이 때문에 네트워크 중복 고려사항은 Spiffy사의 분산 관계형 데이터베이스 관리자에게 중요한 사항이 됩니다. 예를 들어, 서비스 계약이나 고객 부품 구매를 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 이는 판매 대리점이 지역 재고 관리 센터로부터 부품 정보를 요구할 수 없음을 의미합니다. 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.

지역의 판매 대리점에 지역 재고 자료에 대한 액세스 제공은 Spiffy사 분산 관계형 데이터베이스 관리자에게 중요한 일입니다. 네트워크를 통한 자료에 대한 경로를 제공하는 방법은 여러 가지로 설명될 수 있습니다. Spiffy사의 원래의 네트워크 구성은 끝 노드의 판매 대리점을 이것의 해당 네트워크 노드의 지역 센터에 링크하였습니다.

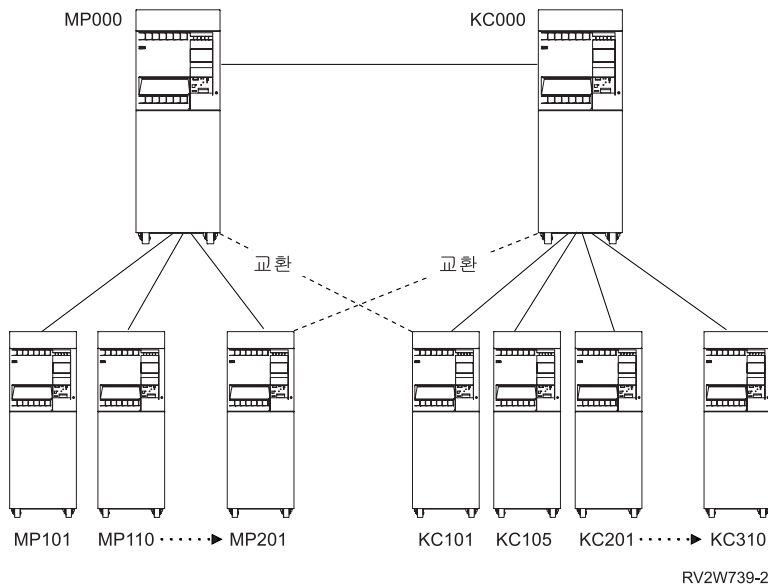


그림 13. 대체 네트워크 경로

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. 첫 번째 그림에서 MP000 서버가 지원하는 일부 AR은 MP000 서버를 사용할 수 없을 때마다 사용할 수 있는 KC000 서버에 대한 링크를 설정합니다. 서버 오퍼레이터나 분산 관계형 데이터베이스 관리자는 VRYCFG(구성 변경) 또는 WRKCFGSTS(구성 상태에 대한 작업) 명령을 사용하여 필요할 때 회선을 켜고 1차 AS가 사용 가능하면 회선을 끌 수 있습니다.

또다른 방법은 더 큰 지역의 판매 대리점 중 하나가 다른 판매 대리점의 AS로도 활동하는 경우가 될 수 있습니다. 두 번째 그림에서 보는 바와 같이 네트워크 노드 전체에서 다른 끝 노드에 대한 끝 노드는 AS 뿐입니다. 첫 번째 그림에서 미니애폴리스로의 링크가 다운될 경우 판매 대리점은 다른 끝 노드에 재고를 조회할 수 없습니다. 위에 설명된 구성은 판매 대리점 중 하나가 APPN 네트워크 노드로 구성되고 다른 지역의 판매 대리점으로부터 이 판매 대리점으로 회선이 설정되도록 변경될 수 있습니다.

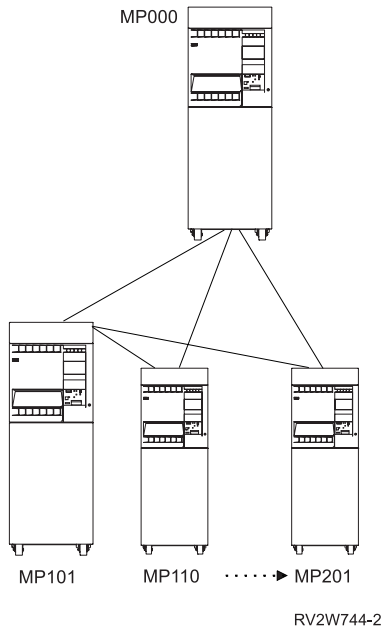


그림 14. 대체 어플리케이션 서버

관련 정보

VRFCFG(구성 변환) 명령

WRKCFGSIS(구성 상태에 대한 작업) 명령

분산 관계형 데이터베이스 네트워크에서 자료 중복

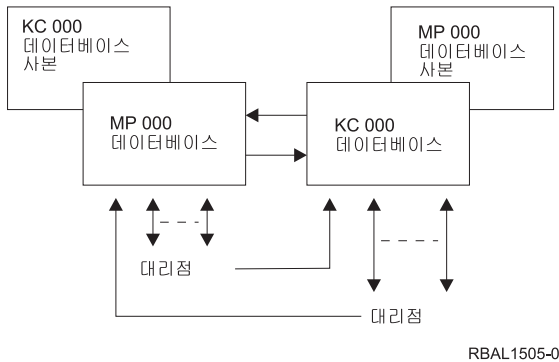
분산 관계형 데이터베이스에서 자료 중복은 또한 분산 관계형 데이터베이스 네트워크의 사용자가 네트워크의 데이터베이스에 액세스할 수 있는 여러 가지 방법을 제공합니다.

분산 관계형 데이터베이스 관리자가 자료 중복 전략을 작성하기 위해 검사하는 고려사항은 해당 자료에 대해 통신 경로를 사용할 수 있는지 확인하는 것 이상으로 복잡합니다.

표는 네트워크에서 서버 전체를 통해 복사될 수 있고 자료의 스냅샷은 자료 가용성을 제공하는 데 사용될 수 있습니다. iSeries용 DataPropagator, V8.1은 이 기능을 제공할 수 있습니다.

다음 그림은 KC000 서버에 MP000 서버 분산 관계형 데이터베이스 사본을 저장할 수 있고 MP000 서버에 KC000 서버 분산 관계형 데이터베이스 사본을 저장할 수 있다는 것을 보여줍니다. 한 지역의 어플리케이션

리퀘스터(AR)는 다른 어플리케이션 서버(AS)에 링크하여 관계형 데이터베이스의 복제된 사본을 조회하거나 갱신할 수 있습니다.



RBAL1505-0

그림 15. 자료 중복 예

관리자는 분산 관계형 데이터베이스 처리를 할 수 있는 가장 효율적이고 효과적인 전략을 결정해야 합니다. 대체 전략이 이 시나리오에 포함될 수도 있습니다.

한 가지 대안은 MP000 서버를 사용할 수 없을 때 해당 AR을 KC000 서버에 연결하여 MP000 분산 관계형 데이터베이스의 읽기 전용 스냅샷을 조회함으로써 서비스 작업을 스케줄할 수 있게 하는 것입니다.

iSeries용 DataPropagator, V8.1은 표의 읽기 전용 사본(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 회선으로 돌아오면 해당 AR은 MP000 분산 관계형 데이터베이스를 조회하여 재고 요구 또는 스냅샷에서 조회된 기타 작업을 완전히 처리합니다.

또다른 방법은 Spiffy사가 판매 대리점 사용자가 지역 AS를 사용할 수 없을 때 다른 AS에서 복제된 표를 갱신할 수 있도록 하는 것일 수 있습니다.

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. 이러한 저널 항목이 원래 MP000 표에 적용될 때 분산 관계형 데이터베이스 사용자는 MP000을 AS로서 다시 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.

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.

관련 개념

172 페이지의 『문제 해결』

분산 관계형 데이터베이스에 액세스하는 중 문제점이 발생할 때 관리자는 문제점의 특성을 판별하고 그 문제점이 어플리케이션에 대한 것인지 로컬 또는 리모트 시스템에 대한 것인지 판별합니다.

네트워크를 통해 분산 관계형 데이터베이스 성능 향상

여러 가지 방법으로 네트워크 성능을 개선할 수 있습니다.

다음과 같은 방법이 있습니다.

- 회선 속도
- 페이싱
- 프레임 크기
- RU 크기
- 연결 유형(비전환 대 전환)

주: 다음 상황 중 하나에서 DRDA 연결에 비보호 대화를 사용합니다.

- 리모트 작업 단위(RUW) 연결 관리를 사용하여 프로그램에서 연결이 수행될 때
- 연결을 작성하는 프로그램이 확약 제어 하에서 실행 중이지 않을 때
- 연결이 작성된 데이터베이스가 사용 중인 프로토콜에 대해 2단계 확약을 지원하지 않을 때

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)로 실행할 때 가능한 비보호 대화로 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.

관련 정보

통신 관리 PDF

APPC, APPN 및 HPR

TCP/IP 설정

서버를 통해 분산 관계형 데이터베이스 성능 향상

충분한 서버 성능을 발휘하려면 서버 자원 간의 적절한 균형이 필요합니다. 임의의 자원을 남용하는 것은 성능에 좋지 않은 영향을 미칩니다. 이 주제는 서버의 성능을 관찰하는 데 도움을 줄 수 있는 서버 명령에 대해 설명합니다.

실행 분석을 돕기 위해 iSeries 성능 분석 툴 라이선스 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
- WRKACTJOB(활동 작업에 대한 작업) 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.

시스템 성능을 관찰할 때 WRKSYSSTS(시스템 상태에 대한 작업) 및 WRKACTJOB(활동 작업에 대한 작업) 명령을 둘 다 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 관찰하십시오.
- 스포 처리량. 활동 출력기에 대한 AuxIO 및 CPU% 값을 관찰하십시오.

변경 조절을 할 때마다 반드시 모든 기본 성능 측정을 측정하고 비교해야 합니다. 한 번에 하나씩 조정하고 평가하십시오.

관련 태스크

116 페이지의 『분산 관계형 데이터베이스의 활동 작업에 대한 작업』

여러 사용자에게 대해 실행 중인 작업을 모니터하려는 경우 또는 작업을 찾는 중이며 작업 이름이나 사용자 ID를 알지 못할 경우 WRKACTJOB(활동 작업에 대한 작업) 명령을 사용하십시오.

관련 정보

시스템 상태에 대한 작업(WRKSYSSTS) 명령

디스크 상태에 대한 작업(WRKDSKSTS) 명령

활동 작업에 대한 작업(WRKACTJOB) 명령

작업 관리

데이터베이스를 통해 분산 관계형 데이터베이스 성능 향상

분산 관계형 데이터베이스 성능은 전체적인 데이터베이스의 설계에 의해 영향을 받습니다. 분산 자료의 위치, 사용하는 제약 제어의 레벨 및 SQL 색인의 설계는 모두 성능에 영향을 줍니다.

관련 개념

17 페이지의 『계획 및 설계』

분산 관계형 데이터베이스의 성공적인 작업을 위한 첫 번째 요구사항은 계획을 통해 이루어집니다. 분산 관계형 데이터베이스를 사용하기 위한 결정을 할 때 기업의 요구와 목표를 고려해야 합니다.

DRDA 자료 위치 결정

어플리케이션과 이 어플리케이션에 필요한 자료 사이에 네트워크가 있으면 성능이 떨어질 수 있으므로 자료를 넣을 위치를 결정할 때는 다음을 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.

DRDA에 대한 블로킹에 영향을 미치는 요소

어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 간 자료 전송 시 블로킹이 발생하는지 여부는 아주 중요한 성능 요소입니다. 자료 블록으로서 전송된 행의 그룹은 한 번에 하나의 행을 송신할 때보다 훨씬 적은 통신 오버헤드를 필요로 합니다.

다른 iSeries 서버로 연결될 때 블로킹을 제어하는 한 가지 방법은 SQL 복수 행 INSERT 및 복수 행 FETCH 문을 사용하는 것입니다. 복수 행 FETCH 명령어는 중대한 오류나 자료 끝이 충돌하지 않는 한, FOR n ROWS 절에 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 각 경우에 대해 나열된 모든 조건을 사용하면 블로킹을 예방할 수 있습니다.

관련 참조

175 페이지의 『어플리케이션이 예상 시간 내에 완료되지 않음』

요구를 완료하는 데 예상된 시간보다 오래 걸리는 경우 먼저 어플리케이션 리퀘스터(AR)를 검사합니다.

iSeries용 DB2 Universal Database 대 iSeries용 DB2 Universal Database 블로킹:

다음의 경우 블로킹이 발생하지 않습니다.

- 커서가 갱신 가능합니다(주 1 참조).
- 커서가 잠재적으로 갱신 가능합니다(주 2 참조).
- ALWBLK(*NONE) 사전컴파일 옵션이 사용되었습니다.
- 확약 제어 레벨이 *CS이고 OS/400의 레벨이 V3R1보다 이전입니다.
- 확약 제어 레벨이 *ALL이고 외부 부속 선택이 다음 항목 중 하나를 포함하지 않습니다.
 - DISTINCT 키워드
 - UNION 오퍼레이터
 - ORDER BY절 및 절 안의 필드 길이 합의 정렬이 필요합니다.
 - 서버 데이터베이스 파일(서버 데이터베이스 파일은 QSYS 라이브러리에서 QADBxxxx라는 이름의 파일이며 이들 파일에 대해 뷰가 빌드되어 있음)에 대한 참조.
- 행 크기는 약 2KB를 초과합니다. 또는 SBMRMTCMD(리모트 명령 제출) 명령이나 저장된 프로시저어를 사용하여 디폴트 AS 데이터베이스 버퍼의 크기를 확장한 경우 행 크기는 OVRDBF(데이터베이스 파일로 대체) 명령 SEQONLY 레코드 수 매개변수의 스펙에서 나타나는 데이터베이스 버퍼 크기의 약 절반을 초과합니다. (OVRDBF 명령을 리모트로 작동하려면 OVRSCOPE(*JOB)을 지정해야 하는 점을 주지하십시오.)
- 커서의 화면이동이 가능하도록 선언되었으며(DECLARE...SCROLL CURSOR...) FETCH문에 지정된 화면 이동 옵션이 RELATIVE, PRIOR 또는 CURRENT 중 하나입니다(위에서 언급한 대로 복수 행 FETCH를 완료한 경우는 제외합니다).

관련 정보

SBMRMTCMD(리모트 명령 제출) 명령

OVRDBF(데이터베이스 파일로 대체) 명령

iSeries용 DB2 Universal Database 대 비iSeries용 DB2 Universal Database 블로킹:

다음의 경우 블로킹이 발생하지 않습니다.

- 커서가 갱신 가능합니다(주 1 참조).
- 커서가 잠재적으로 갱신 가능합니다(주 2 참조).
- ALWBLK(*NONE) 사전컴파일 옵션이 사용되었습니다.
- 행 크기가 약 16KB를 초과합니다.

비iSeries용 DB2 Universal Database 대 iSeries용 DB2 Universal Database 블로킹:

다음의 경우 블로킹이 발생하지 않습니다.

- 커서가 갱신 가능합니다(주 1 참조).
- 커서가 잠재적으로 갱신 가능합니다(주 2 참조).

- 사전컴파일 또는 바인드 옵션이 사용되며 단일 행 프로토콜이 되기 위한 패키지 디폴트 값을 지정하는 원인이 되었습니다.
 - iSeries용 DB2 Universal Database의 경우 이를 수행하기 위한 옵션은 제공되지 않습니다.
 - DB2 Universal Database for VM의 경우 SQLPREP의 NOBLOCK 키워드입니다(디폴트).
 - DB2 for OS/2®(DB2/2)의 경우 SQLPREP 또는 SQLBIND의 /K=NO입니다.
- 행 크기는 약 0.5*QRYBLKSIZ를 초과합니다. 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	32KB	32KB
DB2 Universal Database for z/OS	32KB	64KB
DB2 Universal Database for VM	8KB	32KB
Linux, UNIX 및 Windows용 DB2 Universal Database	32KB	64KB

최신 레벨의 분산 관계형 데이터베이스 구조(DRDA)에서 서버는 결과 세트를 리턴할 때 고정된 조회 블록 크기로 제한되지 않는 모드로 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 전용인 경우(주 3 참조) 및
 - 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.
- COMMIT(*ALL)을 지정하지 않고 다음 조건이 모두 해당되는 경우 블로킹이 발생합니다.
 - SELECT문 내에 FOR UPDATE OF절이 없습니다.
 - 프로그램 내의 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 명령문이 없습니다.
 - 프로그램이 동적 SQL문을 포함하지 않거나 사전컴파일/바인드 옵션을 사용하여 제한 블록 프로토콜을 요구했습니다(DB2 Universal Database for Linux, UNIX 및 Windows의 경우 /K=ALL, DB2 UDB for iSeries의 경우 ALWBLK(*ALLREAD), DB2 Universal Database for z/OS의 경우 CURRENTDATA(NO), DB2 Universal Database for VM의 경우 SBLOCK).

주:

1. 커서가 읽기 전용(주 3 참조)이 아니고 다음 조건 중 하나에 해당할 경우 갱신이 가능합니다.
 - SELECT문에 FOR UPDATE OF절이 들어 있습니다.
 - 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 구가 프로그램 내에 있습니다.
2. 커서가 읽기 전용(주 3 참조)이 아니고 프로그램에 EXECUTE 또는 EXECUTE IMMEDIATE 명령문(또는 비iSeries 서버에 연결된 경우 동적 명령문)이 있으며 사전 컴파일 또는 바인드 옵션을 사용하여 패키지 디폴트 값이 단일 행 프로토콜이 된 경우 이 커서는 잠재적으로 갱신 가능합니다.
 - iSeries용 DB2 Universal Database의 경우, ALWBLK(*READ) 사전컴파일 옵션입니다(디폴트).
 - DB2의 경우, BIND PACKAGE 상의 CURRENTDATA(YES)가 있습니다(디폴트).
 - DB2 Universal Database for VM의 경우 SQLPREP의 SBLOCK 키워드입니다.
 - DB2 for OS/2(DB2/2)의 경우 SQLPREP 또는 SQLBIND의 /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 않았습니다.
 - DECLARE CURSOR 명령문이 FOR FETCH ONLY절을 지정했습니다.
 - DECLARE CURSOR 명령문이 DYNAMIC를 포함하지 않고 SCROLL 키워드를 지정했습니다(i5/OS 전용).
 - 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 전용입니다(i5/OS 전용).

DRDA 조회 블록의 크기에 영향을 미치는 요소

조회에서 대량의 자료가 리턴될 경우, 조회 자료 블록의 크기를 증가시켜 성능을 개선할 수 있습니다. 이를 수행하는 방법은 조회에 참여하는 서버의 유형에 따라 다릅니다.

상이한 환경에서 조회 블록의 크기는 조회 열기 명령과 함께 송신된 매개변수에 의해 어플리케이션 리퀘스터에서 결정됩니다. iSeries 서버가 어플리케이션 리퀘스터(AR)일 때 초기에 필요한 조회 블록 크기는 항상 32KB입니다. 복수 블록을 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 SQLJ 및 JDBC, DB2 Universal Database for z/OS, DB2 Universal Database for VM 및 Linux, UNIX 및 Windows용 DB2 Universal Database의 디폴트 조회 블록 크기는 각각 32KB,

32KB, 8KB 및 32KB입니다. iSeries용 DB2 Universal Database 서버가 같은 종류가 아닌 어플리케이션 리퀘스터에 연결되었을 때 어플리케이션 리퀘스터로서 사용된 플랫폼에 대한 제품 문서를 참조하십시오.

iSeries용 DB2 UDB에서 iSeries용 DB2 UDB의 환경에서 조회 블록 크기는 데이터베이스 관리자가 사용하는 버퍼의 크기에 따라 결정됩니다. 디폴트 크기는 4KB입니다. 버전 2 릴리스 3 이상의 어플리케이션 서버에서는 이 크기를 변경할 수 있습니다. 크기를 변경하려면 SBMRMTCMD(리모트 명령 제출) 명령을 사용하여 어플리케이션 서버(AS)로 OVRDBF(데이터베이스 파일과 대체) 명령을 송신하여 이를 실행하십시오. 대체되는 파일 이름 외에 OVRDBF 명령은 OVRSCOPE(*JOB) 및 SEQONLY(*YES nnn)를 포함해야 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.

비iSeries 서버에서 또는 iSeries 서버 간에 SQL CALL문(저장 프로시저)을 사용하여 조회 블록 크기를 변경할 수도 있습니다.

관련 정보

SBMRMTCMD(리모트 명령 제출) 명령

OVRDBF(데이터베이스 파일로 대체) 명령

문제 해결

분산 관계형 데이터베이스에 액세스하는 중 문제점이 발생할 때 관리자는 문제점의 특성을 판별하고 그 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.

- i5/OS 라이선스 프로그램 지원에 대한 이해.
- 문제점이 어플리케이션 리퀘스터(AR)에 대한 것인지 또는 어플리케이션 서버(AS)에 대한 것인지 결정하는 방법.
- i5/OS 문제점 관리 기능에 대한 지식.

분산 관계형 데이터베이스에서 문제점을 진단하는 데 대한 자세한 방법은 *Distributed Relational Database Problem Determination Guide*, SC26-4782를 참조하십시오.

관련 개념

165 페이지의 『성능』

서버에서 실행되는 어플리케이션 프로그램의 종류에 관계없이 성능은 항상 주요 관심사입니다. 분산 관계형 데이터베이스의 경우, 네트워크, 서버 및 어플리케이션 성능이 모두 중요합니다.

iSeries 문제점 처리 개요

i5/OS 라이선스 프로그램을 사용하여 로컬 및 리모트 iSeries 서버에서 발생하는 사용자 및 시스템 감지 문제점을 모두 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 스프 파일도 작성됩니다. 오류 기록부 및 문제점 레코드는 다음 정보를 포함할 수 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 감지할 때 서버는 문제점 분석이 실행되거나 사용자가 Operational Assistant USERHELP 메뉴에서 문제점 해결을 위한 정보를 저장하는 옵션을 선택할 때까지 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.

관련 개념

196 페이지의 『시스템 및 통신 문제점』

시스템 또는 통신에 대한 문제점이 발생하면 메시지가 생성됩니다. 시스템 감지 문제점은 자동적으로 문제점 기록부로 입력되며 이곳에서 문제점이 보여지고 분석됩니다.

분산 관계형 데이터베이스 문제점 분리

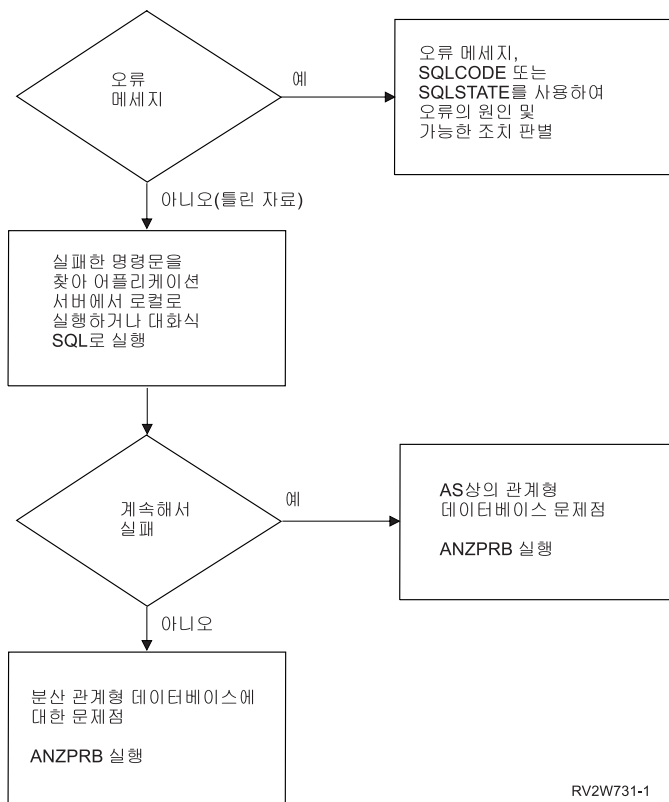
분산 관계형 데이터베이스 어플리케이션을 실행할 때 발생하는 문제점은 두 가지 일반적인 증상, 즉, 잘못된 출력 또는 어플리케이션이 예상된 시간에 완료되지 않는 증상을 나타낼 수 있습니다.

이 주제의 그림은 일반적으로 문제점을 어플리케이션 프로그램 문제점, 성능 관련 문제점, 서버 관련 문제점으로 분류하는 방법을 보여줍니다. 따라서 사용자는 표준 iSeries 서버 문제점 분석 방법을 사용하여 문제점을 해결할 수 있습니다.

DRDA 잘못된 출력 문제점

만일 오류 메시지를 수신했다면, 오류 메시지 SQLCODE 또는 SQLSTATE를 사용하여 문제점의 원인을 판별합니다.

다음 그림을 참조하십시오. 메시지 설명에서는 문제점이 무엇인지 표시하고 정확한 조치를 제공합니다. 오류 메시지가 수신되지 않으면 분산 관계형 데이터베이스가 실패를 일으키고 있는지 여부를 판별해야 합니다. 이를 수행하려면 어플리케이션 서버(AS)에서 실패하고 있는 명령문을 로컬로 실행하거나 대화식 SQL을 사용하여 어플리케이션 서버(AS)에서 명령문을 실행하십시오. 문제점을 로커로 작성할 수 있다면 이 문제점은 분산 관계형 데이터베이스 지원에 대한 것이 아닙니다. 이 조작의 결과에 따라 지원 스템에 대한 특정 정보를 제공하기 위해서는 iSeries 서버 문제점 분석을 사용하십시오.



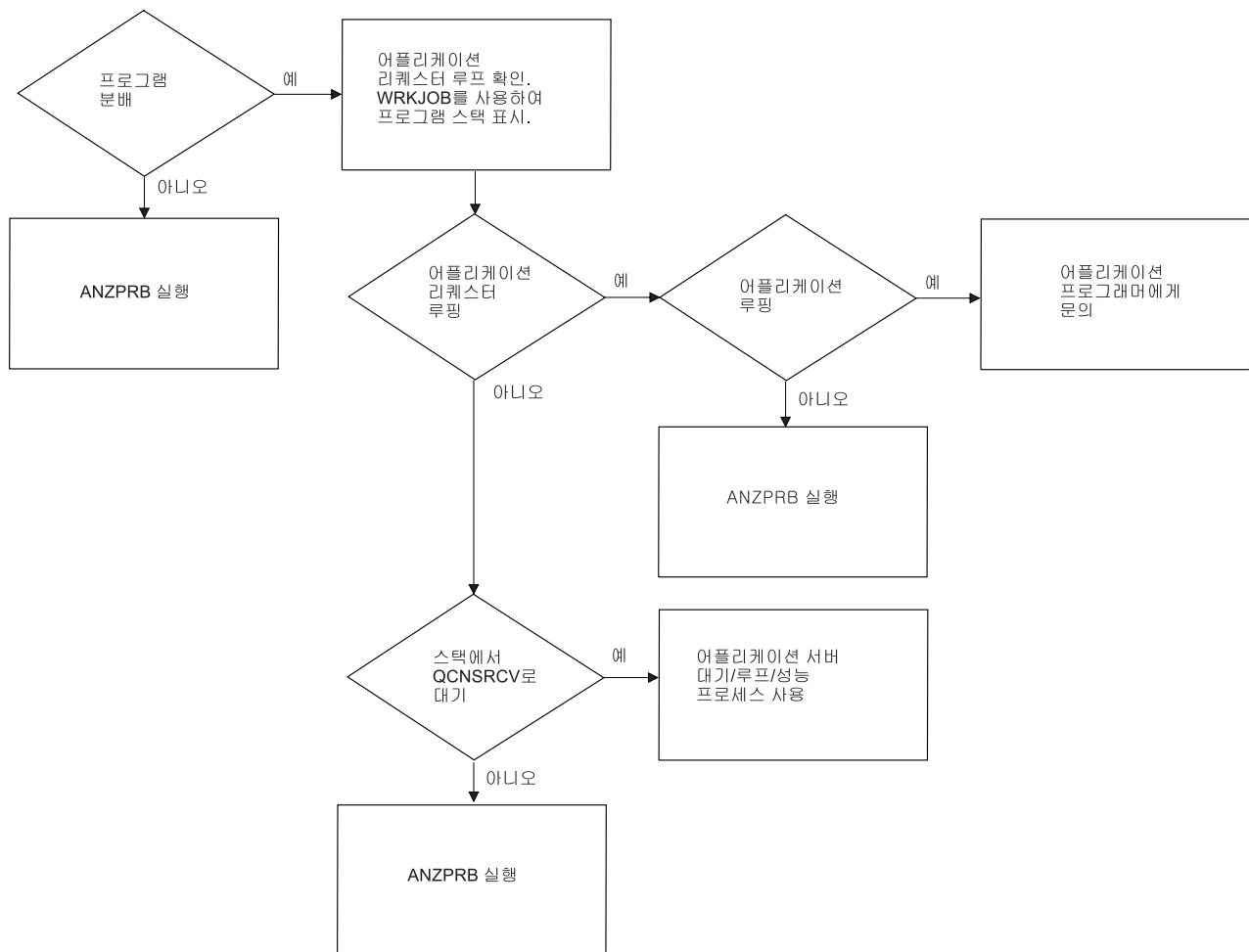
RV2W731-1

그림 16. 잘못된 출력 문제점 해결

어플리케이션이 예상 시간 내에 완료되지 않음

요구를 완료하는 데 예상된 시간보다 오래 걸리는 경우 먼저 어플리케이션 리퀘스터(AR)를 검사합니다.

작업 기록부에서 SQL7969 메시지를 검사하십시오. 이 메시지는 관계형 데이터베이스에 대한 연결이 완료되었음을 표시하며 어플리케이션이 분산 관계형 데이터베이스 어플리케이션임을 나타냅니다. WRKJOB(작업에 대한 작업) 명령을 사용하여 프로그램 스택을 표시하여 AR에서 루프를 검사하고 프로그램 스택을 검사하여 시스템이 루핑하는지 여부를 판별하십시오. 만일 어플리케이션 자체에 루프가 있다면 어플리케이션 프로그래머에게 솔루션을 의뢰하십시오. 스택에 QAPDEQUE 및 QCNSRCV가 있으면 AR이 어플리케이션 서버(AS)를 대기 중입니다. 만일 시스템이 통신 대기 상태에 있지 않다면, 문제점 분석 프로시저를 사용하여 성능에 문제점이 있는지 또는 다른 곳이 대기 상태에 있는지 확인하십시오.



RV2W732-2

그림 17. 어플리케이션 리퀘스터의 대기, 루프 또는 성능 문제점 해결

어플리케이션 서버 상의 작업 기록부를 검토하여 어플리케이션 리퀘스터 작업명을 찾을 수 있습니다. AS 작업을 검사해야 하는 경우 WRKJOB(작업에 대한 작업), WRKACTJOB(활동 작업에 대한 작업) 또는 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하여 AS의 작업을 찾으십시오.

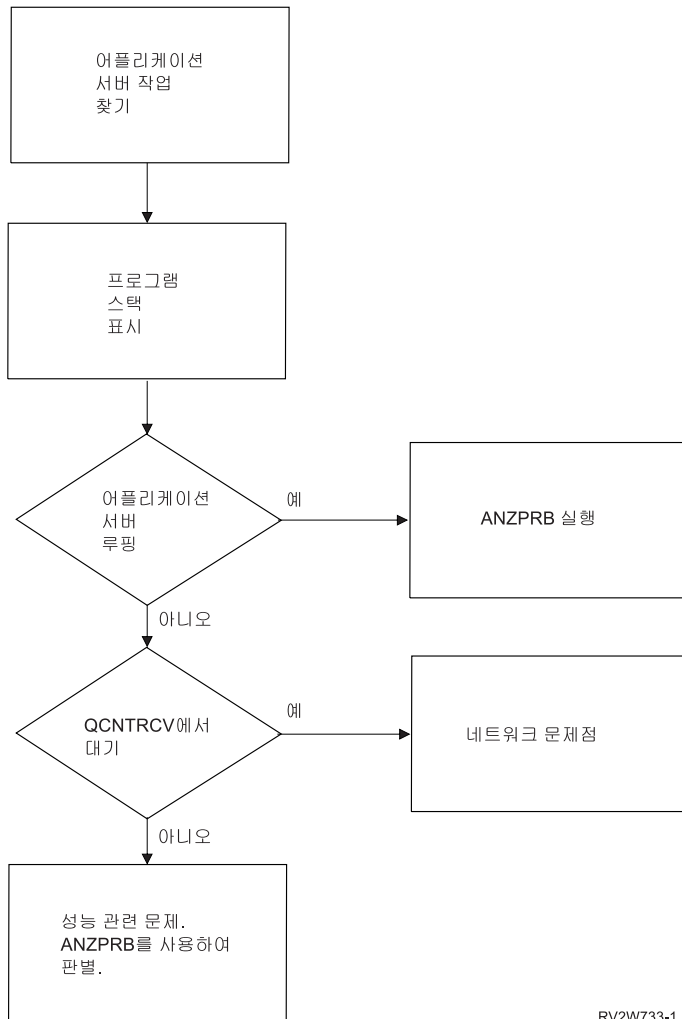
이 작업 표시장치 중 하나를 통하여, 어플리케이션 서버가 루핑하고 있는지 여부를 확인하기 위해 프로그램 스택을 점검하십시오. 루핑이 있다면, 문제점 처리를 위해 문제점 분석을 사용하십시오. 루핑이 없다면, QCNTRCV로 WAIT에 대한 프로그램 스택을 검사하십시오. QCNTRCV는 어플리케이션 서버가 어플리케이션 리퀘스터에 대해 대기 중이라는 뜻입니다. 서버 모두가 이 통신 대기 상태에 있다면, 네트워크 상에 문제점이 있다는 뜻입니다. AS가 대기 상태에 있지 않다면 검토해야 할 성능 문제점이 있다는 것입니다.

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.

Linux, UNIX 및 Windows용 DB2 Universal Database 또는 DB2 JDBC Universal Driver와 같은 제품을 사용하여 워크스테이션에서 iSeries용 DB2 Universal Database에 처음 연결할 때 iSeries용 DB2 UDB에 해당 제품의 SQL 패키지를 이미 작성하지 않은 경우 패키지가 자동으로 작성되며 NULLID 콜렉션도 자동으로 작성될 필요가 있을 수 있습니다. 이로 인해 초기 연결 후에 발행된 첫 번째 SQL문 중 하나에 대해 서버에서 응답을 수신할 때 약간의 지연이 발생합니다.

만일 TCP/IP를 통해 연결하려는 서버가 사용 불가할 경우, 상당한 지연이 발생할 것입니다. 수 분 동안 시간 종료 후 리모트 호스트가 시간종료 내에 응답하지 않았습니더라는 메시지가 표시됩니다. RDB 디렉토리 내의 IP 주소가 부정확할 경우도 이 작동의 원인이 됩니다.



RV2W733-1

그림 18. 어플리케이션 서버의 대기, 루프 또는 성능 문제점 해결

관련 개념

168 페이지의 『DRDA에 대한 블로킹에 영향을 미치는 요소』

어플리케이션 리퀘스터(AR)와 어플리케이션 서버(AS) 간 자료 전송 시 블로킹이 발생하는지 여부는 아주 중요한 성능 요소입니다. 자료 블록으로서 전송된 행의 그룹은 한 번에 하나의 행을 송신할 때보다 훨씬 적은 통신 오버헤드를 필요로 합니다.

관련 태스크

120 페이지의 『분산 관계형 데이터베이스 작업 찾기』

어플리케이션 리퀘스터(AR)에서 분산 관계형 데이터베이스 작업에 대한 정보를 찾을 때 사용되는 사용자 프로파일을 알고 있으면 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하여 해당 작업을 찾을 수 있습니다.

114 페이지의 『분산 관계형 데이터베이스의 작업에 대한 작업』

WRKJOB(작업에 대한 작업) 명령은 작업에 대한 작업 메뉴를 표시합니다. 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.

115 페이지의 『분산 관계형 데이터베이스의 사용자 작업에 대한 작업』

작업에서 사용되고 있는 사용자 프로파일(사용자명)을 알고 있는 경우 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하여 작업 정보를 표시하거나 변경할 수 있습니다. 사용자 프로파일을 사용하여 서버에 있는 작업 리스트를 얻으려면 어떠한 매개변수도 필요없이 이 명령을 입력하십시오.

116 페이지의 『분산 관계형 데이터베이스의 활동 작업에 대한 작업』

여러 사용자에 대해 실행 중인 작업을 모니터하려는 경우 또는 작업을 찾는 중이며 작업 이름이나 사용자 ID를 알지 못할 경우 WRKACTJOB(활동 작업에 대한 작업) 명령을 사용하십시오.

분산 관계형 데이터베이스 사용자에 대한 작업

문제점에 대한 연구를 처음 시작하는 것은 대부분 사용자입니다. 사용자는 프로그램을 실행할 때 예상된 결과를 얻지 못하거나 문제점을 나타내는 메시지를 받을 수 있습니다. 때때로 문제점을 진단하고 해결하는 가장 좋은 방법은 사용자와 함께 해당 프로시저어를 살펴보는 것입니다.

화면 복사 기능을 사용하면 사용자와 함께 실시간으로 이를 수행하거나 사용자가 이전에 본 표시장치의 파일을 검사할 수 있습니다.

단지 표시장치의 맨 아래에 표시되는 텍스트 행보다는 메시지에서 더 많은 정보를 수집할 수도 있습니다. 이 주제에서는 다른 사용자가 보고 있는 표시장치를 복사하는 방법 및 분산 관계형 데이터베이스 작업을 수행할 때 사용자가 수신하는 메시지에 대한 자세한 정보를 얻을 수 있는 방법을 설명합니다.

프로그래밍 문제점 외에도 APPC 또는 TCP/IP 사용 시 서버에 연결하거나 프로그램을 시작할 때 문제점이 있을 수도 있습니다.

화면 복사

STRCPYSCN(화면 복사 시작) 명령을 사용하면 워크스테이션에 사인 온하여 다른 워크스테이션에서 다른 사용자가 보고 있는 것과 동일한 화면을 볼 수 있습니다.

사용자와 똑같은 iSeries 서버를 시작해야만 합니다. 이 사용자가 리모트 서버에 있는 경우, 표시장치 pass-through를 사용하여 해당 서버에 사인 온 한 후 STRCPYSCN 명령을 입력하여 다른 표시장치를 볼 수 있습니다. 화면 이미지는 다른 워크스테이션으로 복사되거나 다른 워크스테이션을 사용할 수 없을 때 동시에 데이터베이스 파일로 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.

1. STRPASTHR(passthru 시작) 명령을 입력하십시오.

```
STRPASTHR RMTLOCNAME(KC105)
```

2. 어플리케이션 서버(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 *REQUESTER 명령을 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. 요구 장치로 이미지 송신을 시작하려면 g(이동)을 입력하십시오.

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.

소인 표시장치로부터 화면 복사 기능을 종료하려면 명령행에서 ENDCPYSCN(화면 복사 종료) 명령을 입력하고 Enter를 누르십시오.

```
ENDCPYSCN
```

화면 복사 기능을 시작할 때 봤던 화면이 보일 것입니다.

관련 정보

Pass-Through 시작(STRPASTHR) 명령

메세지

iSeries 서버에서는 간단한 입력 오류에서 서버 장치 또는 프로그램에 대한 문제점을 망라한 조건을 표시하는 시스템 메시지를 다양하게 송신할 수 있습니다.

다음과 같은 메시지가 표시될 수 있습니다.

- 현재 화면 상의 오류 메시지

이 메시지는 작업을 인터럽트시키거나 경보를 발생시킬 수 있습니다. 명령행에 DSPMSG를 입력하여 이들 메시지를 표시할 수 있습니다.

- 서버 오퍼레이터 메시지 대기행렬에 송신되었거나 분리된 메시지에 대한 작업 화면 상 표시된 서버 문제점에 관계된 메시지.

이들 메시지를 보려면 서버 명령행에서 DSPMSG QSYSOPR을 입력하십시오.

- 장치 설명에 명시된 메시지 대기행렬에 송신된 서버 문제점에 관계된 메시지.

이들 메시지를 보려면 서버 명령행에서 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.

일부 메시지 화면에서 메시지는 다음과 같은 문자 및 번호 코드와 함께 표시됩니다.

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 범주

범주	설명	라이브러리
CPA에서 CPZ까지	오퍼레이팅 시스템발 메시지	QSYS/QCPFMSG
MCH	라이선스 내부 코드 메시지	QSYS/QCPFMSG
SQ 및 SQL	SQL(Structured Query Language) 메시지	QSYS/QSQLMSG
TCP	TCP/IP 메시지	QTCP/QTCPMSGF

남아 있는 네 자리수(접두부가 SQ일 경우 다섯 자리수)는 메시지의 순번을 표시합니다. 보여진 메시지 ID 예에서는 오퍼레이팅 시스템에서 메시지 0083번임을 표시합니다.

메시지 대기행렬 또는 화면의 메시지 행에서 메시지에 대한 자세한 정보를 얻으려면 다음 단계를 따르십시오.

1. 커서를 메시지가 있는 같은 행으로 옮기십시오.
2. 도움말 키를 누르십시오. 추가 메시지 정보화면이 나타납니다.

추가 메시지 정보

메세지 ID :	CPD6A64	심각도 :	30
메세지 유형 :	진단		
송신 날짜 :	03/29/92	송신 시간. :	13:49:06
송신 프로그램 :	QUIACT	지시 :	080D
발신 프로그램 :	QUIMGFLW	지시 :	03C5

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 Enter 키를 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 얻으려면 DSPMSGD(메세지 설명 표시) 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
- 앞의 모든 항목

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 프로그램 및 덤프 매개변수에 대한 값(정의되었을 때)입니다. 이 정보를 사용하여 메세지가 나타날 때 사용자가 수행하는 작업을 판별할 수 있습니다.

관련 정보

DSPMSGD(메세지 설명 표시) 명령

메세지 유형:

추가 메세지 정보 화면에서 메세지 유형 및 심각도 코드를 볼 수 있습니다.

다음 표에서는 iSeries 메시지에 대한 다양한 메시지 유형 및 관련된 심각도 코드를 보여줍니다.

표 9. 메시지 심각도 코드

메시지 유형	심각도 코드
정보용 메시지. 정보용 목적 전용으로서라면 응답이 요구되지 않습니다. 메시지는 기능이 진행 중인지 또는 기능이 성공적으로 완료되었는 지를 표시합니다.	00
경고, 잠재 오류 조건이 존재합니다. 프로그램은 누락된 자료 공급과 같은 디폴트 작업을 수행했을 수 있습니다. 작업의 결과는 성공적인 것으로 전제합니다.	10
오류. 오류가 발견되었습니다. 하지만 자동 회복 프로시더어가 적용되었습니다. 처리가 계속되었습니다. 잘못된 자료를 대체하기 위해 디폴트 작업이 수행되었을 수 있습니다. 조작의 결과가 올바르지 않을 수 있습니다. 기능이 완료되지 않았을 수 있습니다. 예를 들어, 리스트의 몇 가지 항목이 올바로 실행되었지만 다른 항목이 올바로 실행되지 않았습니다.	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 종료로 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 비정상 종료. 작업이 시작되지 않았거나 시작하는 데 실패했습니다. 작업 레벨 기능이 요구된 대로 수행되지 않았거나 작업이 취소되었을 수 있습니다.	50
시스템 상태. 시스템 오퍼레이터 메시지 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 메시지 대기행렬에만 발행되었습니다.	70
시스템 경고 및 사용자 메시지. 비록 현재 시스템을 중단할 만큼 심각하지는 않지만 예방 측정이 취해지지 않는다면 조건이 더욱 심각해질 수 있습니다.	80
시스템 무결성. 시스템 오퍼레이터 메시지 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 메시지:

Application Server(AS) 또는 어플리케이션 리퀘스터(AS)에 오류 메시지가 발생할 경우 서버 메시지는 작업 기록부에 로그되어 실패의 이유를 표시합니다.

서버 메시지는 iSeries용 DB2 Universal Database 프로그램이 지원하는 SQL문에서 리턴된 각 SQLCODE에 대해 존재합니다. 이 메시지는 디버그 모드에서 실행될 때 사전컴파일러 리스팅, 대화식 SQL 또는 작업 기록부에서 사용 가능합니다. 그러나 iSeries 서버가 아닌 AS에서 작업하는 경우 다음 경우의 모든 오류 조건에 대해 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 SQLID는 리모트 오브젝트를 참조하는 명령문 내에서 사용될 수 없습니다란 존재하지 않습니다.

- 제품에서 기인한 오류이며 iSeries용 DB2 UDB를 사용하는 중에 결코 발생하지 않습니다.

iSeries용 DB2 UDB에는 결코 SQLCODE -925(SQLSTATE 56021), SQL 확약 또는 롤백이 IMS™ 또는 CICS 환경 내에서 유효하지 않습니다가 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 릴리즈에 상응하는 제품 문서를 참조하십시오.

CPx3E00에서 CPx3EFF까지 및 CPI9100에서 CPI91FF까지 범위의 메시지는 분산 관계형 데이터베이스 서버 메시지로 사용됩니다. 다음 리스트는 총괄적이지는 않지만 iSeries 서버의 분산 데이터베이스 작업 기록부에서 볼 수 있는 더 많은 일반 서버 메시지를 보여줍니다.

표 10. 분산 관계형 데이터베이스 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)
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.

표 10. 분산 관계형 데이터베이스 메시지 (계속)

MSG ID	설명
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 종료가 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.
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.

표 10. 분산 관계형 데이터베이스 메시지 (계속)

MSG ID	설명
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 사용자 yyy에게 서비스 중인 DDM 작업 xxxx(이는 QRWOPTIONS에 의해 억제될 수 있음).
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.

관련 개념

213 페이지의 『QRWOPTIONS 자료 영역』

DDM 또는 DRDA TCP/IP 서버 작업이 시작될 때 이들 작업은 사용자가 진단 및 기타 옵션을 지정할 수 있는 자료 영역을 찾습니다. 해당 이름은 QRWOPTIONS이며, 유효하려면 QGPL 라이브러리에 상주해야 합니다. 48자 스트링으로 이루어집니다.

관련 태스크

119 페이지의 『분산 관계형 데이터베이스의 작업 기록부를 사용하여 요구 정보 추적』

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

관련 정보

SQL 메시지 및 코드

APPC에 대한 프로그램 시작 요구 실패 처리

어플리케이션 서버(AS)의 i5/OS 서브시스템이 프로그램 시작 요구를 수신하면 서버는 프로그램 시작 요구와 함께 송신된 정보에 기초하여 작업을 시작하려고 시도합니다. 어플리케이션 서버(AS)에 대한 어플리케이션 리퀘스터(AR) 사용자의 권한과 요구된 데이터베이스의 존재 및 기타 여러 항목을 검사합니다.

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 메시지는 두 가지 이유 코드가 들어 있습니다(이유 코드 중 하나는 0일 수 있으며 이를 무시할 수 있습니다).

0이 아닌 이유 코드는 프로그램 시작 요구가 거부된 이유를 나타냅니다. 리모트 작업이 AS에서 시작되어야 하므로, 메시지 및 이유 코드는 어플리케이션 서버에 제공되며 어플리케이션 리퀘스터에는 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.

관련 정보

ICF 프로그래밍 PDF

TCP/IP에 대한 핸들 연결 요구 실패

TCP/IP용으로 구성된 분산 관계형 데이터베이스 구조(DRDA) 서버에서 연결 요구 실패의 기본 원인은 DDM TCP/IP 서버가 시작되지 않았거나 권한 오류가 발생했거나 기계가 실행되지 않는 것입니다.

서버가 시작되지 않았거나 포트 ID가 유효하지 않음:

DDM TCP/IP 서버가 시작되지 않은 경우 표시되는 오류 메시지는 CPE3425입니다.

메세지는 다음과 같습니다.

리모트 호스트가 연결 조작을 거부했습니다.

ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 또는 CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 명령에 잘못된 포트를 지정할 경우에도 이 메시지를 받을 수 있습니다. iSeries용 DB2 Universal Database 서버의 경우, 포트는 대개 *DRDA(DRDA 잘알려진 446 포트)입니다. 하지만 만일 IPsec를 사용하기 위해 포트 447로 구성했다면, 깨지기 쉬운 자료를 전송을 위해 그 포트를 사용할 수도 있습니다. SSL(Secure Sockets Layer)을 지원하는 DRDA 클라이언트를 사용하는 경우, 서버의 448 포트에 연결해야 합니다.

리모트 서버에서 DDM 서버를 시작하려면 STRTCPSVR(TCP/IP 서버 시작) *DDM 명령을 실행하십시오. CHGDMMTCPA(DDM TCP/IP 속성 변경) AUTOSTART(*YES) 명령을 실행하여 TCP/IP가 시작될 때마다 DDM 서버가 시작되도록 요구할 수 있습니다.

관련 정보

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

CHGDMMTCPA(DDM TCP/IP 속성 변경) 명령

관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 명령

DDM 및 DRDA에 대한 보안 소켓층

STRTCPSVR(TCP/IP 서버 시작) 명령

DRDA 연결 권한 실패:

권한 실패에 대해 주어진 오류 메시지는 SQ30082입니다.

메세지 텍스트는 다음과 같습니다.

분배 데이터베이스 연결 시도 상 권한 실패

메세지의 원인 섹션에서는 이유 코드 및 가능한 이유 코드에 대한 의미 리스트를 나타내고 있습니다. 이유 코드 17은 지원되지 않는 보안 체계가 있다는 의미입니다(SECMEC).

iSeries용 DB2 Universal Database는 iSeries 어플리케이션 리퀘스터(AR)가 사용할 수 있는 여러 가지 분산 관계형 데이터베이스 구조(DRDA) SEMEC를 구현합니다.

- 사용자 ID 전용
- 암호가 있는 사용자 ID
- 암호화된 암호 보안 메카니즘
- Kerberos(V5R2)

암호화된 암호는 연결이 개시될 때 암호가 사용될 수 있는 경우에만 송신됩니다.

iSeries 서버에 대한 디폴트 SECMEC를 사용하려면 암호가 있는 사용자 ID가 필요합니다. 어플리케이션 리퀘스터가 디폴트 보안 구성을 사용하여 암호가 없는 사용자 ID를 서버로 송신할 경우, 이유 코드가 17인 오류 메시지 SQ30082가 표시됩니다.

다음은 지원되지 않는 보안 메카니즘 실패에 대한 솔루션입니다.

- 클라이언트가 서버의 신뢰를 받고 있으며 인증이 필요하지 않을 경우 DDM TCP/IP 서버의 인증 설정을 암호가 필요하지 않은 것으로 변경하십시오.
- 클라이언트가 서버의 신뢰를 받지 않으며 인증이 필요한 경우 어플리케이션을 암호 또는 인증된 보안 토큰(예: Kerberos 토큰)을 송신하도록 변경하십시오.

DDM TCP/IP 서버의 인증 설정은 CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령을 사용하거나 Operation Navigator의 네트워크 → 서버 → **DDM** 등록 정보를 사용하여 변경할 수 있습니다.

SQL CONNECT문의 USER/USING 형식을 사용하거나 ADDSVRAUTE(서버 인증 항목 추가) 명령을 사용하여 암호를 송신하여 연결 시도가 작성된 사용자 프로파일의 서버 권한 항목에 리모트 사용자 ID 및 암호를 추가할 수 있습니다. V4R5 이후 시스템에서는 암호화된 암호를 송신하려는 시도가 자동으로 실행됩니다. V4R5 이전의 iSeries 서버는 암호화된 암호를 송신하거나 V4R5 iSeries AR이 송신한 암호화된 암호 유형을 해독할 수 없는 점을 주지하십시오.

반드시 시스템 값 QRETSVRSEC(서버 보안 자료를 유지)를 '1'로 설정하여 서버 권한 항목 내에 리모트 암호를 저장할 수 있도록 해야합니다.

경고: DRDA에 사용하기 위해서는 ADDSVRAUTE(서버 인증 항목 추가) 명령에서 RDB 이름을 대문자로 입력해야 합니다. 그렇지 않으면 연결 처리 중 이름이 인식되지 않으며 권한 항목의 정보가 사용되지 않습니다.

관련 정보

ADDSVRAUTE(서버 인증 항목 추가) 명령

CHGDDMTCPA(DDM TCP/IP 속성 변경) 명령

서버를 사용할 수 없음:

리모트 서버가 가동되어 실행되지 않는 경우 또는 어플리케이션 서버(AS)에 대한 RDB 디렉토리 항목에 잘못된 IP 주소를 지정할 경우 CPE3447 메시지가 표시됩니다.

이 메시지 텍스트는 다음과 같습니다.

리모트 호스트가 시간 종료 내에 응답하지 않았습니다.

이 메시지가 발생하기 전에 일반적으로 몇 분간의 지연이 있습니다. 해당 시간 동안 무엇인가 정지 중이거나 루핑 중인 것으로 나타날 수 있습니다.

대화식 SQL 특정 연결 실패:

대화식 SQL에서 CONNECT문을 실행할 때 이따금 일반 SQ30080 메시지가 나타납니다.

이 메시지의 텍스트는 다음과 같습니다.

분산 데이터베이스 처리 중 통신 오류가 발생했습니다.

오류의 세부사항을 보려면 대화식 SQL을 종료하고 작업 기록부를 살펴보아야 합니다.

1단계 확약 기능만 있는 서버에 처음 연결할 때(지정된 레벨의 확약 제어의 경우) SQL7020 메시지 SQL 패키지 작성에 실패했습니다가 표시될 경우 이는 읽기 전용 서버인 리모트 서버에 액세스했으며 SQL 패키지를 작성하려면 이를 갱신해야 하기 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.

관련 태스크

44 페이지의 『대화식 SQL에 대한 SQL 패키지 설정』

이 주제는 비iSeries 어플리케이션 서버(AS)에만 적용됩니다.

서버에서 사전시작 작업이 충분하지 않음:

TCP/IP 서버와 연관된 사전시작 작업 수가 QSYSWRK 서브시스템의 QRWTSRVR 사전시작 작업 항목에 의해 제한되며 모든 사전시작 작업이 연결에 사용되고 있는 경우 새 연결에서의 시도는 다음 메시지와 함께 실패합니다.

CPE3426

소켓에 의해 재설정된 리모트 소켓과 연결

CPD3E34

recv() — MSG_PEEK에 DDM TCP/IP 통신 오류 발생.

QWTWSRVR 항목에 대한 CHGPJE(사전시작 작업 항목 변경) 명령의 MAXJOBS 매개변수를 더 높은 숫자 또는 *NOMAX로 설정하고 ADLJOBS 매개변수를 0이 아닌 다른 값으로 설정하면 서버에서 이 문제점이 발생하지 않게 할 수 있습니다.

관련 정보

CHGPJE(사전시작 작업 항목 변경) 명령

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 어플리케이션 서버(AS)의 작업 기록부에서 패키지가 실패한 것을 알 수 있으며 프로그램 또는 패키지의 리스팅을 통해 실패 원인을 알 수 있습니다. SQL 컴파일러는 진단 리스팅에서 사전컴파일 프로세스로 생성된 SQLCODE 및 SQLSTATE를 표시하는 진단 테스트를 제공합니다.

ILE(Integrated Language Environment®) 사전컴파일의 경우 선택적으로 OPTION(*XREF) 및 OUTPUT(*PRINT)을 지정하여 사전컴파일 소스 및 상호 참조 리스팅을 인쇄할 수 있습니다. 비ILE 사전컴파일의 경우 CRTSQLxxx(SQL 프로그램 작성) 명령의 OPTIONS 매개변수에 *SOURCE 및 *XREF를 지정하여 사전컴파일 소스 및 상호 참조 리스팅을 인쇄할 수 있습니다.

리스팅

CRTSQLxxx(SQL 프로그램 작성) 명령으로 생성된 리스팅은 다음과 같은 정보를 제공합니다.

- 사전컴파일 명령의 매개변수에 대해 제공된 값
- 프로그램 소스
- ID 중복 참조
- 사전컴파일로부터 비롯한 메시지

사전컴파일러 리스팅:

```

5722ST1 V5R4M0 060210          Create SQL ILE C Object UPDATEPGM 02/10/06 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/QSYSPRT
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 02/10/06 14:25:33

```

```

5722ST1 V5R4M0 060210          Create SQL ILE C Object UPDATEPGM 02/10/06 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 19. 사전컴파일러의 리스팅

그림 20. 사전컴파일러의 리스팅(계속)

```

5722ST1 V5R4M0 060210          Create SQL ILE C Object UPDATEPGM 02/10/06 14:30:10   Page    3
CROSS REFERENCE
Data Names                    Define   Reference
DEPTCODE                      ****    COLUMN
18
DPT1                          ****    TABLE IN RWDS
17
RWDS                          ****    COLLECTION
17
5722ST1 V5R4M0 060210          Create SQL ILE C Object UPDATEPGM 02/10/06 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
* * * * * E N D O F L I S T I N G * * * * *

```

CRTSQLPKG 리스팅:

CRTSQLPKG(SQL 패키지 적성) 명령으로 생성된 리스팅은 여러 가지 유형의 정보를 제공합니다.

- 명령 매개변수에 사용된 값
- 오류를 초래한 명령문(있을 경우)
- CRTSQLPKG(SQL 패키지 작성) 명령의 실행으로 생성된 메시지

다음 그림은 이 정보를 나타냅니다.

```

5722ST1 V5R4M0 060210          Create SQL package 02/10/06 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

5722ST1 V5R4M0 060210          Create SQL package 02/10/06 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
* * * * * E N D O F L I S T I N G * * * * *

```

그림 21. CRTSQLPKG에서 비롯한 리스팅

관련 정보

SQL 패키지 작성(CRTSQLPKG) 명령

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 제공하는 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 감지될 때 SQLSTATE라는 5자 글로벌 변수가 오류의 특성을 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 관계없이 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). 명령문 처리 중 SQL에서 오류 또는 예외 조건이 발생하지 않을 경우 SQLCODE는 0입니다.

니다. 모든 iSeries용 DB2 Universal Database SQLCODE는 QSYS 라이브러리의 QSQLMSG 메시지 파일에 해당 메시지를 가지고 있습니다. 예를 들어, SQLCODE -204는 메시지 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. 아래에 설명된 메시지 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.

DSPMSGD(메세지 설명 표시) 명령을 사용하여 전체 메시지를 온라인으로 볼 수 있습니다.

관련 정보

SQL 참조

SQL 메세지 및 코드

DSPMSGD(메세지 설명 표시) 명령

분산 관계형 데이터베이스 SQLCODE 및 SQLSTATE:

이 리스트는 분산 관계형 데이터베이스 처리와 연관된 일반적인 SQLCODE 및 SQLSTATE 중 일부를 나열합니다.

이러한 SQLCODE(및 연관된 SQLSTATE)의 간략한 설명에서 메시지 자료 필드는 앰퍼샌드(&)와 숫자(예: &1)로 식별됩니다. 이들 필드에 대한 대체 텍스트는 어플리케이션 프로그램이 SQLCA를 사용할 경우 SQLERRM에 저장되고, SQL 진단 영역을 사용할 경우 DB2_ORDINAL_TOKEN_n(여기서 n은 토큰 번호임)에 저장됩니다. DSPMSGD(메세지 설명 표시) 명령을 사용하여 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;

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
+304	01515	이 SQLSTATE는 호스트 변수가 검색된 값을 보유할 정도로 크지 않기 때문에 호스트 변수 리스트 또는 구조에 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 &가 없습니다.
+552	01542	&1에 대한 권한이 없습니다.
+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 매개변수가 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.
-351 -352	56084	지원되지 않는 SQLTYPE이 선택 리스트 또는 입력 리스트에서 발견되었습니다.
-426	2D528	어플리케이션 실행 환경에 유효하지 않은 조각입니다. 이 SQLSTATE는 동적 COMMIT 제한 환경에서 COMMIT를 실행하기 위해 EXCSQLIMM 또는 EXCSQLSTT를 사용하려는 시도가 있었다고 보고합니다.
-427	2D529	어플리케이션 실행 환경에 유효하지 않은 조각입니다.

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
-501 -502 -507	24501	유효하지 않은 커서 상태로 인해 실행이 실패했습니다. 식별된 커서가 열리지 않습니다.
-510	42828	이 SQLSTATE는 블로킹 프로토콜을 이용하여 행을 폐치하는 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 대한 연결 처리 중 사용자를 인증하는 데 실패했음을 보고합니다.
-925 -926	2D521	현재 환경에서 SQL COMMIT 또는 ROLLBACK문은 유효하지 않습니다.
-950	42705	관계형 데이터베이스(RDB) &1이 관련 디렉토리에 존재하지 않습니다.
-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.

표 11. SQLCODE 및 SQLSTATE (계속)

SQLCODE	SQLSTATE	설명
-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.
-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.

관련 정보

SQL 메시지 및 코드

DSPMSGD(메시지 설명 표시) 명령

시스템 및 통신 문제점

시스템 또는 통신에 대한 문제점이 발생하면 메시지가 생성됩니다. 시스템 감지 문제점은 자동적으로 문제점 기록부로 입력되며 이곳에서 문제점이 보여지고 분석됩니다.

시스템 명령행에서 ANZPRB(문제점 분석) 명령을 입력하여 기록된 문제점에 대한 문제점 분석을 언제든지 실행할 수 있습니다. 이 명령으로 분석 프로시저어를 실행하고 문제점 기록부 내의 추가적인 문제점 관련 정보를 저장할 수 있습니다.

문제점 기록부를 보려면 WRKPRB(문제점에 대한 작업) 명령을 사용하십시오. 다음의 화면들은 문제점 기록부의 두 가지 뷰를 나타내고 있습니다.

문제점에 대한 작업

시스템: KC000
 위치: 문제점 ID
 옵션을 입력한 후 Enter 키를 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 Enter 키를 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. 다음 항목과 같이 특정 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 취해져야할 지 판별할 수 있습니다.

관련 개념

172 페이지의 『iSeries 문제점 처리 개요』

i5/OS 라이선스 프로그램을 사용하여 로컬 및 리모트 iSeries 서버에서 발생하는 사용자 및 시스템 감지 문제점을 모두 관리할 수 있습니다.

관련 정보

문제점 분석(ANZPRB) 명령

WRKPRB(문제점에 대한 작업) 명령

실패 보고를 위한 자료 수집

이 주제는 iSeries 서버의 분산 관계형 데이터베이스에서 문제점 진단에 도움이 되도록 인쇄할 수 있는 자료의 유형을 설명합니다.

이 자료는 i5/OS 라이선스 프로그램에 의해 생성됩니다. 또한 시스템 오퍼레이터 메시지 및 어플리케이션 프로그램(그 자료와 함께)을 사용하여 문제점을 진단할 수 있습니다.

작업 기록부 인쇄

iSeries 서버 상의 모든 작업에는 그 작업에 대해 입력된 요구와 관련된 정보를 수록한 작업 기록부가 있습니다. 사용자가 어플리케이션 리퀘스터(AR)에서 문제점이 있는 경우 작업 기록부의 정보가 문제점 진단에 도움이 될 수 있습니다.

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.

어플리케이션 서버(AS)에 대한 작업 기록부도 문제점 진단에 도움이 될 수 있습니다.

관련 태스크

119 페이지의 『분산 관계형 데이터베이스의 작업 기록부를 사용하여 요구 정보 추적』

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

120 페이지의 『분산 관계형 데이터베이스 작업 찾기』

어플리케이션 리퀘스터(AR)에서 분산 관계형 데이터베이스 작업에 대한 정보를 찾을 때 사용되는 사용자 프로파일을 알고 있으면 WRKUSRJOB(사용자 작업에 대한 작업) 명령을 사용하여 해당 작업을 찾을 수 있습니다.

TCP/IP 서버 사전시작 작업에서 작업 기록부 찾기

분산 자료 관리(DDM) TCP/IP 서버와 연관된 QRWTSRVR 사전시작 작업 중 하나가 서비스하는 연결이 종료될 때 사전시작 작업은 다른 연결에서 사용되도록 재순환됩니다. 이 상황이 발생하면, 종료된 연결과 관련된 작업 기록부는 지워집니다.

하지만 어떤 경우에는, 작업 기록부가 지워지기 전에 프린터 파일로 스푼됩니다. 클라이언트 사용자 ID 및 암호의 유효성이 확인되지 않을 경우 작업 기록부는 스푼 파일에 인쇄되지 않습니다. 유효성이 확인된 경우 작업 기록부가 스푼 파일에 인쇄되는 조건은 다음과 같습니다.

- 오퍼레이팅 시스템이 V5R1 이상이고 서버 작업의 메시지 로깅 텍스트 레벨이 *SECLVL 또는 *MSG일 경우.
- 연결을 종료한 요구를 처리하는 중 서버 요구 핸들러 라우팅 프로그램이 심각한 오류가 발생했음을 감지할 경우.
- 사전시작 작업이 서비스되는 경우(STRSRVJOB(서비스 작업 시작) 명령을 사용하여).
- 클라이언트 또는 서버의 QRWOPTIONS 자료 영역이 서버 작업이 충족시킨 작업 기록부 출력 조건을 지정한 경우.

여러 가지 이유로 작업 기록부 정보를 얻으려 할 수 있습니다. 이는 분명 오류를 진단하는데 유용합니다. 또 성능에 관련된 문제점을 분석하는데도 유용합니다. 예를 들어, 디버그 모드에서 실행할 때 생성되는 SQL Optimizer 자료를 얻으려는 경우 수동으로 서비스 작업을 시작하고 STRDBG(디버그 시작) 명령을 실행하거나 QRWOPTIONS 자료 영역에서 적용 가능한 옵션을 하나 이상 설정하여 작업 기록부가 보유되게 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
```

여기서 *userid*는 어플리케이션 서버에 대한 CONNECT에서 사용되는 사용자 ID입니다. 사용자 ID를 모를 경우 AS에서 DSPLOG(기록부 표시) 명령을 사용하여 사용자 ID를 찾을 수 있습니다.

다음과 같이 매개변수 사용으로 원하지 않는 메시지를 필터링 할 수 있습니다.

```
DSPLOG PERIOD(('11:00')) MSGID(CPI3E34)
```

다음 메시지를 찾으십시오. 그러나 이 메시지(CPI3E34)를 억제하기 위해 QRWOPTIONS 자료 영역을 사용한 경우 이력 기록부에 메시지가 나타나지 않는다는 점을 주의하십시오.

```
ddd, ttt에 yyy 사용자를 서버스하는 DDM 작업 xxxxx
```

관련 개념

213 페이지의 『QRWOPTIONS 자료 영역』

DDM 또는 DRDA TCP/IP 서버 작업이 시작될 때 이들 작업은 사용자가 진단 및 기타 옵션을 지정할 수 있는 자료 영역을 찾습니다. 해당 이름은 QRWOPTIONS이며, 유효하려면 QGPL 라이브러리에 상주해야 합니다. 48자 스트링으로 이루어집니다.

관련 정보

서비스 작업 시작(STRSRVJOB) 명령

DSPLOG(기록부 표시) 명령

디버그 시작(STRDBG) 명령

제품 활동 기록부 인쇄

iSeries 서버의 제품 활동 기록부는 기계 점검, 장치 오류 및 테이프 통계의 레코드입니다. 오류 기록부에는 각 FFDC 덤프의 처음 1000바이트를 포함하는 FFDC 정보도 수록되어 있습니다. 이들 오류를 검토하면 문제점의 특성을 판별할 수 있습니다.

사인 온한 서버의 제품 활동 기록부를 인쇄하려면 다음 단계를 수행하십시오.

1. 명령행에 PRTERLOG(오류 기록부 인쇄) 명령을 입력하고 F4(프롬프트)를 누르십시오. 오류 기록부 인쇄 화면이 나타납니다.
2. 인쇄를 원하는 기록부 정보 유형에 대한 매개변수를 입력하고 Enter 키를 누르십시오. 기록부 정보가 작업에 대해 식별된 출력 대기행렬에 송신됩니다.
3. WRKJOB(작업에 대한 작업) 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.

관련 정보

PRTERLOG(오류 기록부 인쇄) 명령

WRKJOB(작업(Job)에 대한 작업) 명령

작업 추적

때때로 어떤 문제점은 특정 프로그램에 대해 추적이 불가능합니다. 이러한 경우 STRTRC(추적 시작) 및 TRCJOB(작업 추적) 명령을 사용하여 실행된 CL 명령과 i5/OS 자료 획득 및 모듈 흐름을 추적할 수 있습니다.

이들 틀은 문제점 분석 프로시더가 문제점에 대한 충분한 정보를 공급하지 못할 때 사용되어야 합니다. 분산 데이터베이스 애플리케이션의 경우, 이들 명령은 또한 분산 데이터베이스 요구 및 응답 자료 스트림을 캡처하는 데 유용합니다.

어플리케이션 서버 작업의 작업 추적을 얻어야 하는 경우 서버에서 서비스 작업을 시작해야 합니다.

관련 개념

209 페이지의 『어플리케이션 서버 문제점을 진단하기 위한 서비스 작업 시작』

어플리케이션이 분산 관계형 데이터베이스 구조(DRDA)를 사용할 때 SQL문은 어플리케이션 서버 작업 내에서 실행됩니다. 따라서 i5/OS 오퍼레이팅 시스템에서 실행 중인 어플리케이션 서버 작업에 대해 작업 추적 또는 디버그를 시작할 필요가 있을 수 있습니다. 이를 수행하기 위한 기술은 APPC를 사용하는지 또는 TCP/IP를 사용하 지에 따라 다릅니다.

202 페이지의 『통신 추적』

분산 관계형 데이터베이스 구조(DRDA)를 사용하여 분산 관계형 데이터베이스에 액세스할 때 CPF3Exx 범위 또는 CPF91xx 범위의 메시지를 얻은 경우 통신 추적을 실행해야 합니다.

추적 작업:

TRCJOB(추적 작업) 명령은 두 가지 추적 톨 중에서 더 오래된 톨입니다. 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.

관련 개념

261 페이지의 『추적 작업 및 FFDC 자료 해석』

이 주제에서는 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.

관련 정보

TRCJOB(작업 추적) 명령

추적 시작:

또한 STRTRC(추적 시작) 명령을 사용하여 추적을 수행할 수도 있습니다. STRTRC 명령은 TRCJOB(작업 추적) 명령보다 좀더 융통적이며 덜 강제적입니다. 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 세트에 저장됩니다. 그런 다음 이들 파일은 스펴 프린터 파일(QPSRVTRCJ)에 기록되거나 PRTRC를 발행하여 데이터베이스 출력 파일로 보내질 수 있습니다.

샘플 추적 시나리오는 다음과 같습니다.

```
STRTRC SSNID(DRDATRACE) JOB((*ALL/QUSER/QRWTSRVR)) MAXSTG(160000)
TRCFULL(*STOPTRC)
```

실패한 DRDA 시나리오 실행:

```
ENDTRC SSNID(DRDATRACE) DTALIB(TRACELIB)
PRTTTC DTAMBR(DRDATRACE) DTALIB(TRACELIB)
```

관련 정보

ENDTRC(추적 종료) 명령

STRTRC(추적 시작) 명령

통신 추적

분산 관계형 데이터베이스 구조(DRDA)를 사용하여 분산 관계형 데이터베이스에 액세스할 때 CPF3Exx 범위 또는 CPF91xx 범위의 메시지를 얻은 경우 통신 추적을 실행해야 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. TRCTCPAPP 함수는 IPSec 또는 보안 소켓 프로토콜이 자료 스트림을 암호화한 위치를 추적하는 지능 추적을 제공합니다. 암호화 이전, 그리고 암호해제 이후 자료를 캡처합니다. 또한 암호화되지 않은 자료 스트림의 추적도 잘 수행합니다. LOOPBACK이 사용되는 시스템 내부의 DRDA 흐름을 추적하는 데 필수적입니다.

관련 개념

200 페이지의 『작업 추적』

때때로 어떤 문제점은 특정 프로그램에 대해 추적이 불가능합니다. 이러한 경우 STRTRC(추적 시작) 및 TRCJOB(작업 추적) 명령을 사용하여 실행된 CL 명령과 i5/OS 자료 획득 및 모듈 흐름을 추적할 수 있습니다.

관련 태스크

205 페이지의 『TCP/IP 통신 추적』

추적 툴을 사용하는 한 가지 방법은 자료가 암호화된 환경에서 clear text 전송을 표시하는 것입니다.

관련 정보

69 페이지의 『DDM 및 DRDA에 대한 연결 보안 프로토콜』

TCP/IP를 통한 분산 자료 관리(DDM) 또는 분산 관계형 데이터베이스 구조(DRDA)의 현재 iSeries용 DB2 UDB 구현은 여러가지 연결 보안 프로토콜을 지원합니다.

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.

다음 상황에 대해서 통신 추적을 사용해야만 합니다.

- 문제점 분석 프로시듀어가 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. DRDA 자료 스트림을 해석하는 데 대한 정보는 262 페이지의 『예: RW 추적 자료 분석』의 내용을 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.

TCP/IP 추적의 출력에서 원하지 않는 자료가 출력되지 않도록 하려면 소스 및 어플리케이션 서버(AS)의 IP 주소를 지정할 수 있습니다. 간혹 간단하게 포트 번호를 대신 지정해도 충분합니다.

다음 명령이 통신 추적을 시작, 중단, 인쇄 및 삭제합니다.

STRCMNTRC(통신 추적 시작) 명령

지정된 행 또는 네트워크 인터페이스 설명에 대한 통신 추적을 시작합니다. 추적 매개변수에서 바이트 수의 시작 바이트 값으로 *MAX를 지정하십시오. 통신 추적은 ENDCMNTRC(통신 추적 종료) 명령을 실행할 때까지 계속 수행됩니다.

ENDCMNTRC(통신 추적 종료) 명령

지정된 행 또는 네트워크 인터페이스 설명에서 실행된 통신 추적을 종료합니다.

PRTCMNTRC(통신 추적 인쇄) 명령

지정된 행 또는 네트워크 인터페이스 설명에 대한 통신 추적 자료를 스푼 파일 또는 출력 파일에 이동시킵니다. 포맷 SNA(Systems Network Architecture) 자료 중 매개변수에만 *YES를 지정하십시오.

DLTCMNTRC(통신 추적 삭제) 명령

지정된 행 또는 네트워크 인터페이스 설명에 대한 통신 추적을 삭제합니다.

관련 정보

통신 관리 PDF

DLTCMNTRC(통신 추적 삭제) 명령

ENDCMNTRC(통신 추적 종료) 명령

PRTCMNTRC(통신 추적 인쇄) 명령

STRCMNTRC(통신 추적 시작) 명령

회선, 제어기 및 장치 설명 찾기:

어플리케이션 서버 작업이 시작되는 제어기 및 장치를 찾으려면 WRKCFGSTS(구성 상태에 대한 작업) 명령을 사용하십시오.

예를 들면, 다음과 같습니다.

```
WRKCFGSTS CFGTYPE(*DEV)
                CFGD(*LOC)
                RMTLOCNAME(DB2ESYS)
```

RMTLOCNAME 키워드에 대한 값은 어플리케이션 서버명입니다.

WRKCFGSTS(구성 상태에 대한 작업) 명령은 지정된 서버명이 리모트 위치명인 모든 장치를 표시합니다. 한번에 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. WRKCFGSTS 명령을 사용하여 제어기 및 장치 설명에 대한 작업을 수행할 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 주제의 첫 번째 예의 장치 설명으로부터 확보된 제어명입니다.

WRKCFGSTS(구성 상태에 대한 작업) 명령의 출력에는 통신 추적에 대해 작업할 때 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 서버의 모든 회선 설명을 나열하는 WRKLIND(회선 설명에 대한 작업) 명령을 사용하는 것입니다.

관련 정보

WRKCFGSI(구성 상태에 대한 작업) 명령

WRKLIND(회선 설명에 대한 작업) 명령

TCP/IP 통신 추적

추적 툴을 사용하는 한 가지 방법은 자료가 암호화된 환경에서 clear text 전송을 표시하는 것입니다.

추적 자료는 송신지측에서 암호화하기 전과 수신지측에서 암호화한 후에 캡처됩니다. 그러나 이 추적 툴은 다른 환경에서도 유용합니다. 통신에 TCP/IP를 사용할 때만 이 기능을 사용할 수 있습니다.

TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령을 사용하려면 *SERVICE 특수 권한이 있는 사용자 프로파일 이 있어야 합니다. 추적을 시작하려면 다음 행을 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 어플리케이션에 대해 TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령을 사용할 때, 단일 송신 또는 수신 메시지에 대해 추적할 수 있는 최대 자료량은 6000바이트로 제한됩니다.

관련 개념

202 페이지의 『통신 추적』

분산 관계형 데이터베이스 구조(DRDA)를 사용하여 분산 관계형 데이터베이스에 액세스할 때 CPF3Exx 범위 또는 CPF91xx 범위의 메시지를 얻은 경우 통신 추적을 실행해야 합니다.

관련 정보

TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령

TCP/IP 통신 추적 포매팅:

TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령을 사용하여 DRDA 및 DDM 흐름을 읽기 쉬운 논리 표시로 구분할 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 주제의 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 E2D8F0F5F0F3F000 .....éëë..... 400....QSQ05030.
0A00350006119C00 2500062121241E00 C...L.ó...LbbfK. ....ä.....
062120241E0010D1 2A01000000000000 LbafK..çk..... .....J.....
0000000000001621 35C1D7D7D54BD3D7 .....b. &&+.<& .....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: éāç 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: TYPDEF0VR - 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

LL: 5 CP: 213B

BOOLEAN: TRUE

관련 정보

TRCTCPAPP(TCP/IP 어플리케이션 추적) 명령

첫실패 자료 캡처 자료

이 주제는 iSeries 서버에서 첫실패 자료 캡처(FFDC) 자료를 찾는 방법에 대한 추가 정보를 설명합니다. 이 정보는 어플리케이션 서버(AS)에서 FFDC 자료 출력을 일으키는 실패가 발생할 경우에 가장 유용합니다. 어플리케이션 리퀘스터(AR)에 대한 FFDC 자료는 보통 어플리케이션 프로그램을 실행하는 작업과 연관된 스펴 파일 중 하나에서 찾을 수 있습니다.

주: FFDC 자료는 QSFWERRLOG 시스템 값이 *LOG로 설정된 경우에만 생성됩니다.

1. DSPMSG(메세지 표시) QSYSOPR 명령을 실행하고 QSYSOPR 메세지 기록부에서 Qccxyyyy에 감지된 소프트웨어 문제점 메세지를 찾으십시오(프로그램명의 cc는 보통 RW이지만 CN 또는 SQ가 될 수도 있습니다). 이 메세지의 존재는 FFDC 자료가 생성되었다는 사실을 암시합니다. 메세지에 관계된 세부사항을 얻기 위해서 도움말 키를 사용할 수 있습니다. 메세지 도움말은 WRKPRB(문제점에 대한 작업) 명령으로 표시되는 리스트에서 문제점을 식별하는 데 사용할 수 있는 문제점 ID를 제공합니다. 문제점 레코드가 존재할 경우 리스트의 맨 위 또는 그 근처에 있을 수 있으므로 이 단계를 건너 뛸 수 있습니다.
2. WRKPRB(문제점에 대한 작업) 명령을 입력하고 Qccxyyyy에서 감지된 소프트웨어 문제점 메세지로부터 프로그램명(Qccxyyyy)을 지정하십시오. 프로그램명을 사용하여 원치 않는 리스트 항목을 필터링하십시오. 문제점 리스트가 제공된 때, 중후 스트링 및 오류 기록부 ID와 같은 문제점 세부사항을 더 많이 얻기 위해 문제점 ID가 수록된 행에서 옵션 5를 지정하십시오.
3. 오류 기록부 ID가 있으면 STRSST(시스템 서비스 툴 시작) 명령을 입력하십시오. 첫 번째 화면에서 서비스 툴 시작을 선택하십시오. 두 번째 화면에서 1을 입력하여 오류 기록부 유틸리티를 선택하십시오. 그 다음 화면에서 2를 입력하여 오류 기록부 ID별로 표시 또는 인쇄를 선택하십시오. 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 80자 워크스테이션 프린터에서 화면 인쇄 및 온라인 보기에 사용할 수 있습니다. 표시 옵션을 선택했을 경우, Enter 키를 입력한 이후 16진 자료를 보기 원한다면 F6을 사용하십시오.

16진 자료는 FFDC 덤프 자료의 처음 1KB를 포함하며 이 자료 앞에는 일부 다른 자료가 있습니다. 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 기록부에 저장된 덤프 중 처음 1KB를 사용하십시오.
- FFDC 자료의 1KB가 충분하지 않을 경우 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 기록부에서 FFDC 덤프의 각 섹션은 4바이트 헤더가 접두부로 표시됩니다. 헤더의 처음 2바이트는 다음 섹션의 길이입니다(접두부를 뺀). 다음 2바이트는 섹션 번호이며 색인의 섹션 번호에 해당합니다.

관련 참조

271 페이지의 『FFDC 덤프 출력 설명』

이 정보는 FFDC(First Failure Data Capture) 덤프 출력에서 사용할 수 있는 자료 영역 및 정보 유형에 대해 설명합니다.

관련 정보

DSPMSG(메세지 표시) 명령

WRKPRB(문제점에 대한 작업) 명령

STRSST(시스템 서비스 툴 시작) 명령

어플리케이션 서버 문제점을 진단하기 위한 서비스 작업 시작

어플리케이션이 분산 관계형 데이터베이스 구조(DRDA)를 사용할 때 SQL문은 어플리케이션 서버 작업 내에서 실행됩니다. 따라서 i5/OS 오퍼레이팅 시스템에서 실행 중인 어플리케이션 서버 작업에 대해 작업 추적 또는 디버그를 시작할 필요가 있을 수 있습니다. 이를 수행하기 위한 기술은 APPC를 사용하는지 또는 TCP/IP를 사용하 지에 따라 다릅니다.

관련 개념

200 페이지의 『작업 추적』

때때로 어떤 문제점은 특정 프로그램에 대해 추적이 불가능합니다. 이러한 경우 STRTRC(추적 시작) 및 TRCJOB(작업 추적) 명령을 사용하여 실행된 CL 명령과 i5/OS 자료 획득 및 모듈 흐름을 추적할 수 있습니다.

APPC 서버에 대한 서비스 작업

iSeries용 DB2 Universal Database 어플리케이션 서버가 특별한 트랜잭션 프로그램명(TPN)을 인식할 때, 어플리케이션 서버는 시스템 오퍼레이터에게 메시지를 송신한 후 응답을 대기합니다.

이는 어플리케이션 서버 작업에 대해 작업 추적 또는 디버그를 시작할 수 있도록 하는 STRSRVJOB(서비스 작업 시작) 명령을 발행할 수 있도록 합니다.

iSeries용 DB2 UDB 어플리케이션 서버 작업을 중지하고 이를 디버그 모드로 재시작하려면 다음 단계를 수행하십시오.

1. 어플리케이션 리퀘스터의 TPN으로 QCNTSRVC를 지정하십시오. 이 지정 방법은 각 플랫폼에 따라 다릅니다. 이 주제는 서로 다른 방법에 대해 설명합니다. i5/OS 어플리케이션이 QCNTSRVC의 TPN을 수신하면 어플리케이션은 CPF9188 메시지를 QSYSOPR로 송신하고 G(이동(go)에 대한) 응답을 대기합니다.

2. G 응답을 입력하기 전에 STRSRVJOB(서비스 작업 시작) 명령을 사용하여 어플리케이션 서버 작업에 대한 서비스 작업을 시작하고 이를 디버그 모드에 넣습니다. (작업명을 표시하는 CPF9188 메시지에 대한 도움말을 요구하십시오.)
3. STRDBG(디버그 시작) 명령을 입력하십시오.
4. 어플리케이션 서버 작업에 대한 디버그를 시작한 후 G로 QSYSOPR 메시지에 응답하십시오. G 응답을 수신한 후 어플리케이션 서버는 정상 DRDA 처리를 계속 수행합니다.
5. 어플리케이션이 실행된 후 SQL 디버그 메시지를 보려면 어플리케이션 서버 작업 기록부를 보십시오.

관련 개념

『사용자 자신의 TPN 작성 및 QCNTSRVC 설정』

사용자 자신의 TPN을 작성하고 QCNTSRVC를 설정하려면 다음 작업을 수행하십시오.

관련 정보

서비스 작업 시작(STRSRVJOB) 명령

디버그 시작(STRDBG) 명령

사용자 자신의 TPN 작성 및 QCNTSRVC 설정

사용자 자신의 TPN을 작성하고 QCNTSRVC를 설정하려면 다음 작업을 수행하십시오.

관련 태스크

209 페이지의 『APPC 서버에 대한 서비스 작업』

iSeries용 DB2 Universal Database 어플리케이션 서버가 특별한 트랜잭션 프로그램명(TPN)을 인식할 때, 어플리케이션 서버는 시스템 오퍼레이터에게 메시지를 송신한 후 응답을 대기합니다.

iSeries용 DB2 Universal Database 어플리케이션 리퀘스터의 TPN으로 QCNTSRVC 설정:

ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 또는 CHGRDBDIRE(관계형 데이터베이스 디렉토리 항목 변경) 명령의 TNSPGM 매개변수에 QCNTSRVC를 지정하십시오.

디버깅 종료 시 다시 되돌릴 것을 기억할 수 있도록 RDB 디렉토리 항목의 텍스트에 특별한 트랜잭션명(TPN)을 표시해 두는 것을 좋습니다.

관련 정보

관계형 데이터베이스(RDB) 디렉토리 항목 추가(ADDRDBDIRE) 명령

관계형 데이터베이스(RDB) 디렉토리 항목 변경(CHGRDBDIRE) 명령

iSeries용 DB2 Universal Database 어플리케이션 서버 작업을 디버깅하기 위한 사용자 자신의 TPN 작성:

맨 끝에 디버그 명령문과 TFRCTL QSYS/QCNTEDDM문을 포함하는 CL 프로그램을 컴파일하여 사용자 자신의 TPN을 작성할 수 있습니다. 이 방법의 장점은 연결을 작성할 때 수동으로 간섭할 필요가 없다는 점입니다.

다음은 프로그램 코드의 예입니다.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```
PGM
  MONMSG CPF0000
    STRDBG UPDPROD(*YES) PGM(CALL/QRWTEXEC) MAXTRC(9999)
    ADDBKP 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 항목에 있는 TPN명은 사용자가 제공하는 이름입니다. 텍스트 필드를 사용하여 특별한 TPN을 사용 중이라는 경고를 입력하고 디버깅을 완료하면 반드시 TPN명을 다시 변경하십시오.

RDB의 TPN을 변경할 때, 다시 변경값을 되돌릴 때까지는 어플리케이션 리퀘스터로부터의 모든 연결이 새로운 TPN을 사용하여 발생한다는 사실을 명심해야 합니다. 이는 성능 저하, 장시간의 오퍼레이터 응답 대기 및 기억장치가 디버그 자료로 가득차는 예상치 못한 상태를 일으킬 수 있습니다.

DB2 Universal Database for VM 어플리케이션 리퀘스터의 TPN으로 QCNTRVC 설정:

TPN 태그 내에 QCNTRVC를 지정하기 위해 UCOMDIR NAMES 파일을 변경하십시오.

예를 들면, 다음과 같습니다.

```
:nick.RCHASLAI :tpn.QCNTRVC
                                     :!uname.VM4GATE RCHASLAI
                                     :modename.MODE645
                                     :security.NONE
```

그런 다음 SET COMDIR RELOAD USER을 발행하십시오.

DB2 Universal Database for z/OS 어플리케이션 리퀘스터의 TPN으로 QCNTRVC 설정:

SYSIBM LOCATIONS 표를 갱신하여 iSeries용 DB2 Universal Database 어플리케이션 서버의 RDB-NAME을 수록하고 있는 행에 대한 TPN 열에 QCNTRVC를 지정하십시오.

Linux, UNIX 및 Windows용 DB2 Universal Database 어플리케이션 리퀘스터의 TPN으로 QCNTRVC 설정:

Linux, UNIX 및 Windows용 DB2 Universal Database에 대해 작업 중이며 이 제품군에서 TPN을 설정하는 방법에 대해 알고 싶으면 해당 웹 페이지를 참조하십시오.

Linux, UNIX 및 Windows용 DB2 Universal Database 웹 페이지를 참조하십시오. 서로 다른 버전별로 여러 가지 책을 찾을 수 있습니다.

TCP/IP 서버에 대한 서비스 작업

DDM TCP/IP 서버는 APPC(Advanced Program-to-Program Communication) 서버가 사용하는 것처럼 트랜잭션 프로그래밍(TPN)을 사용하지 않습니다. 하지만, TCP/IP 서버에 의한 사전시작 작업을 사용하여 환경 내에서 서비스 작업을 시작할 수 있습니다.

그러나 QRWOPTIONS 자료 여역 사용법과 연관된 기능의 도입으로 여러 경우에 서비스 작업을 시작할 필요가 없다는 점을 주지하십시오. 그 특성은 추적을 시작하고 다른 진단 기능을 수행할 수 있도록 해 줍니다. 작업의 연결 단계를 추적해야 하는 경우는 여전히 서비스 작업을 시작할 필요가 있을 수 있습니다.

다음에 해당할 경우 지정된 연결에 사용되는 서버 작업의 이름을 보고하는 CPI3E34 메시지를 찾으려면 DSPLOG(기록부 표시) 명령을 사용할 수 있습니다.

- 연결 조작을 하는 동안 서버의 조치를 추적할 필요가 없습니다.
- QRWOPTIONS 함수를 사용하지 않기로 합니다.
- 대화식 SQL에서처럼 서버에서 일부 설정을 수행할 수 있을 때까지 어플리케이션 리퀘스터(AR)의 실행을 지연할 수 있는 능력이 있습니다.

그런 다음 STRSRVJOB(서비스 작업 시작) 명령을 사용할 수 있습니다.

연결 명령문을 추적해야 하거나 연결 후 서버에서 수동 설정을 수행할 시간이 없는 경우, 연결되기 전에 연결에 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.

QSYSWRK에서 실행하는 QRWTSRVR에 대해 CHGPJE(사전시작 작업 항목 변경) 명령에서 다음 매개변수를 다음과 같이 지정된 값으로 설정하여 사전시작 작업 수를 강제로 1이 되게 할 수 있습니다.

- 초기 작업 수: **1**
- 임계값(threshold): **1**
- 추가 작업 수: **0**
- 최대 작업 수: **1**

이러한 방법을 사용할 경우 해당 매개변수를 다시 사용자 환경에 적당한 값으로 변경해야 합니다. 그렇지 않으면 하나의 사전시작 작업이 사용되고 있을 때 연결을 시도하면 '리모트 소켓을 사용하는 연결이 이 소켓에 의해 재설정되었습니다'라는 메시지가 표시됩니다.

관련 개념

213 페이지의 『QRWOPTIONS 자료 영역』

DDM 또는 DRDA TCP/IP 서버 작업이 시작될 때 이들 작업은 사용자가 진단 및 기타 옵션을 지정할 수 있는 자료 영역을 찾습니다. 해당 이름은 QRWOPTIONS이며, 유효하려면 QGPL 라이브러리에 상주해야 합니다. 48자 스트링으로 이루어집니다.

관련 정보

서비스 작업 시작(STRSRVJOB) 명령

CHGPJE(사전시작 작업 항목 변경) 명령

DSPLOG(기록부 표시) 명령

QRWOPTIONS 자료 영역

DDM 또는 DRDA TCP/IP 서버 작업이 시작될 때 이들 작업은 사용자가 진단 및 기타 옵션을 지정할 수 있는 자료 영역을 찾습니다. 해당 이름은 QRWOPTIONS이며, 유효하려면 QGPL 라이브러리에 상주해야 합니다. 48자 스트링으로 이루어집니다.

주: 자료 영역의 정보는 CCSID 37 또는 500으로 대문자로 입력되어야 합니다.

자료 영역의 형식은 다음과 같습니다.

표 13. 자료 영역 형식

열	목적
1-15	전환 값으로 I를 지정한 경우(그렇지 않으면 무시함) 사용할 점분리 십진 형식의 클라이언트 IP 주소입니다.
16	서버가 무시하는 예약된 영역(사용자를 위한 용도의 문자를 포함할 수 있음)
17-26	전환 값으로 U를 지정한 경우(그렇지 않으면 무시함) 비교하기 위한 사용자 프로파일명입니다.
27	A, I 또는 U로 설정된 경우 작업 기록부가 보관되도록 전환합니다(주 1 및 2 참조).
28	A, I 또는 U로 설정된 경우 DSPJOB 출력이 인쇄되도록 전환합니다(주 1 및 2 참조).
29	A, I 또는 U로 설정된 경우 작업이 추적되도록 전환합니다(주 1 및 2 참조).
30	A, I 또는 U로 설정된 경우 작업에 대해 디버그가 시작되도록 전환합니다(주 1 참조).
31	A, I 또는 U로 설정된 경우 QRYOPTLIB 값을 사용하여 CHGQRYA(조회 속성 변경) 명령을 호출하도록 전환합니다. QRYOPTLIB 값은 39-48열에서 추출되며 이는 올바른 QAQQINI 파일이 들어 있는 라이브러리명을 포함해야 합니다(주 1 참조).
	주: 이 열에 I 또는 A가 지정된 경우 QUSER는 *JOBCTL 특수 권한이 있어야 유효합니다.
32	A, I 또는 U로 설정된 경우 클라이언트 디버그 옵션을 새도우하도록 전환합니다(주 1 참조).
33	T로 설정된 경우 작업 추적에 새 STRTRC 대신 이전 TRCJOB을 사용하도록 전환하고 29열이 추적을 요구합니다.
	주: 이 열이 T로 설정될 경우 TRCJOB을 작업 추적에 사용합니다. STRTRC를 사용하려면 공백 또는 S로 설정하십시오.
34	이력 기록부에서 CPI3E34 메시지를 억제하려면 N으로 설정하십시오(OS/400 V5R1에서는 PTF SI02613이 있는 경우에만 사용할 수 있습니다).
35	A, I 또는 U로 설정된 경우 특별한 서브루틴 추적을 시작하도록 전환합니다(주 1 및 2 참조).
36-38	예약
39-48	일반 자료 영역(31열의 적절한 값에 의해 CHGQRYA(조회 속성 변경) 명령이 트리거될 경우 라이브러리 명을 포함합니다).

주:

- 이름이 이 값이 나타나는 열에 해당되는 기능을 활성화하는 전환 값입니다.
 - A는 서버 작업을 모두 사용하도록 해당 기능을 활성화합니다.
 - I는 1-15열에 지정된 클라이언트 IP 주소가 연결 시도에 사용된 주소와 일치할 경우 해당 기능을 활성화합니다.

- U는 17-26열에 지정된 사용자 ID가 연결 시도에 사용된 ID와 일치할 경우 해당 기능을 활성화합니다.
2. 이 기능의 결과로 생긴 스푼 파일을 찾으려면 WRKJOB 사용자 프로파일/QPRTJOB(작업에 대한 작업) 명령을 사용하십시오. 여기서 사용자 프로파일은 연결 요구에 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	

3. 특수 DRDA 서버루틴 추적이 들어 있는 파일이 QGPL 라이브러리에 QRWDBmmdy 형식의 이름으로 작성됩니다. 여기서 mm은 추적이 기록된 월, dd는 일, y는 연도의 마지막 숫자입니다. 모든 서버 프로그램이 추적되는 것은 아닙니다.

관련 태스크

144 페이지의 『이력 기록부 표시』

클라이언트 사용자가 서버 작업과의 연결을 성공적으로 설치할 때마다 해당 작업은 해당 클라이언트 사용자의 프로파일하에 실행하도록 교체됩니다.

관련 참조

182 페이지의 『분산 관계형 데이터베이스 메시지』

Application Server(AS) 또는 어플리케이션 리퀘스터(AS)에 오류 메시지가 발생할 경우 서버 메시지는 작업 기록부에 로그되어 실패의 이유를 표시합니다.

199 페이지의 『TCP/IP 서버 사전시작 작업에서 작업 기록부 찾기』

분산 자료 관리(DDM) TCP/IP 서버와 연관된 QRWTSRVR 사전시작 작업 중 하나가 서비스하는 연결이 종료될 때 사전시작 작업은 다른 연결에서 사용되도록 재순환됩니다. 이 상황이 발생하면, 종료된 연결과 관련된 작업 기록부는 지워집니다.

212 페이지의 『TCP/IP 서버에 대한 서비스 작업』

DDM TCP/IP 서버는 APPC(Advanced Program-to-Program Communication) 서버가 사용하는 것처럼 트랜잭션 프로그래밍(TPN)을 사용하지 않습니다. 하지만, TCP/IP 서버에 의한 사전시작 작업을 사용하여 환경 내에서 서비스 작업을 시작할 수 있습니다.

관련 정보

CHGQRYA(조회 속성 변경) 명령

WRKJOB(작업(Job)에 대한 작업) 명령

예: 자료 영역을 작성하는 CL 명령:

이 예에는 표에 표시된 기능이 필요합니다.


```

CRTDTAARA DTAARA(QGPL/QRWOPTIONS) TYPE(*CHAR) LEN(48)
VALUE('9.5.114.107 :MYUSERID AAUIU TN INILIBRARY')
TEXT('DRDA TCP SERVER DIAGNOSTIC OPTIONS'

```

주: 예에 나타난 적절한 띄어쓰기가 정확해야 하므로 VALUE 매개변수의 내용은 표 형식으로 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	29 및 31열의 스위치와 함께 사용하기 위한 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 요구에서 사용자 ID가 자료 영역의 17-26열에 지정된 것(이 예에서 'MYUSERID')과 일치하면 TRCJOB 명령으로 작업을 추적합니다.
30	I	클라이언트 IP 주소(이 예에서 '9.5.114.107')가 자료 영역의 1-5열에 지정된 것과 일치할 경우 STRDBG(디버그 시작) 명령(프로그램을 지정하지 않음)으로 디버그를 시작합니다.
31	U	연결 요구의 사용자 ID가 자료 영역의 17-26열에 지정된 것(이 예에서 'MYUSERID')과 일치하면 CHGQRYA(조회 속성 변경) QRYOPLIB(INILIBRARY) 명령을 호출합니다. 주: 자료 영역의 39-48열에서 자료명을 가져올 수 있습니다.
32		클라이언트 디버그 옵션을 서버로 새도우하지 않습니다.
33	T	작업 추적을 위해 오래된 TRCJOB 기능을 사용합니다.
34	N	이력 기록부에 CPI3E34 메시지를 기록하지 않습니다.
35		서브루틴 추적을 시작하지 않음
39-48	INILIBRARY	전환 31로 사용되는 라이브러리

관련 정보

CHGQRYA(조회 속성 변경) 명령

디버그 시작(STRDBG) 명령

예: 어플리케이션 프로그래밍

분산 관계형 데이터베이스 사용을 위한 이 어플리케이션 예는 RPG/400, COBOL/400, Java 및 ILE C/400 프로그래밍 언어로 작성되었습니다. 이 예는 기능적 스펙 타스크에 대한 분산 관계형 데이터베이스의 사용 방법을 보여줍니다.

예: 분산 관계형 데이터베이스에 대한 비즈니스 요구사항

이 예에 있는 분산 관계형 데이터베이스의 어플리케이션은 자동차 판매상 또는 판매 대리점 네트워크의 부품 재고 관리 어플리케이션입니다.

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 단위(LUW)의 개념을 사용합니다. 이 프로그램은 리모트 작업 단위를 사용하므로 다른 서버에서 새 작업 단위를 시작하기 전에 먼저 한 서버(COMMIT)의 현재 LUW를 닫아야 합니다.

커서 위치 재지정

LUW가 확약되거나 어플리케이션이 다른 데이터베이스에 연결되면 모든 커서가 닫혀집니다. 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 동일한 부품 번호에 중복 행이 있으면 작동하지 않습니다.

| 예; 프로그램 정의

| 주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

|

```

| /*****
| /* Local Database */
| /*****
| CREATE COLLECTION DRDA
|
| CREATE TABLE DRDA/PART_STOCK (PART_NUM CHAR(5) NOT NULL,
| PART_UM CHAR(2) NOT NULL,
| PART_QUANT INTEGER NOT NULL WITH DEFAULT,
| PART_ROP INTEGER NOT NULL,
| PART_EOQ INTEGER NOT NULL,
| PART_BIN CHAR(6) NOT NULL WITH DEFAULT
| )
|
| CREATE UNIQUE INDEX DRDA/PART_STOCI ON DRDA/PART_STOCK (PART_NUM ASC)
|
| /*****
| /* Remote Database */
| /*****
| CREATE COLLECTION DRDA
|
| CREATE TABLE DRDA/PART_ORDER (ORDER_NUM SMALLINT NOT NULL,
| ORIGIN_LOC CHAR(4) NOT NULL,
| ORDER_TYPE CHAR(1) NOT NULL,
| ORDER_STAT CHAR(1) NOT NULL,
| NUM_ALLOC SMALLINT NOT NULL WITH DEFAULT,
| URG_REASON CHAR(1) NOT NULL WITH DEFAULT,
| CREAT_TIME TIMESTAMP NOT NULL,
| ALLOC_TIME TIMESTAMP,
| CLOSE_TIME TIMESTAMP,
| REV_REASON CHAR(1)
| )
|
| CREATE UNIQUE INDEX DRDA/PART_ORDEI ON DRDA/PART_ORDER (ORDER_NUM ASC)
|
| CREATE TABLE DRDA/PART_ORDLN (ORDER_NUM SMALLINT NOT NULL,
| ORDER_LINE SMALLINT NOT NULL,
| PART_NUM CHAR(5) NOT NULL,
| QUANT_REQ INTEGER NOT NULL,
| LINE_STAT CHAR(1) NOT NULL
| )
|
| CREATE UNIQUE INDEX PART_ORDLI ON DRDA/PART_ORDLN (ORDER_NUM ASC,
| ORDER_LINE ASC)
|
| CREATE TABLE DRDA/SHIPMENTLN (SHIP_NUM SMALLINT NOT NULL,
| SHIP_LINE SMALLINT NOT NULL,
| ORDER_LOC CHAR(4) NOT NULL,
| ORDER_NUM SMALLINT NOT NULL,
| ORDER_LINE SMALLINT NOT NULL,
| PART_NUM CHAR(5) NOT NULL,
| QUANT_SHIP INTEGER NOT NULL,
| QUANT_RECV INTEGER NOT NULL WITH DEFAULT
| )
|
| CREATE UNIQUE INDEX SHIPMENTLI ON DRDA/SHIPMENTLN (SHIP_NUM ASC,
| SHIP_LINE ASC)

```

그림 22. 컬렉션 및 표 작성

```

| /*****
| /* Local Database */
| /*****
| /*****
| /* PART_STOCK */
| /*****
| INSERT INTO DRDA/PART_STOCK VALUES('14020','EA',038,050,100,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('14030','EA',043,050,050,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('14040','EA',030,020,030,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('14050','EA',010,005,015,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('14060','EA',110,045,090,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('14070','EA',130,080,160,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('18020','EA',013,025,050,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('18030','EA',015,005,010,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('21010','EA',029,030,050,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('24010','EA',025,020,040,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('24080','EA',054,050,050,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('24090','EA',030,025,050,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('24100','EA',020,015,030,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('24110','EA',052,050,080,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('25010','EA',511,300,600,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('36010','EA',013,005,010,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('36020','EA',110,030,060,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('37010','EA',415,100,200,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('37020','EA',010,020,040,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('37030','EA',154,055,060,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('37040','EA',223,120,120,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('43010','EA',110,020,040,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('43020','EA',067,050,050,' ')
|
| INSERT INTO DRDA/PART_STOCK VALUES('48010','EA',032,030,060,' ')
|
|
| /*****

```

그림 23. 표에 자료 삽입

```

| /* Remote Database */
| /*****
| /*****
| /* PART_ORDER TABLE */
| /*****
| INSERT INTO DRDA/PART_ORDER VALUES(1,'DB2B','U','0',0,' ','1991-03-12-17.00.00',
| NULL,NULL,NULL)
|
| INSERT INTO DRDA/PART_ORDER VALUES(2,'SQLA','U','0',0,' ','1991-03-12-17.01.00',
| NULL,NULL,NULL)
|
| INSERT INTO DRDA/PART_ORDER VALUES(3,'SQLA','U','0',0,' ','1991-03-12-17.02.00',
| NULL,NULL,NULL)
|
| INSERT INTO DRDA/PART_ORDER VALUES(4,'SQLA','U','0',0,' ','1991-03-12-17.03.00',
| NULL,NULL,NULL)
|
| INSERT INTO DRDA/PART_ORDER VALUES(5,'DB2B','U','0',0,' ','1991-03-12-17.04.00',
| NULL,NULL,NULL)
|
| /*****
| /* PART_ORDLN TABLE */
| /*****
| INSERT INTO DRDA/PART_ORDLN VALUES(1,1,'24110',005,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(1,2,'24100',021,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(1,3,'24090',018,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(2,1,'14070',004,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(2,2,'37040',043,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(2,3,'14030',015,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(3,2,'14030',025,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(3,1,'43010',003,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(4,1,'36010',013,'0')
|
| INSERT INTO DRDA/PART_ORDLN VALUES(5,1,'18030',005,'0')
|
| /*****
| /* SHIPMENTLN TABLE */
| /*****
| INSERT INTO DRDA/SHIPMENTLN VALUES(1,1,'DB2B',1,1,'24110',5,5)
|
| INSERT INTO DRDA/SHIPMENTLN VALUES(1,2,'DB2B',1,2,'24100',10,1)
|
| INSERT INTO DRDA/SHIPMENTLN VALUES(2,1,'SQLA',2,1,'14070',4,4)
|
| INSERT INTO DRDA/SHIPMENTLN VALUES(2,2,'SQLA',2,2,'37040',45,25)
|
| INSERT INTO DRDA/SHIPMENTLN VALUES(2,3,'SQLA',2,3,'14030',5,5)
|
| INSERT INTO DRDA/SHIPMENTLN VALUES(3,1,'SQLA',2,3,'14030',5,5)

```

그림 24. 표에 자료 삽입(계속)

예: RPG 프로그램

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```

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 25. RPG 프로그램 예

```

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          03/29/92
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          03/29/92
7700    C+    DECLARE NEXT PART CURSOR FOR          03/29/92
7800    C+    SELECT PART_NUM,          03/29/92
7900    C+    PART_QUANT,          03/29/92
8000    C+    PART_ROP,          03/29/92
8100    C+    PART_EOQ          03/29/92
8200    C+    FROM PART_STOCK          03/29/92
8300    C+    WHERE PART_ROP > PART_QUANT          03/29/92
8400    C+    AND PART_NUM > :PRTTBL          03/29/92
8500    C+    ORDER BY PART_NUM ASC          03/29/92
8600    C/END-EXEC          03/29/92
8700    C*
8800    C* ORDERS WHICH ARE ALREADY MADE FOR CURRENT PART          03/29/92
8900    C/EXEC SQL          03/29/92
9000    C+    DECLARE NEXT_ORDER_LINE CURSOR FOR          03/29/92
9100    C+    SELECT A.ORDER_NUM,          03/29/92
9200    C+    ORDER_LINE,          03/29/92
9300    C+    QUANT_REQ          03/29/92
9400    C+    FROM PART_ORDLN A,          03/29/92
9500    C+    PART_ORDER B          03/29/92
9600    C+    WHERE PART_NUM = :PRTTBL          03/29/92
9700    C+    AND LINE_STAT <> 'C'          03/29/92
9800    C+    AND A.ORDER_NUM = B.ORDER_NUM          03/29/92
9900    C+    AND ORDER_TYPE = 'R'          03/29/92
10000   C/END-EXEC          03/29/92
10100   C*
10200   C*****
10300   C*    SQL RETURN CODE HANDLING          *          03/29/92
10400   C*****
10500   C/EXEC SQL          03/29/92
10600   C+    WHENEVER SQLERROR GO TO DBERRO          03/29/92
10700   C/END-EXEC          03/29/92
10800   C/EXEC SQL          03/29/92
10900   C+    WHENEVER SQLWARNING CONTINUE          03/29/92
11000   C/END-EXEC          03/29/92
11100   C*
11200   C*

```



```

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
11300 C*****
11400 C*   PROCESS - MAIN PROGRAM LOGIC          *          03/29/92
11500 C*   MAIN PROCEDURE WORKS WITH LOCAL DATABASE          *          03/29/92
11600 C*****
11700 C*
11800 C*CLEAN UP TO PERMIT RE-RUNNING OF TEST DATA          03/29/92
11900 C           EXSR CLEANU          03/29/92
12000 C*
12100 C*
12200 C           RTCOD1   DOUEQ100          03/29/92
12300 C*
12400 C/EXEC SQL
12500 C+   CONNECT TO :LOCADB          03/29/92
12600 C/END-EXEC
12700 C/EXEC SQL
12800 C+   OPEN   NEXT_PART          03/29/92
12900 C/END-EXEC
13000 C/EXEC SQL
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
13700 C           MOVE SQLCOD   RTCOD1          03/29/92
13800 C/EXEC SQL
13900 C+   COMMIT          03/29/92
14000 C/END-EXEC
14100 C           RTCOD1   IFNE 100          03/29/92
14200 C           EXSR CHECKO          03/29/92
14300 C           ENDIF          03/29/92
14400 C*
14500 C           ENDDO          03/29/92
14600 C*
14700 C           GOTO SETLR          03/29/92
14800 C*
14900 C*
15000 C*****
15100 C*   SQL RETURN CODE HANDLING ON ERROR SITUATIONS          *          03/29/92
15200 C*****
15300 C*
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
15900 C+   WHENEVER SQLERROR CONTINUE          03/29/92
16000 C/END-EXEC
16100 C/EXEC SQL
16200 C+   ROLLBACK          03/29/92
16300 C/END-EXEC
16400 C/EXEC SQL          03/29/92

```

```

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
16500 C+          WHENEVER SQLERROR GO TO DBERRO          03/29/92
16600 C/END-EXEC
16700 C*
16800 C*
16900 C          SETLR          TAG          03/29/92
17000 C*          *-----*          03/29/92
17100 C/EXEC SQL
17200 C+          CONNECT          RESET          03/29/92
17300 C/END-EXEC
17400 C          MOVE *ON          *INLR          03/29/92
17500 C*
17600 C*****
17700 C*          THE END OF THE PROGRAM          *          03/29/92
17800 C*****
17900 C*
18000 C*
18100 C*****
18200 C* SUBROUTINES TO WORK WITH REMOTE DATABASES          *          03/29/92
18300 C*****
18400 C*
18500 C*
18600 C          CHECKO          BEGSR          03/29/92
18700 C*          *-----*          03/29/92
18800 C*****
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*****
19300 C*
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*
19800 C/EXEC SQL
19900 C+          CONNECT          TO          :REMO DB          03/29/92
20000 C/END-EXEC
20100 C/EXEC SQL
20200 C+          OPEN          NEXT_ORDER_LINE          03/29/92
20300 C/END-EXEC
20400 C*
20500 C          RTCOD2          DOWNE100          03/29/92
20600 C*
20700 C/EXEC SQL
20800 C+          FETCH          NEXT_ORDER_LINE          03/29/92
20900 C+          INTO          :CURORD,          03/29/92
9 ...+... 0
21000 C+          :CURORL,          03/29/92
21100 C+          :QUANTI          03/29/92
21200 C/END-EXEC
21300 C*

```

```

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 ...+...
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                        03/29/92
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
25500 C*****                                                    03/29/92
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*****                                                    03/29/92

```

```

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
26100 C*
26200 C          *IN89      IFEQ *OFF          FIRST ORDER ?
26300 C          EXSR STRORD
26400 C          MOVE *ON          *IN89          ORD.HEAD.DONE
26500 C          EXCPTHEADER      WRITE HEADERS
26600 C          ENDIF
26700 C*
26800 C          ADD 1          NXTORL          NEXT ORD.LIN
26900 C/EXEC SQL
27000 C+          INSERT
27100 C+          INTO      PART_ORDLN
27200 C+          (ORDER_NUM,
27300 C+          ORDER_LINE,
27400 C+          PART_NUM,
27500 C+          QUANT_REQ,
27600 C+          LINE_STAT)
27700 C+          VALUES  (:NXTORD,
27800 C+          :NXTORL,
27900 C+          :PRTTBL,
28000 C+          :QTYORD,
28100 C+          '0')
28200 C/END-EXEC
28300 C*
28400 C          *INOF      IFEQ *ON
28500 C          EXCPTHEADER
28600 C          END
28700 C          EXCPTDETAIL
28800 C*
28900 C          ENDSR          ORDERP
29000 C*
29100 C*
29200 C          STRORD      BEGSR
29300 C*          *-----*
29400 C*****
29500 C* SEARCHES FOR NEXT ORDER NUMBER AND MAKES AN ORDER HEADER *
29600 C* USING THAT NUMBER. WRITES ALSO HEADERS ON REPORT. *
29700 C*****
29800 C*
29900 C/EXEC SQL
30000 C+          SELECT      (MAX(ORDER_NUM) + 1)
30100 C+          INTO        :NXTORD
30200 C+          FROM        PART_ORDER
30300 C/END-EXEC
30400 C/EXEC SQL
30500 C+          INSERT
30600 C+          INTO      PART_ORDER
30700 C+          (ORDER_NUM,
30800 C+          ORIGIN_LOC,

```

```

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
30900 C+ ORDER_TYPE, 03/29/92
31000 C+ ORDER_STAT, 03/29/92
31100 C+ CREAT_TIME) 03/29/92
31200 C+ VALUES (:NXTORD, 03/29/92
31300 C+ :LOC, 03/29/92
31400 C+ 'R', 03/29/92
31500 C+ 'O', 03/29/92
31600 C+ CURRENT_TIMESTAMP) 03/29/92
31700 C/END-EXEC 03/29/92
31800 C ENDSR STRORD 03/29/92
31900 C* 03/29/92
32000 C* 03/29/92
32100 C CLEANU BEGSR 03/29/92
32200 C* *-----* 03/29/92
32300 C***** 03/29/92
32400 C* THIS SUBROUTINE IS ONLY REQUIRED IN A TEST ENVIRONMENT 03/29/92
32500 C* TO RESET THE DATA TO PERMIT RE-RUNNING OF THE TEST 03/29/92
32600 C***** 03/29/92
32700 C* 03/29/92
32800 C/EXEC SQL 03/29/92
32900 C+ CONNECT TO :REMO DB 03/29/92
33000 C/END-EXEC 03/29/92
33100 C/EXEC SQL 03/29/92
33200 C+ DELETE 03/29/92
33300 C+ FROM PART_ORDLN 03/29/92
33400 C+ WHERE ORDER_NUM IN 03/29/92
33500 C+ (SELECT ORDER_NUM 03/29/92
33600 C+ FROM PART_ORDER 03/29/92
33700 C+ WHERE ORDER_TYPE = 'R') 03/29/92
33800 C/END-EXEC 03/29/92
33900 C/EXEC SQL 03/29/92
34000 C+ DELETE 03/29/92
34100 C+ FROM PART_ORDER 03/29/92
34200 C+ WHERE ORDER_TYPE = 'R' 03/29/92
34300 C/END-EXEC 03/29/92
34400 C/EXEC SQL 03/29/92
34500 C+ COMMIT 03/29/92
34600 C/END-EXEC 03/29/92
34700 C* 03/29/92
34800 C ENDSR CLEANU 03/29/92
34900 C* 03/29/92
35000 C* 03/29/92
35100 C***** 03/29/92
35200 O* OUTPUTLINES FOR THE REPORT * 03/29/92
35300 O***** 03/29/92
35400 O* 03/29/92
35500 OQPRINT E 2 HEADER 03/29/92
35600 O + 0 '***** ROP PROCESSING' 03/29/92

```

```

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
35700      0                      + 1 'REPORT *****'          03/29/92
35800      0*
35900      QQPRINT E 2              HEADER          03/29/92
36000      0                      + 0 ' ORDER NUMBER = '      03/29/92
36100      0                      NXTORDZ + 0          03/29/92
36200      0*
36300      QQPRINT E 1              HEADER          03/29/92
36400      0                      + 0 '-----'          03/29/92
36500      0                      + 0 '-----'          03/29/92
36600      0*
36700      QQPRINT E 1              HEADER          03/29/92
36800      0                      + 0 ' LINE '          03/29/92
36900      0                      + 0 'PART '          03/29/92
37000      0                      + 0 'QTY '          03/29/92
37100      0*
37200      QQPRINT E 1              HEADER          03/29/92
37300      0                      + 0 ' NUMBER '          03/29/92
37400      0                      + 0 'NUMBER '          03/29/92
37500      0                      + 0 'REQUESTED '       03/29/92
37600      0*
37700      QQPRINT E 11             HEADER          03/29/92
37800      0                      + 0 '-----'          03/29/92
37900      0                      + 0 '-----'          03/29/92
38000      0*
38100      QQPRINT EF1             DETAIL          03/29/92
38200      0                      NXTORLZ + 4          03/29/92
38300      0                      PRTTBL + 4          03/29/92
38400      0                      QTYORD1 + 4         03/29/92
38500      0*
38600      QQPRINT T 2             LRN99          03/29/92
38700      0                      + 0 '-----'          03/29/92
38800      0                      + 0 '-----'          03/29/92
38900      QQPRINT T 1             LRN99          03/29/92
39000      0                      + 0 'NUMBER OF LINES ' 03/29/92
39100      0                      + 0 'CREATED = '      03/29/92
39200      0                      NXTORLZ + 0          03/29/92
39300      0*
39400      QQPRINT T 1             LRN99          03/29/92
39500      0                      + 0 '-----'          03/29/92
39600      0                      + 0 '-----'          03/29/92
39700      0*
39800      QQPRINT T 2             LRN99          03/29/92
39900      0                      + 0 '*****'          03/29/92
40000      0                      + 0 ' END OF PROGRAM ' 03/29/92
40100      0                      + 0 '*****'          03/29/92
40200      0*
40300      QQPRINT E 2              ERRLIN          03/29/92
40400      0                      + 0 '** ERROR **'     03/29/92
40500      0                      + 0 '** ERROR **'     03/29/92
40600      0                      + 0 '** ERROR **'     03/29/92
40700      QQPRINT E 1              ERRLIN          03/29/92
40800      0                      + 0 '* SQLCOD:'       03/29/92
40900      0                      SQLCODM + 0          03/29/92
41000      0                      33 '* '              03/29/92
41100      QQPRINT E 1              ERRLIN          03/29/92
41200      0                      + 0 '* SQLSTATE:'     03/29/92
41300      0                      SQLSTT + 2          03/29/92
41400      0                      33 '* '              03/29/92
41500      QQPRINT E 1              ERRLIN          03/29/92
41600      0                      + 0 '** ERROR **'     03/29/92
41700      0                      + 0 '** ERROR **'     03/29/92
41800      0                      + 0 '** ERROR **'     03/29/92

```

예: COBOL 프로그램

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```

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.
400  *****
500  *  MODULE NAME = DDBPT6CB
600  *
700  *  DESCRIPTIVE NAME = D-DB SAMPLE APPLICATION
800  *                      REORDER POINT PROCESSING
900  *                      AS/400
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,
1500 *              CREATES A SUPPLY ORDER AND PRINTS A REPORT.
1600 *
1700 *  DEPENDENCIES = NONE
1800 *
1900 *  INPUT = PARAMETERS EXPLICITLY PASSED TO THIS FUNCTION:
2000 *
2100 *          LOCAL-DB      LOCAL DB NAME
2200 *          REMOTE-DB     REMOTE DB NAME
2300 *
2400 *  TABLES = PART-STOCK      - LOCAL
2500 *              PART_ORDER    - REMOTE
2600 *              PART_ORDLN    - REMOTE
2700 *              SHIPMENTLN    - REMOTE
2800 *
2900 *  CRTSQLCBL SPECIAL PARAMETERS
3000 *  PGM(DDBPT6CB) RDB(remotedbname) OPTION(*APOST *APOSTSQL)
3100 *
3200 *  INVOKE BY : CALL DDBPT6CB PARM(localdbname remotedbname)
3300 *
3400 *****
3500 ENVIRONMENT DIVISION.
3600 *-----*
3700 INPUT-OUTPUT SECTION.
3800 FILE-CONTROL.
3900     SELECT RELAT ASSIGN TO PRINTER-QPRINT.
4000 DATA DIVISION.
4100 *-----*
4200 FILE SECTION.
4300 *-----*
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 *-----*
5100 *  PRINT LINE DEFINITIONS
5200     01 LINE0          PIC X(33) VALUE SPACES.
5300     01 LINE1          PIC X(33) VALUE

```

그림 26. COBOL 프로그램 예


```

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'.                                       03/29/92
10500         10 MSG-ID-2     PIC 99999.                             03/29/92
    
```

```

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.

```

```

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              :EQQ-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      *****

```

```

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_RECV)                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-REQ
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      *****                                03/29/92
24100      * RECONNECT TO LOCAL DATABASE *                03/29/92
24200      *****                                03/29/92
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.

```

```

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
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                       '0')
26700          END-EXEC.
26800          PERFORM DETAIL-PROC THRU DETAIL-EXIT.
26900          ORDER-EXIT. EXIT.
27000          *-----
27100
27200          CREATE-ORDER-PROC.
27300          *-----
27400          *GET NEXT ORDER NUMBER
27500          EXEC SQL
27600              SELECT (MAX(ORDER_NUM) + 1)
27700              INTO  :NEXT-NUM:IND-NULL
27800              FROM  PART_ORDER
27900          END-EXEC.
28000          IF IND-NULL < 0
28100              MOVE 1 TO NEXT-NUM.
28200          EXEC SQL
28300              INSERT
28400              INTO    PART_ORDER
28500                  (ORDER_NUM,
28600                   ORIGIN_LOC,
28700                   ORDER_TYPE,
28800                   ORDER_STAT,
28900                   CREAT_TIME)
29000              VALUES (:NEXT-NUM,
29100                       :LOC, 'R', 'O',
29200                       CURRENT_TIMESTAMP)
29300          END-EXEC.
29400          MOVE NEXT-NUM TO MASK0.
29500          PERFORM HEADER-PROC THRU HEADER-EXIT.
29600          CREATE-ORDER-EXIT. EXIT.
29700          *-----
29800
29900          DB-ERROR.
30000          *-----
30100          PERFORM ERROR-MSG-PROC THRU ERROR-MSG-EXIT.
30200          *****
30300          *   ROLLBACK THE LUW *

```

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

03/29/92

```

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.
30600 ****
30700 EXEC SQL ROLLBACK WORK END-EXEC.
30800 ****
30900 PERFORM END-OF-PROGRAM THRU MAIN-PROGRAM-EXIT.
31000 * -- NEXT LINE INCLUDED TO RESET THE "GO TO" DEFAULT --
31100 EXEC SQL WHENEVER SQLERROR GO TO DB-ERROR END-EXEC.
31200
31300 ERROR-MSG-PROC.
31400 *-----
31500 MOVE SQLCODE TO MSG-ID-2.
31600 DISPLAY 'SQL STATE =' SQLSTATE ' SQLCODE =' MSG-ID-2.
31700 * -- ADD HERE ANY ADDITIONAL ERROR MESSAGE HANDLING --
31800 ERROR-MSG-EXIT. EXIT.
31900 *-----
32000
32100 *****
32200 * REPORT PRINTING *
32300 *****
32400 HEADER-PROC.
32500 *-----
32600 WRITE REPRES FROM LINE1 AFTER ADVANCING PAGE.
32700 WRITE REPRES FROM LINE2 AFTER ADVANCING 3 LINES.
32800 WRITE REPRES FROM LINE3 AFTER ADVANCING 2 LINES.
32900 WRITE REPRES FROM LINE4 AFTER ADVANCING 1 LINES.
33000 WRITE REPRES FROM LINE5 AFTER ADVANCING 1 LINES.
33100 WRITE REPRES FROM LINE3 AFTER ADVANCING 1 LINES.
33200 WRITE REPRES 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 REPRES FROM LINE6 AFTER ADVANCING 1 LINES.
34500 DETAIL-EXIT. EXIT.
34600 *-----
34700 TRAILER-PROC.
34800 *-----
34900 MOVE CONTL TO MASK3.
35000 WRITE REPRES FROM LINE3 AFTER ADVANCING 2 LINES.
35100 WRITE REPRES FROM LINE7 AFTER ADVANCING 2 LINES.
35200 WRITE REPRES FROM LINE3 AFTER ADVANCING 2 LINES.
35300 WRITE REPRES FROM LINE8 AFTER ADVANCING 1 LINES.
35400 TRAILER-EXIT. EXIT.
35500 *-----

```

```

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          *-----
***** END OF SOURCE *****

```

| 예: 삽입된 SQL을 사용하는 C 프로그램

| 주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

|

```

| /*****
| /* PROGRAM NAME: SAMPEMBC */
| /* */
| /* DESCRIPTIVE NAME: Sample embedded C application using DRDA */
| /* */
| /* FUNCTION: This module processes the PART_STOCK table and */
| /* for each part below the ROP (REORDER POINT) */
| /* creates a supply order. */
| /* */
| /* LOCAL TABLES: PART_STOCK */
| /* */
| /* REMOTE TABLES: PART_ORDER, PART_ORDLN, SHIPMENTLN */
| /* */
| /* COMPILE OPTIONS: */
| /* CRTSQLCI OBJ(SAMPEMBC) COMMIT(*CHG) RDB(rdbname) OBJTYPE(*PGM) */
| /* RDBCNNMTH(*RUW) */
| /* */
| /* INVOKED BY: */
| /* CALL PGM(SAMPEMBC) PARM('lcldbname' 'rmtdbname') */
| /*****/
| #include <stdlib.h>
| #include <string.h>
| #include <stdio.h>
|
| EXEC SQL INCLUDE SQLCA;
|
| /*****
| /* SQL Variables */
| /*****/
| EXEC SQL BEGIN DECLARE SECTION;
| char loc[4] = "SQLA"; /* dealer's database name */
| char remote_db[18] = " ";
| /* sample remote database */
| char local_db[18] = " ";
| /* sample local database */
| char part_table[5] = " "; /* part number in table part_stock */
| long quant_table; /* quantity in stock, tbl part_stock */
| long rop_table; /* reorder point , tbl part_stock */
| long eoq_table; /* reorder quantity , tbl part_stock */
| short next_num; /* next order nbr,table part_order */
| short ord_table; /* order nbr. , tbl order_line */
| short orl_table; /* order line , tbl order_line */
| long qty_table; /* ordered quantity , tbl order_line */
| long line_count = 0; /* total number of order lines */
| short ind_null; /* null indicator for qty_table */
| short contl = 0; /* continuation line, tbl order_line */
| EXEC SQL END DECLARE SECTION;
|
| /*****
| /* Other Variables */
| /*****/
| char first_time, what_time;
| long qty_rec = 0, qty_req = 0;
|
| EXEC SQL WHENEVER SQLERROR GOTO error_tag;
| EXEC SQL WHENEVER SQLWARNING CONTINUE;

```

| 그림 27. 삽입된 SQL을 사용하는 C 프로그램 예


```

|
|
|
| /*****
| /* Function Declarations */
| /*****
| /* Function For Declaring Cursors */
| /*****
| declare_cursors() {
|
| /* SQL Cursor declaration and reposition for local UW */
| EXEC SQL DECLARE NEXT_PART CURSOR FOR
|         SELECT PART_NUM, PART_QUANT, PART_ROP, PART_EQQ
|         FROM DRDA/PART_STOCK
|         WHERE PART_ROP > PART_QUANT AND
|               PART_NUM > :part_table
|         ORDER BY PART_NUM;
|
| /* SQL Cursor declaration and connect for RUW */
| EXEC SQL DECLARE NEXT_OLINE CURSOR FOR
|         SELECT A.ORDER_NUM, ORDER_LINE, QUANT_REQ
|         FROM DRDA/PART_ORDLN A,
|              DRDA/PART_ORDER B
|         WHERE PART_NUM = :part_table AND
|               LINE_STAT <> 'C' AND
|               A.ORDER_NUM = B.ORDER_NUM AND
|               ORDER_TYPE = 'R';
|
| /* upline exit function in connectable state */
| goto function_exit;
|
| error_tag:
|   error_function();
|
| function_exit:
|   ;
| } /* function declare_cursor */
|
|

```

```

| /*****
| /* Function For Reseting Tables */
| /*****
| reset_tables() {
|
| /* Clean up for rerunability in test environment */
| EXEC SQL CONNECT TO :remote_db;
| EXEC SQL DELETE FROM DRDA/PART_ORDLN
|           WHERE ORDER_NUM IN
|           (SELECT ORDER_NUM
|            FROM DRDA/PART_ORDER
|            WHERE ORDER_TYPE = 'R');
| EXEC SQL DELETE FROM DRDA/PART_ORDER
|           WHERE ORDER_TYPE = 'R';
| /* Exit function in connectable state */
| EXEC SQL COMMIT;
| goto function_exit;
|
| error_tag:
| error_function();
|
| function_exit:
| ;
| } /* function delete_for_rerun */
|
|

```

```

| /*****
| /* Function For Declaring Cursors
| /*****
| calculate_order_quantity() {
|
|     /* Set current connection to local database
|     EXEC SQL CONNECT TO :local_db;
|     /* available qty = Stock qty + qty in order - qty received
|     EXEC SQL OPEN NEXT_PART;
|     EXEC SQL FETCH NEXT_PART
|         INTO :part_table, :quant_table, :rop_table, :eoq_table;
|
|     if (sqlca.sqlcode == 100) {
|         printf("-----\n");
|         printf("NUMBER OF LINES CREATED = %d\n",line_count);
|         printf("-----\n");
|         printf("***** END OF PROGRAM *****\n");
|         rop_table = 0;
|         /* no (more) orders to process
|     }
|     else {
|         qty_rec = 0;
|         qty_req = 0;
|         EXEC SQL COMMIT;
|         EXEC SQL CONNECT TO :remote_db;
|         EXEC SQL OPEN NEXT_OLINE;
|         do {
|             EXEC SQL FETCH NEXT_OLINE
|                 INTO :ord_table, :orl_table, :qty_table;
|             qty_rec = qty_rec + qty_table;
|         } while(sqlca.sqlcode != 100);
|         EXEC SQL CLOSE NEXT_OLINE;
|         EXEC SQL SELECT SUM(QUANT_RECV)
|             INTO :qty_table:ind_null
|             FROM DRDA/SHIPMENTLN
|             WHERE ORDER_LOC = :loc AND
|                   ORDER_NUM = :ord_table AND
|                   ORDER_LINE = :orl_table;
|         if (ind_null != 0)
|             qty_rec = qty_rec + qty_table;
|     } /* end of else branch
|
|     goto function_exit;
|
| error_tag:
|     error_function();
|
| function_exit:
|     ;
| } /* end of calculate_order_quantity

```

```

| /*****
| /* Function For Declaring Cursors */
| /*****
| process_order() {
|
| /* insert order and order_line in remote database */
| if (cont1 == 0) {
|     EXEC SQL SELECT (MAX(ORDER_NUM) + 1)
|         INTO :next_num
|         FROM DRDA/PART_ORDER;
|     EXEC SQL INSERT INTO DRDA/PART_ORDER
|         (ORDER_NUM, ORIGIN_LOC, ORDER_TYPE, ORDER_STAT, CREAT_TIME)
|         VALUES (:next_num, :loc, 'R', 'O', CURRENT_TIMESTAMP);
|     printf("***** ROP PROCESSING *****\n");
|     printf("ORDER NUMBER = %d \n\n",next_num);
|     printf("-----\n");
|     printf("  LINE    PART      QTY      \n");
|     printf("  NBR      NBR      REQUESTED\n");
|     printf("-----\n");
|     cont1 = cont1 + 1;
| } /* if cont1 == 0 */
|
| EXEC SQL INSERT INTO DRDA/PART_ORDLN
|     (ORDER_NUM, ORDER_LINE, PART_NUM, QUANT_REQ, LINE_STAT)
|     VALUES (:next_num, :cont1, :part_table, :eqq_table, 'O');
| line_count = line_count + 1;
| printf("  %d      %.5s      %d\n",line_count,part_table,eqq_table);
| cont1 = cont1 + 1;
| /* Exit function in connectable state */
|     EXEC SQL COMMIT;
|
| goto function_exit;
|
| error_tag:
|     error_function();
|
| function_exit:
|     ;
| } /* end of function process_order */
|
|

```

```

| /*****
| /* Function For Declaring Cursors
| /*****
| error_function() {
|
|     printf("*****\n");
|     printf("*      SQL ERROR      *\n");
|     printf("*****\n");
|     printf("SQLCODE      = %d\n",sqlca.sqlcode);
|     printf("SQLSTATE     = %5s",sqlca.sqlstate);
|     printf("\n*****\n");
|     EXEC SQL WHENEVER SQLERROR CONTINUE;
|     EXEC SQL ROLLBACK;
|     /* Reset Current Connection To Local Database
|     EXEC SQL CONNECT RESET;
|
|     exit(999);
| } /* end of function error_function
|
|
| /*****
| /* Mainline
| /*****
| main(int argc, char *argv[]) {
|
|     memcpy(local_db,argv[1],strlen(argv[1]));
|     memcpy(remote_db,argv[2],strlen(argv[2]));
|
|     /* Initialization
|     declare_cursors();
|     reset_tables();
|
|     /* Main Work
|     do {
|         calculate_order_quantity();
|         if (rop_table > quant_table + qty_req - qty_rec) {
|             process_order();
|             quant_table = qty_req = qty_rec = 0;
|         }
|     } while (sqlca.sqlcode == 0);
|     EXEC SQL COMMIT;
|     /* Reset Current Connection To Local Database
|     EXEC SQL DISCONNECT :local_db;
|
|     exit(0);
| } /* end of main
|

```

예: Java 프로그램

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```

| /*****
| /* PROGRAM NAME: SampJava */
| /* */
| /* DESCRIPTIVE NAME: Sample java application using DRDA */
| /* */
| /* FUNCTION: This module processes the PART_STOCK table and
| /* for each part below the ROP (REORDER POINT)
| /* creates a supply order. */
| /* */
| /* LOCAL TABLES: PART_STOCK */
| /* */
| /* REMOTE TABLES: PART_ORDER, PART_ORDLN, SHIPMENTLN */
| /* */
| /* COMPILE OPTIONS: */
| /* javac SampJava.java */
| /* */
| /* INVOKED BY: */
| /* java SampJava lcldbname rmtdbname */
| /*****
| import java.sql.*;
|
| public class SampJava {
|     private static String JDBCdriver = "com.ibm.db2.jcc.DB2Driver";
|     private static String part_table = " "; /* part number in table part_stock */
|     private static long line_count = 0; /* total number of order lines */
|     private static long eqq_table = 0; /* reorder quantity , tbl part_stock */
|     private static long quant_table = 0; /* quantity in stock, tbl part_stock */
|     private static long rop_table = 0; /* reorder point , tbl part_stock */
|     private static int contl = 0; /* continuation line, tbl order_line */
|     private static short next_num = 0; /* next order nbr, table part_order */
|
|     /*****
|     /* Method For Reseting Environment */
|     /*****
|     private static void resetTables(Connection rmtConn) throws SQLException {
|
|         Statement stmt1 = rmtConn.createStatement();
|
|         /* Clean up for rerunability in test environment */
|         stmt1.executeUpdate("DELETE FROM DRDA.PART_ORDLN WHERE ORDER_NUM IN " +
|             " (SELECT ORDER_NUM FROM DRDA.PART_ORDER " +
|             " WHERE ORDER_TYPE = 'R')");
|         stmt1.executeUpdate("DELETE FROM DRDA.PART_ORDER WHERE ORDER_TYPE = 'R'");
|         stmt1.close();
|         rmtConn.commit();
|
|     } /* function delete_for_rerun */
|
|
|
|

```

```

|
| /*****
| /* Method For Calculating Order Quantity */
| /*****
| private static void calculateOrderQuantity(Connection lclConn, Connection rmtConn, String loc)
| throws SQLException {
|     PreparedStatement prpStmt1;
|     PreparedStatement prpStmt2;
|     ResultSet rsltSet1;
|     ResultSet rsltSet2;
|     short ord_table = 0;          /* order nbr.          , tbl order_line */
|     short orl_table = 0;         /* order line       , tbl order_line */
|
|     prpStmt1 = lclConn.prepareStatement("SELECT PART_NUM, PART_QUANT, PART_ROP, PART_EOQ " +
|                                        "          FROM DRDA.PART_STOCK WHERE PART_ROP > PART_QUANT AND " +
|                                        "          PART_NUM > ? ORDER BY PART_NUM");
|
|     prpStmt1.setString(1,part_table);
|     rsltSet1 = prpStmt1.executeQuery();
|     if (rsltSet1.next() == false) {
|         System.out.println("-----");
|         System.out.println("NUMBER OF LINES CREATED = " + line_count);
|         System.out.println("-----");
|         System.out.println("***** END OF PROGRAM *****");
|         rop_table = 0;          /* no (more) orders to process */
|     }
|     else {
|         /* available qty = Stock qty + qty in order - qty received */
|         part_table = rsltSet1.getString(1);
|         quant_table = rsltSet1.getLong(2);
|         rop_table = rsltSet1.getLong(3);
|         eoq_table = rsltSet1.getLong(4);
|         long qty_rec = 0;
|
|         prpStmt2 = rmtConn.prepareStatement("SELECT A.ORDER_NUM, ORDER_LINE, QUANT_REQ " +
|                                            "          FROM DRDA.PART_ORDLN A, DRDA.PART_ORDER B " +
|                                            "          WHERE PART_NUM = ? AND LINE_STAT <> 'C' AND " +
|                                            "          A.ORDER_NUM = B.ORDER_NUM AND ORDER_TYPE = 'R'");
|
|         prpStmt2.setString(1,part_table);
|         rsltSet2 = prpStmt2.executeQuery();
|         while (rsltSet2.next()) {
|             ord_table = rsltSet2.getShort(1);
|             orl_table = rsltSet2.getShort(2);
|             long qty_table = rsltSet2.getLong(3);
|             qty_rec = qty_rec + qty_table;
|         }
|         rsltSet2.close();
|
|
|
|
|
|

```

```

| prpStmt2 = rmtConn.prepareStatement("SELECT SUM(QUANT_RECV) FROM DRDA.SHIPMENTLN " +
|                                     "         WHERE ORDER_LOC = ? AND ORDER_NUM = ? AND " +
|                                     "         ORDER_LINE = ?");
|
|     prpStmt2.setString(1,loc);
|     prpStmt2.setShort(2,ord_table);
|     prpStmt2.setShort(3,orl_table);
|     rsltSet2 = prpStmt2.executeQuery();
|     rsltSet2.next();
|     long qty_table = rsltSet2.getLong(1);
|     qty_rec = qty_rec + qty_table;
|     rsltSet2.close();
|     prpStmt2.close();
| }
| rsltSet1.close();
| prpStmt1.close();
|
| } /* end of calculate_order_quantity                                     */
|
| /*****
| /* Method For Processing Orders                                     */
| /*****
| private static void processOrder(Connection rmtConn, String loc) throws SQLException {
|     PreparedStatement prpStmt1;
|     ResultSet rsltSet1;
|
|     /* insert order and order_line in remote database                                     */
|     if (cont1 == 0) {
|         prpStmt1 = rmtConn.prepareStatement("SELECT (MAX(ORDER_NUM) + 1) FROM DRDA.PART_ORDER");
|         rsltSet1 = prpStmt1.executeQuery();
|         rsltSet1.next();
|         next_num = rsltSet1.getShort(1);
|         rsltSet1.close();
|         prpStmt1 = rmtConn.prepareStatement("INSERT INTO DRDA.PART_ORDER (ORDER_NUM, ORIGIN_LOC,
| ORDER_TYPE, ORDER_STAT, CREAT_TIME) " +
|                                     "         VALUES (?, ?, 'R', 'O', CURRENT_TIMESTAMP)");
|
|         prpStmt1.setShort(1,next_num);
|         prpStmt1.setString(2,loc);
|         prpStmt1.executeUpdate();
|         System.out.println("***** ROP PROCESSING *****");
|         System.out.println("ORDER NUMBER = " + next_num);
|         System.out.println("-----");
|         System.out.println("  LINE      PART      QTY      ");
|         System.out.println("  NBR      NBR      REQUESTED");
|         System.out.println("-----");
|         cont1 = cont1 + 1;
|     } /* if cont1 == 0                                               */
|
|
|
|

```



```

|
| prpStmt1 = rmtConn.prepareStatement("INSERT INTO DRDA.PART_ORDLN (ORDER_NUM, ORDER_LINE,
| PART_NUM, QUANT_REQ, LINE_STAT) " +
|
|         "          VALUES (?, ?, ?, ?, '0')");
|
|     prpStmt1.setShort(1,next_num);
|     prpStmt1.setInt(2,cont1);
|     prpStmt1.setString(3,part_table);
|     prpStmt1.setLong(4,eqq_table);
|     prpStmt1.executeUpdate();
|     line_count = line_count + 1;
|     System.out.println(" " + line_count + "          " + part_table + "          " + eqq_table + "");
|     cont1 = cont1 + 1;
|     prpStmt1.close();
|
| } /* end of function processOrder                                */
|
|
| /*****
| /* Method For Displaying Errors                                */
| /*****
| private static void errorFunction(SQLException e, Connection lclConn, Connection rmtConn) {
|
|     System.out.println("*****");
|     System.out.println("*      SQL ERROR      *");
|     System.out.println("*****");
|     System.out.println("SQLCODE   = " + e.getErrorCode());
|     System.out.println("SQLSTATE  = " + e.getSQLState());
|     System.out.println("*****");
|     try {
|         lclConn.rollback();
|         rmtConn.rollback();
|     }
|     catch (SQLException uowErr) {
|     }
|
| } /* end of function errorFunction                                */
|
|
|

```

```

| /*****
|  /* Mainline
|  *****/
| public static void main(String[] args) {
|     String User = "myuser";
|     String Password = "mypwd";
|     String lclUrl = null;
|     String rmtUrl = null;
|     String loc = "SQLA";          /* dealer's database name
|     Connection lclConn = null;
|     Connection rmtConn = null;
|
|     try {
|         Class.forName(JDBCdriver).newInstance();
|     }
|     catch (Exception e) {
|         System.out.println("Error: Failed to load DB2 driver.");
|         System.exit(1);
|     }
|
|     try {
|         lclUrl = "jdbc:db2:" + args[0];
|         lclConn = DriverManager.getConnection(lclUrl, User, Password);
|         rmtUrl = "jdbc:db2:" + args[1];
|         rmtConn = DriverManager.getConnection(rmtUrl, User, Password);
|     }
|     catch (Exception e) {
|         System.out.println("Error: Failed to get database connections.");
|         System.exit(1);
|     }
|
|     try {
|         /* Initialization
|         resetTables(rmtConn);
|
|         /* Main Work
|         do {
|             calculateOrderQuantity(lclConn, rmtConn, loc);
|             if (rop_table > quant_table) {
|                 processOrder(rmtConn, loc);
|                 quant_table = 0;
|             }
|         } while (rop_table != 0);
|
|         /* End Work
|         lclConn.commit();
|         rmtConn.commit();
|     }
|     catch (SQLException e) {
|         e.printStackTrace();
|         errorFunction(e, lclConn, rmtConn);
|         System.exit(1);
|     }
| }
|

```

예: 프로그램 출력

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```
***** 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 28. 예: 프로그램 출력

사용자 FAQ

이 주제에서는 iSeries 서버의 네트워크(유사 환경)에서 분산 관계형 데이터베이스에 대한 iSeries 지원을 중점적으로 설명합니다.

많은 분산 관계형 데이터베이스 구현이 다른 DRDA 지원 플랫폼의 네트워크에 있습니다. 이 주제에서는 상이한 DRDA 환경에서 iSeries 서버를 사용할 때 고려해야 할 추가 정보 및 기술 리스트를 제공합니다.

이 주제는 다른 특정 IBM 제품으로 작업할 때 고려해야 할 몇 가지 조건에 대해 설명합니다. 포괄적으로 나열하지는 않습니다. 여기에서 설명하는 내용과 유사한 많은 문제나 조건은 사용하는 어플리케이션에 따라 많이 다릅니다. *IBM SQL* 참조서 볼륨 2, SC26-8416 또는 *DRDA 어플리케이션 프로그래밍 안내서*, SC26-4773에서 다양한 IBM 플랫폼 간의 차이점에 대해 좀 더 자세한 정보를 얻을 수 있습니다.

관련 참조

99 페이지의 『iSeries 서버 지원』

iSeries 서버의 작업에 대한 디폴트 CCSID는 CHGJOB(작업 변경) 명령을 사용하여 지정됩니다.

분산 관계형 데이터베이스로 연결

비iSeries 어플리케이션 리퀘스터(AR)에서 iSeries용 DB2 Universal Database 어플리케이션 서버(AS)로 연결할 때 CCSID 65535 태그가 표시된 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(실제 파일 변경) 명령을 사용하십시오. 실제 파일 위에 논리 파일을 빌드한 경우, 표시된 오류 메시지(CPD#22D)의 회복 주제에 제공된 지시사항을 따르십시오.

iSeries 시스템 값 QCCSID

iSeries 서버는 QCCSID 시스템 값을 65535로 설정하여 출하됩니다.

CCSID가 태그로 붙은 데이터는 수신 서버에 의해 변환되지 않습니다. iSeries 서버 어플리케이션 리퀘스터(AR)가 이 CCSID를 사용 중인 경우 다른 서버에 연결하지 못할 수 있습니다. 또한 다른 서버에서 어플리케이션을 작성하는 데 이 CCSID 태그가 붙은 소스 파일을 사용하지 못할 수 있습니다.

연결 시간에 사용되는 CCSID는 CCSID 작업에 의해 결정됩니다. 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. CHGJOB(작업 변경) 명령을 사용하여 CCSID 작업을 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 자료를 변환해야 하는 사용자만 영향을 받습니다. iSeries용 DB2 Universal Database 어플리케이션 서버(AS)로 작업하는 경우 AS가 사용하는 사용자 프로파일을 변경해야 합니다.

다른 어플리케이션 리퀘스터가 작업 CCSID 65535를 사용하여 iSeries용 DB2 UDB 어플리케이션 서버에 연결하면 작업은 작업 디폴트 CCSID를 사용하도록 전환됩니다. 작업 디폴트 CCSID는 작업의 언어 ID(LANGID)에 의해 결정됩니다. 이 경우에는 보다 좋은 성능을 위해 작업의 CCSID를 65535이 아닌 값으로 전환해야 합니다. 예를 들어, CCSID 값을 서버 작업이 실행되는 사용자 프로파일의 값으로 변경할 수 있습니다.

사용자 프로파일의 디폴트 CCSID 값은 *SYSVAL입니다. 이 값은 QCCSID 시스템값을 참조합니다. CHGSYSVAL(시스템 값 변경) 명령으로 모든 사용자 프로파일에서 사용되는 QCCSID 시스템 값을 변경할 수 있습니다. 서버에 있는 대부분의 사용자(모든 사용자가 아님)를 나타내는 CCSID를 선택할 수 있습니다.

작업 중인 서버가 작업이나 서버가 사용하는 CCSID를 지원하는지의 여부는 작업 기록부, SQLCA 또는 SQL 진단 영역에서 다음 인디케이터를 참조하십시오.

메세지 SQ30073

SQLCODE 또는 DB2_RETURNED_SQLCODE

-30073

SQLSTATE

58017

텍스트 분산 자료 관리(DDM) 매개변수 X'0035'가 지원되지 않습니다.

메세지 SQL0332

SQLCODE 또는 DB2_RETURNED_SQLCODE

-332

SQLSTATE

57017

텍스트 CCSID &1과 CCSID &2 사이의 전체 변환은 유효하지 않습니다.

관련 개념

i5/OS 국제화

관련 참조

CHGJOB(작업 변경) 명령

CHGSYSVAL(시스템 값 변경) 명령

DB2 Universal Database for z/OS 및 DB2 Universal Database Server for VM 데이터베이스 관리자에 대한 CCSID(코드화 문자 세트 ID) 변환 고려사항

iSeries용 DB2 Universal Database 및 다른 DB2 데이터베이스 사이의 차이점 중 하나는 iSeries 시스템이 더 큰 CCSID 세트를 지원한다는 점입니다. 따라서 다른 데이터베이스 관리자가 자료에서 문자 변환을 시도할 때 오류가 발생할 수 있습니다(SQLCODE -332 및 SQLSTATE 57017).

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 Universal Database for z/OS 또는 DB2 Universal Database Server for VM 어플리케이션 리퀘스터(AR)에서 이들 필드를 선택할 경우 DB2 Universal Database for z/OS 및 DB2 Universal Database Server for VM 데이터베이스가 이 CCSID로의 변환을 지원하지 않을 수 있으므로 SELECT문이 실행되지 않을 수 있습니다.

이 오류를 방지하려면 DB2 Universal Database for z/OS 데이터베이스 또는 DB2 Universal Database Server for VM AR을 다음 중 하나로 실행되도록 변경해야 합니다.

- iSeries SQL 카탈로그 표의 DBCS 개방 필드와 동일한 혼합 바이트 CCSID.
- iSeries SQL 카탈로그 표에 있는 DBCS 개방 필드의 혼합 바이트 CCSID 자료일 경우 서버가 자료 변환을 허용하는 CCSID. iSeries SQL 카탈로그 표 DBCS 개방 필드의 자료가 모두 1바이트 자료일 경우 이 CCSID는 1바이트 CCSID가 될 수 있습니다.

이렇게 하려면 DB2 Universal Database for z/OS 또는 DB2 Universal Database Server for VM에서 지원되는 CCSID 변환을 일부 분석하여 서버를 올바르게 변경할 수 있도록 해야 합니다. 이 오류를 처리하는 방법에 대한 특정 정보는 *DB2 UDB for z/OS Administration Guide*를 참조하십시오.

Linux, UNIX 및 Windows용 DB2 Universal Database에서 연결하려고 시도할 때 왜 SQL5048N 메시지가 표시됩니까?

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. Linux, UNIX 및 Windows용 DB2 Universal Database, IBM DB2 Universal Driver for SQLJ 및 JDBC, 그리고 기타 어플리케이션 리퀘스터는 필요한 SQL 패키지를 빌드하기 위해 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 오류 원인이 있다고 생각되면 Authorized Problem Analysis Report 웹 사이트를 참조하십시오. 검색 필드에 APAR II12722를 입력하십시오.

iSeries 파일을 저널링해야 합니까?

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.

이 질문에 대한 답은 『조회 자료는 성능 향상을 위해 언제 블로킹됩니까?』의 질문과 밀접한 관련이 있습니다.

분리 레벨 변경 방법에 대한 예는 다음과 같습니다.

- Linux, UNIX 및 Windows용 DB2 Universal Database 사전컴파일러는 ISOLATION UR 매개변수를 사용하여 미확약 읽기를 지정합니다.
- Linux, UNIX 및 Windows용 DB2 Universal Database 명령행 프로세서(CLP)는 DBM CHANGE SQLISL TO UR 명령을 사용하여 미확약 읽기를 지정합니다.
- Linux, UNIX 및 Windows용 DB2 Universal Database 명령행 프로세서(CLP)는 DBM CHANGE SQLISL TO NC 명령을 사용하여 비확약을 지정합니다.
- JDBC 클라이언트는 연결 등록 정보 분리 레벨을 TRANSACTION_READ_UNCOMMITTED로 설정하여 미확약 읽기를 지정합니다.

조회 자료는 성능 향상을 위해 언제 블로킹됩니까?

다음 조건이 모두 해당되지 않으면 조회 자료가 블로킹됩니다.

- 커서가 갱신 가능합니다(주 1 참조).
- 커서가 잠재적으로 갱신 가능합니다(주 2 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 블록 됩니다.

- 커서가 읽기 전용입니다(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. 커서가 읽기 전용(주 3 참조)이 아니고 다음 항목 중 하나에 해당할 경우 갱신이 가능합니다.
 - SELECT문에 FOR UPDATE OF절이 들어 있습니다.
 - 커서에 대한 UPDATE 또는 DELETE WHERE CURRENT OF 구가 프로그램 내에 있습니다.
2. 커서가 읽기 전용(주 3 참조)이 아니고 프로그램이 동적 명령문을 포함하며 SQLPREP 또는 SQLBIND 에서 BLOCKING UNAMBIG 사전컴파일 또는 바인드 옵션이 사용된 경우 커서는 잠재적으로 갱신 가능합니다.
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?

Linux, UNIX 및 Windows용 DB2 Universal Database에 사용된 클라이언트 지원은 해당 코드가 없는 호스트 SQLCODE 및 토큰을 보고할 때 SQL0969N 메시지를 리턴합니다.

다음은 SQL0969N 메시지의 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".

코드 및 토큰을 해석하려면 DSPMSGD(메세지 설명 표시) 명령을 사용하십시오.

DSPMSGD SQL7008 MSGF(QSQLMSG)

선택 옵션 1(화면 메세지 텍스트)과 서버는 형식화된 메세지 텍스트 표시 화면을 제공합니다. 메세지의 세 개 토큰은 화면에 &1, &2 및 &3으로 표시됩니다. 메세지 예의 이유 코드는 3으로, 화면 맨 아래의 리스트에서 코드 3을 가리킵니다.

```
형식화된 메세지 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 확약 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 Enter 키를 누르십시오.

F3=종료 F11=형식화되지 않은 메세지 텍스트 F12=취소
```

관련 참조

DSPMSGD(메세지 설명 표시) 명령

WHERE절의 호스트 변수 유형은 성능에 어떤 영향을 미칠 수 있습니까?

iSeries 서버의 성능 저하 원인 중 하나는 클라이언트가 C 프로그램에서 SELECT문의 WHERE절에서 비교에 대해 부동 소수점 변수를 사용하는 것입니다.

오퍼레이팅 시스템이 해당 열에 대해 자료 변환을 수행해야 하는 경우 이는 해당 열에서 색인을 사용할 수 없게 합니다. 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.

관련 개념

데이터베이스 성능에 대한 프로그래밍 기술

규정되지 않은 표 및 뷰 이름을 분석하기 위해 라이브러리 리스트를 사용할 수 있습니까?

iSeries 서버는 Linux, UNIX 및 Windows용 DB2 Universal Database 제품을 사용하는 프로그램과 같은 비iSeries DRDA 클라이언트에서 iSeries용 DB2 Universal Database 자료에 액세스할 때 오퍼레이팅 시스템 명령 옵션을 사용하는 데 대해 제한된 기능을 지원합니다.

이전에는 상이한 DRDA 클라이언트로부터 연결할 때, SQL 명령 옵션만 사용할 수 있었습니다. 시스템 명령은 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. 규정된 오브젝트명을 이들이 상주하는 라이브러리 또는 콜렉션으로부터 분리시키기 위해 마침표(.) 대신 슬래시(/)를 사용합니다.
4. 어떤 권한부여 특성은 변경됩니다.

세부사항은 SQL 참조의 서버 명령에 대해 읽으십시오. 권한 부여에 관한 자세한 정보는 계획 및 설계를 참조하십시오.

Linux, UNIX 및 Windows용 DB2 Universal Database는 두 가지 프로그램 준비 명령인 PREP(사전컴파일) 명령과 BIND 명령에서 일반 바인드 옵션의 스펙을 지원합니다. Windows 일괄처리 파일에서 가져온 다음 예에서와 같이 이들 명령 중 하나에 i5/OS 명령 옵션을 지정할 수 있습니다.

Linux, UNIX 및 Windows용 DB2 Universal Database, 버전 8 이상의 경우:

```
DB2 PREP %1.SQC BINDFILE OS400NAMING SYSTEM ...
DB2 BIND %1.BND OS400NAMING SYSTEM ...
```

For Linux, UNIX 및 Windows용 DB2 Universal Database, 버전 8보다 이전 버전의 경우:

```
DB2 PREP %1.SQC BINDFILE GENERIC 'OS400NAMING SYSTEM' ...
DB2 BIND %1.BND GENERIC 'OS400NAMING SYSTEM' ...
```

Windows 개발 플랫폼에서는 일반 옵션인 이름/값 쌍 주위에 작은따옴표를 사용하는 점을 주지하십시오. AIX 또는 UNIX 플랫폼에서는 큰따옴표를 사용해야 합니다.

주: OS/400 V4R5 및 V5R1의 경우, 이 옵션의 이름은 AS400NAMING이며, OS400NAMING이 아닙니다.

SYSTEM 외에 OS400NAMING 옵션의 유일하게 유효한 값은 SQL이며 이는 디폴트 값이자 이 기능이 소개되기 전에 비 iSeries 클라이언트에서 사용할 수 있는 유일한 옵션입니다.

(BIND) 명령에 OS400NAMING 옵션을 사용하고 (PREP) 명령에는 사용하지 않을 경우 서버 플랫폼에서 SQL 오류가 감지되더라도 바인드 파일을 작성해야 한다는 것을 나타내는 매개변수를 (PREP) 명령에 코딩해야 합니다. In the case of Linux, UNIX 및 Windows용 DB2 Universal Database의 경우 이와 같은 용도에는 SQLERROR CONTINUE 매개변수를 사용하십시오. 이 기능은 '제한된' 것으로 설명되는데, 이는 특정 상황에서는 리모트 서버에서 실행되기로 한 SQL문을 클라이언트측 소프트웨어가 분석할 수 있기 때문입니다. 서버 명령에 요구된 대로 테이블 ID에서 스키마 ID를 구분하기 위해 마침표 대신 슬래시를 사용할 경우 이 명령문은 적절하지 않은 구문을 가졌기 때문에 거부될 수 있습니다.

Linux, UNIX 및 Windows용 DB2 Universal Database 사용자가 일반 EBCDIC 순서 대신 iSeries 서버의 DRDA 작업 NLSS 정렬 순서표가 사용되도록 지정할 수 있습니까?

iSeries 서버는 일반 바인드 옵션을 인식하는데, 이 옵션을 사용하면 프로그램을 준비하는 클라이언트가 Linux, UNIX 및 Windows용 DB2 Universal Database에서 실행될 수 있거나 일반 바인드 옵션을 지원하는 다른 클라이언트가 iSeries 서버가 클라이언트의 요구가 실행되는 서버 작업과 연관된 NLSS 정렬 순서를 사용하도록 요구할 수 있습니다.

이 기능은 V5R1의 경우 PTF SI00174를 통해 사용할 수 있습니다. 후속 릴리스의 기본 오퍼레이팅 시스템에 있습니다.

이 향상된 기능을 이용하기로 선택할 경우 DB2 Universal Database에 SQL 패키지를 다시 작성해야 합니다. 왜냐하면 클라이언트 시스템에서 JOBRUN 값을 갖는 SORTSEQ 일반 바인드 옵션을 사용함으로써 새 정렬 순서 옵션이 필요하기 때문입니다.

바인드 옵션을 사용하면 사용자가 흔히 사용되는 EBCDIC 순서 대신 iSeries 서버에서 DRDA 작업의 NLSS 정렬 순서 표를 사용하도록 지정할 수 있습니다. 이전에는, 상이한 DRDA 클라이언트로부터 연결할 때, EBCDIC 순서가 사용되도록 하는 디폴트 *HEX 옵션만 사용할 수 있었습니다.

이 피처는 DRDA 일반 바인드 기능을 지원하는 DRDA 어플리케이션 리퀘스터에서 사용할 수 있습니다. 이는 클라이언트 개발 플랫폼 및 실행 환경으로 Windows에서 실행되는 Linux, UNIX 및 Windows용 DB2 Universal Database 6.1 FixPack 1을 사용하여 제한된 테스트를 거쳤습니다. Linux, UNIX 및 Windows용 DB2 Universal Database는 두 가지 프로그램 준비 명령인 PREP(사전컴파일) 명령 및 BIND 명령에서 일반 바인드 옵션의 스펙을 지원합니다. JOBRUN 정렬 순서는 Windows 일괄처리 파일에서 가져온 다음 예에서처럼 이들 중 하나에서 지정됩니다.

```
DB2 PREP %1.SQC BINDFILE SORTSEQ JOBRUN...
DB2 BIND %1.BND SORTSEQ JOBRUN...
```

주: Windows 개발 플랫폼에서는 일반 옵션 이름/값 쌍 주위에 작은 따옴표를 사용하지만 AIX 또는 UNIX 플랫폼에서는 큰 따옴표를 사용합니다.

SORTSEQ 옵션에 다른 유일한 유효한 값은 HEX이며, 이는 디폴트 값이며 이 피처가 소개되기 전에는 비 iSeries 클라이언트에서 사용할 수 있는 유일한 옵션입니다.

조회를 실행할 때 행이 리턴되지 않는 이유는 무엇입니까?

이 문제점의 한 가지 가능한 원인은 Linux, UNIX 및 Windows용 DB2 Universal Database 데이터베이스 통신 서비스 디렉토리에 iSeries 서버에 대한 항목을 추가하지 못했기 때문입니다.

iSeries용 DB2 Universal Database와 대화하려면 어떤 레벨의 Linux, UNIX, Windows용 DB2 Universal Database가 필요합니까?

대화하려면 다음 수정팩이 필요합니다.

- Linux, UNIX 및 Windows용 DB2 Universal Database 버전 7 수정팩 10

- Linux, UNIX 및 Windows용 DB2 Universal Database 버전 8 수정팩 4

Linux, UNIX 및 Windows용 DB2 Universal Database  웹 사이트에서 이들 수정팩을 얻을 수 있습니다.

Linux, Unix 및 Windows용 DB2 Universal Database 버전 8에서 iSeries까지 사용 가능한 화면 이동 커서 지원을 얻는 방법은 무엇입니까?

클라이언트에 FixPak 4 이상을 사용해야 합니다.

FixPak 4를 사용 중인 경우 다음 조치 중 **하나**를 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 환경에서 상호조작에 대한 기타 추가 정보

이 주제는 Linux, UNIX 및 Windows용 DB2 Universal Database와 함께 iSeries용 DB2 Universal Database 사용에 대한 추가 정보를 제공합니다. 추가 내용은 OS/2 플랫폼에서 제품을 테스트한 경험으로 개발되었지만 제품을 포트한 모든 환경에 적용되는 것으로 알려져 있습니다.

Linux, UNIX 및 Windows용 DB2 Universal Database 대신 DB2 Connect™

사용자는 어플리케이션 리퀘스터(클라이언트) 기능 대신 DRDA 어플리케이션 서버 기능을 수행하는 데 필요한 제품을 혼동하는 경우가 종종 있습니다. AR은 종종 DB2 Connect로 참조되며, AR과 AS는 둘 다 Linux, UNIX 및 Windows용 DB2 Universal Database로 참조됩니다. DB2 UDB는 다음 제품을 참조합니다.

- DB2 Universal Database for AIX
- DB2 Universal Database for HP-UX
- DB2 Universal Database for Linux
- DB2 Universal Database for Sun Solaris
- DB2 Universal Database 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.

표 및 컬렉션 명명

DRDA 어플리케이션이 액세스하는 SQL표의 이름은 세 부분으로 이루어집니다. 첫 번째 부분은 데이터베이스 명이고 두 번째 부분은 컬렉션 ID이며 세 번째 부분은 기본 표 이름입니다. 첫 번째 두 부분은 선택적입니다. iSeries용 DB2 UDB는 두 번째 레벨에서 컬렉션(또는 라이브러리)명으로 표 이름을 규정합니다. 표는 iSeries용 DB2 UDB 데이터베이스에 있습니다.

V5R2 및 독립형 보조 기억장치 풀이 등장하기 전에는 각 iSeries 서버에 대해 하나의 데이터베이스만 있었습니다. 그러나 DB2 UDB에서 표는 사용자 ID(표 작성자의 사용자 ID)로 규정되며 플랫폼의 여러 데이터베이스 중 하나에 있습니다. Linux, UNIX 및 Windows용 DB2 Universal Database는 컬렉션 ID에 대해 사용자 ID를 사용하는 동일한 개념을 가집니다.

조회되는 표의 이름이 컬렉션명이 없이 지정된 경우 Linux, UNIX 및 Windows용 DB2 Universal Database에서 iSeries용 DB2 UDB로의 동적 조회는 디폴트 컬렉션명에 대한 목표측 작업(iSeries 서버에 있는)의 사용자 ID를 사용합니다. 이는 사용자가 예상한 것이 아닐 수 있으며 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 작성하려 할 수 있습니다. UDB의 경우에는 iSeries용 DB2 UDB에서처럼 실제 컬렉션은 없고 작성자의 사용자 ID인 표 규정자만 있음을 기억하십시오.

APPC 통신 설정

Linux, UNIX 및 Windows용 DB2 Universal Database를 AR로 사용하고 DB2 UDB를 AS로 사용하여 APPC를 사용할 때 워크스테이션에 대해 작성된 제어기 및 장치로 i5/OS 통신을 적절히 구성해야 합니다.

RDB 디렉토리 설정

iSeries 서버가 연결할 각 DB2 UDB 데이터베이스에 대한 RDB 디렉토리에 항목을 추가할 때 ADDRDBDIRE(관계형 데이터베이스 디렉토리 항목 추가) 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 잘 알려진 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 . . . . . : 내 UDB 서버

```

Linux, UNIX 및 Windows용 DB2 UDB 및 IBM DB2 Universal Driver for SQLJ 및 JDBC에서 사용되는 NULLID 패키지 작성 방법

iSeries용 DB2 UDB의 자료에 액세스하기 위해 Linux, UNIX 및 Windows용 DB2 Universal Database를 사용하기 전에 iSeries 서버에서 Linux, UNIX 및 Windows용 DB2 Universal Database 유틸리티 및 어플리케이션 프로그램에 대한 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 데이터베이스에서 SQL 패키지를 작성합니다.

Linux, UNIX 및 Windows용 DB2 Universal Database를 iSeries용 DB2 UDB 서버에 바인드하려면 다음을 수행하십시오.

1. CONNECT TO rdbname
2. Bind path@ddcs400.lst BLOCKING ALL SQLERROR CONTINUE MESSAGES DDCCS400.MGS GRANT PUBLIC

위의 path@ddcs400.lst 매개변수에 있는 'path'를 디폴트 경로 C:\SQLLIB\BNDW(INTEL 플랫폼이 아닌 경우는 c:\sqllib/bin/) 또는 디폴트 디렉토리에 설치하지 않은 경우에는 사용자 값으로 대체하십시오.

3. CONNECT RESET

대화식 SQL 패키지 설정 방법

대화식 SQL을 사용하려면 i5/OS에 DB2 UDB 조회 관리자 및 SQL 개발 킷 제품을 설치해야 합니다. DB2 Universal Database의 자료에 액세스하려면 다음을 수행하십시오.

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 SQL을 사용하기 위한 설정의 일부로서 GRANT EXECUTE ON PACKAGE QSQL400.QSQLabcd TO PUBLIC(또는 특정 사용자)문을 사용하여 다른 사용자가 대화식 SQL을 위해 PC에 작성된 SQL PKG를 사용하게 할 수도 있습니다. 다음 GRANT문의 abcd에 대한 실제 값은 다음 표에서 판별할 수 있는데, 이 표는 패키지를 작성할 때 사실상 다양한 옵션 세트에 대해 패키지 이름을 제공합니다. 예를 들어, 패키지를 작성할 때 날짜에 *ISO, 시간에 *ISO, 확약 제어에 *CS, 스트링 분리문자에 작은 따옴표, 그리고 문자 부속 유

형에 1바이트와 같은 옵션을 사용한 경우 GRANT EXECUTE ON PACKAGE QSQL400.QSQL0200 TO *some-user*문을 사용하게 됩니다.

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	확약 제어 십진 분리문자	0 = *CS 확약 제어 마침표 십진 분리문자 1 = *CS 확약 제어 쉼표 십진 분리문자 2 = *RR 확약 제어 마침표 십진 분리문자 3 = *RR 확약 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 = 따옴표 STRING 분리문자, 1바이트 문자 부속 유형 3 = 따옴표 STRING 분리문자, 2바이트 문자 부속 유형

조회 달기

Linux, UNIX 및 Windows용 DB2 Universal Database는 이제 암시적 또는 명시적으로 조회를 닫을 때 읽기 잠금을 해제하도록 요구하는 옵션을 제공합니다. iSeries 서버의 경우처럼 서버가 요구를 받아들이지 않아도 이는 오류로 간주되지 않습니다. Linux, UNIX 및 Windows용 DB2 Universal Database는 서버가 더 읽을 행이 없을 때 화면이동할 수 없는 커서에 대한 조회를 암시적으로 닫아야 하는지 여부를 지정하는 다른 새 옵션을 제공합니다. 이전에는 서버가 이러한 결정을 했습니다. iSeries AS는 V5R3에서 이러한 새 피처를 지원합니다.

이종 환경에서 사용자 ID 암호의 최대 길이

어플리케이션 리퀘스터(AR)로 실행 중인 iSeries용 DB2 UDB를 사용하면 상이한 어플리케이션 서버(AS)로 실행할 때 사용자 ID 및 암호가 10자를 초과할 수 있습니다. 정확한 제한치는 사용되는 특정 인터페이스에 대한 설명에서 지정됩니다. 예를 들어, SQL CONNECT문의 제한사항은 SQL 참조 주제를 참조하십시오.

DB2 UDB Server for VM에서 대화식 SQL 패키지 작성

DB2 Universal Database Server for VM에서 콜렉션명은 사용자 ID와 동의어입니다. DB2 Universal Database Server for VM 어플리케이션 서버에서 대화식 SQL 또는 iSeries 조회 관리자로 사용할 패키지를 작성하려면 i5/OS 오퍼레이팅 시스템에서 QSQL400의 사용자 ID를 작성하십시오. 이 사용자 ID는 DB2 Universal Database Server for VM 어플리케이션 서버에서 필요한 모든 패키지를 작성하는 데 사용될 수 있습니다. 그런 다음 사용자는 사용자 자신의 사용자 ID를 사용하여 i5/OS의 iSeries 조회 관리자 또는 대화식 SQL을 통해 DB2 Universal Database Server for VM에 액세스할 수 있습니다.

>> 접두부 다음에는 추적점을 식별하는 7자 스트링이 있습니다. 처음 두 문자인 RW는 구성요소를 식별합니다. 두 번째 두 문자는 RW 기능이 수행되는 것을 식별합니다. QY는 DDM 명령 OPNQRY, CNTQRY 및 CLSQRY에 해당하는 조회 기능을 나타냅니다. EX는 DDM 명령 EXCSQLSTT, EXCSQLIMM 및 PRPSQLSTT에 해당하는 EXECUTE 기능을 나타냅니다.

이들 각 기능에 해당하는 프로그램 모듈은 작업 추적이 분산 SQL 액세스 조작의 어플리케이션 리퀘스터(AR) 끝에서 수행되었는지 또는 어플리케이션 서버(AS) 끝에서 수행되었는지에 따라 달라집니다. 이 프로세스와 조회 기능을 실행하는 AR 모듈은 QRWSEXEC와 QRWSQRY이고 AS 모듈은 QRWTEXEC와 QRWTQRY입니다.

7바이트 추적점 ID의 마지막 두 문자는 덤프가 수행된 지점 또는 덤프된 자료의 특성을 나타냅니다. 예를 들면, SN은 AR이나 AS에서 송신된 자료 스트림에 해당되고 RC는 AR이 수신한 자료 스트림에 해당됩니다.

관련 참조

TRCJOB(작업 추적) 명령

예: RW 추적 자료 분석

이 주제는 해당 예에서 자료 스트림을 구성하는 요소에 대해 살펴봅니다.

261 페이지의 그림 29에 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.

분산 관계형 데이터베이스 구조(DRDA) 자료 스트림의 해석에 대한 자세한 정보는 *Distributed Relational Database Architecture Reference* 및 *Distributed Data Management Level 4.0 Architecture Reference* 서적을 참조하십시오. 이들 문서는 www.opengroup.org/dbiop/index.htm의 웹 사이트에서 구할 수 있습니다.

추적 자료는 추적점 ID의 끝을 표시하는 콜론(:) 다음에 표시됩니다. 이 예에서 자료 스트림의 처음 6바이트에는 DDM 자료 스트림 구조(DSS) 헤더가 들어 있습니다. 이 DSS 헤더의 처음 두 바이트는 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). 다섯 번째, 여섯 번째 바이트에는 DDM 요구 상관 ID가 있습니다.

그 다음 두 바이트 X'0010'(소수 16자리)는 그 다음 DDM 오브젝트의 길이를 나타내고, 이 경우 이 DDM 오브젝트는 그 뒤에 오는 OPNQRYRM 응답 메시지에 대한 코드점인 X'2205'로 식별됩니다.

16바이트 응답 메시지 다음에는 이 응답 메시지 뒤에 오는 응답 오브젝트의 6바이트 DSS 헤더가 옵니다. 첫 번째 응답 오브젝트는 X'241A' 코드점으로 식별되며 QRYDSC 오브젝트입니다. 이 예에서 두 번째 응답 오브젝트는 X'241B' 코드점으로 식별되는 QRYDTA 구조입니다(추적 출력에서 두 행으로 나뉘어져 있습니다). OPNQRYRM 코드점에서와 같이 앞에 오는 두 바이트는 오브젝트의 길이를 나타냅니다.

QRYDTA 오브젝트를 좀 더 자세히 살펴보면 X'241B' 코드점 다음에 X'FF'가 있는 것을 볼 수 있습니다. 이것은 널 SQLCAGRP를 의미합니다(전선을 따라 흐르는 SQLCA 또는 SQL 진단 영역 양식). SQLCAGRP의 널 양식은 관련 자료에 관한 오류나 경고 정보가 없음을 나타냅니다. 이 경우, 관련 자료는 SQL SELECT

작업으로 생긴 자료 행이며 널 SQLCAGRP 뒤에 옵니다. SQLCAGRP와 마찬가지로 자료 행이 널(null)이 될 수 있으므로 널 SQLCAGRP 다음에 오는 첫 번째 바이트는 자료 행이 널이 아님을 표시하는 X'00'을 포함하는 인디케이터입니다. 널 인디케이터 바이트의 의미는 첫 번째 비트로 결정됩니다. 이 위치에 있는 1은 널을 표시합니다. 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 아니며 그 뒤에 오는 X'0001'이 필드 자료임을 표시합니다. 그 뒤에는 널이 가능한 CHAR(3)가 오며 111이 들어 있습니다. 그 뒤에 오는 부동 소수점 값은 널이 불가능하도록 정의되므로 X'00' 바이트가 뒤에 올 수 없습니다.

첫 번째 자료 행 뒤에는 널 SQLCAGRP가 있는 두 번째 자료 행이 오고 그 뒤에는 다른 6바이트 DSS 헤더가 옵니다. 이 헤더에 들어 있는 형식 바이트의 두 번째 절반(X'2')은 해당 DSS가 REPLY임을 표시합니다. 이전 DSS의 형식 바이트(X'53')는 이전 DSS가 OBJECT DSS였음을 표시합니다. 세 번째 DSS가 전달해 준 ENDQRYRM 응답 메세지에는 REPLY DSS가 있어야 합니다. ENDQRYRM 코드점은 X'220B'입니다. 이 응답 메세지에는 X'0004'의 심각도 코드와 조회 자료('DB2ESYS')를 리턴한 RDB의 이름이 들어 있습니다.

이 예에 나오는 다음 세 번째 DSS는 네 번째이자 마지막 응답 메세지입니다. 이것의 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 알려주는 인디케이터입니다. 그 다음 4바이트 X'00000064'는 +100 SQLCODE를 표시하는데, 이는 '행을 찾을 수 없음' 조건이 발생하여 조회가 종료되었음을 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 필드로 맵핑하는 방법은 *Distributed Relational Database Architecture Reference* 서적을 참조하십시오. 이 문서는

Open Group 웹 사이트  (www.opengroup.org/dbiop/index.htm)에서 구할 수 있습니다.

RW 추적점의 설명

이 주제에서는 RW 추적점에 대해 설명합니다.

RWff RC — 수신 자료 스트림 추적점:

이 자료 스트림에는 어플리케이션 서버(AS) 프로그램의 DDM 응답이 들어 있습니다. DSS 헤더도 이 자료 스트림에 있습니다.

이는 추적점이며 261 페이지의 그림 29에서 설명됩니다.

수행되고 있는(ff) 분산 관계형 데이터베이스 구조(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 지점 조작.

RWff SN — 송신 자료 스트림 추적점:

이 자료 스트림에는 어플리케이션 리퀘스터(AR) 프로그램의 분산 자료 관리(DDM) 요구 또는 어플리케이션 서버(AS) 프로그램의 DDM 응답이 들어 있습니다. 이는 이러한 요구 또는 응답이 전선을 통한 전송 및 헤더를 추가하기 위해 하위 레벨 CN 구성요소에 제공되기 전에 존재하기 때문입니다.

내용 외에도 수신 자료 스트림 및 송신 자료 스트림에 대한 추적 정보 간의 주된 차이점은 송신 자료 스트림의 추적 정보에서 6바이트 DSS 헤더 정보가 누락되었다는 점입니다. 송신 자료 스트림 추적 영역의 첫 번째 DSS의 경우 헤더가 전체로 생략되었으며 그 다음 DSS에서는 6바이트가 0으로 되어 있어 나중에 CN 구성요소 모듈이 이를 구성할 때 헤더로 오버레이됩니다.

수행되고 있는 분산 관계형 데이터베이스 구조(DRDA) 기능의 ID는 RWff RC - 수신 자료 스트림 추적점에 대해 나열된 ID와 동일합니다.

관련 참조

263 페이지의 『RWff RC — 수신 자료 스트림 추적점』

이 자료 스트림에는 어플리케이션 서버(AS) 프로그램의 DDM 응답이 들어 있습니다. DSS 헤더도 이 자료 스트림에 있습니다.

RWQY S1 — 부분 송신 자료 스트림 추적점 1:

유사 환경에서 QRYDTA를 빌드할 때 새 조회 블록이 필요한 경우 QRWTQRY 모듈의 NEWBLOCK 루틴에 이 추적점이 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:

상이한 환경에서 QRYDTA를 빌드할 때 새 조회 블록이 필요한 경우 QRWTQRY 모듈의 NEWBLOCK 루틴에 이 추적점이 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 호출의 결과가 BPCA 구조에 대해 널이 아닌 포인터로 나타날 때(BPCA 버퍼에 하나 이상의 레코드가 리턴되었다는 것을 암시함) QRWTQRY 모듈의 FETCH 루틴에 이 추적점이 발생합니다.

덤프된 자료는 BPCA 구조(관련 버퍼 없음)로서 다른 버퍼 사이에서 리턴된 레코드의 수를 나타냅니다.

RWQY NB — 성공적이지 않은 페치 추적점:

SQFCHCRS에 대한 호출의 결과가 BPCA 구조에 대해 널 포인터로 나타나면(BPCA 버퍼에 레코드가 리턴되지 않았음을 나타냄) QRWTQRY 모듈의 FETCH 루틴에 이 추적점이 발생합니다.

덤프된 자료는 SQLSTATE입니다.

RWQY P0 — 결과 세트 의사-개방:

연관된 정보는 패키지 리스트 항목입니다.

RWQY AR—처리된 배열 결과 세트:

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.

RWQY L1 및 RWEX L1 — 아웃바운드 LOB 표 추적점에 저장됨:

이들 추적점은 나중에 어플리케이션 리퀘스터로 전송되기 위해 QRWTQRY 또는 QRWTEXEC에 의해 저장된 큰 오브젝트(LOB) 열에 관한 주소 및 기타 정보를 기록합니다.

RWQY L2 및 RWEX L2 - LOB 표 추적점으로부터 자료 스트림에 빌드됨:

이 추적점은 QRWTQRY 또는 QRWTEXEC에 의해 통신 버퍼로 복사되는 큰 오브젝트(LOB)에 대한 주소 및 기타 정보를 기록합니다.

RWQY L0 및 RWEX L0 — 인바운드 LOB 표 추적점에 저장됨:

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 — RDB 액세스 요구 추적점:

이 추적점은 DRDA 어플리케이션 리퀘스터(AR)에 있는 QRWSARDB 모듈이나 어플리케이션 서버(AS)에 있는 QRWTARDB에 대한 항목에서 발생합니다.

따라서 내용은 모듈에 따라 다양합니다. 추적이 AS에서 수행되는 경우 자료의 내용은 QRWTARDB가 실행할 DDM 명령을 식별하는 2바이트 코드점이며, 그 뒤에는 해당 명령의 영문 이름이 옵니다. 이 명령 이름은 단절의 경우 SXXDSCT이고, 클린업의 경우 SXXCLNUP이며, 연결의 경우 ACCRDB가 될 수 있습니다. 추적이 AR에서 발생하는 경우, 자료 내용은 다음과 같습니다.

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	'DISCONNE'	==>	DISCONNECT
2	'CLEANUP '	==>	CLEANUP
3	'RELEASE '	==>	RELEASE
4	'EXIT '	==>	EXIT
5	'PRECMT '	==>	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:

이 추적점은 QRWSARDB 모듈이 제공하는 기능과 연관된 제어 블록을 식별합니다.

- cb** 제어 블록의 이름.
- LV** 로컬 변수.
- DD** 확약 정의 디렉토리.
- CD** 확약 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 확약 조작 실행에서 나온 다양한 명령과 응답을 기록합니다.

앞의 'xxx'로 표시되는 자료의 세그먼트는 다음과 같을 수 있습니다.

- 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 인쇄할 수 있는 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 xx로 표시되는 두 개의 문자로 식별됩니다.

세부사항은 가변 길이 yyyyyyy 스트링에 있습니다.

- RC 유형은 수신된 명령을 기록합니다: SYNCCTL 또는 SYNCRSY.
- RW 유형은 매개변수 구조 WrwSYData를 기록합니다.
- LG 유형은 수신된 synclog를 기록합니다(여러 번 발생할 수 있음).
- 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 버퍼 및 로컬 변수를 기록합니다(기록이 두 번 발생함).
- CP 유형은 대화 프로토콜 오류가 감지되었을 때 송신 버퍼 및 로컬 변수를 기록합니다(기록이 두 번 발생함).

RW_ff_m — 어플리케이션 리퀘스터 드라이버(ARD) 제어 블록 추적점:

이 추적점은 작성될 수 있는 서로 다른 유형의 ARD 호출의 ARD 제어 블록의 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.

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

첫실패 자료 캡처(FFDC)

iSeries 서버는 분산 관계형 데이터베이스에 대한 오류 정보를 캡처하여 보고하는 방법을 제공합니다. 이 기능을 첫 번째 실패 자료 캡처(FFDC)라고 합니다.

FFDC 지원의 1차 목적은 i5/OS 오퍼레이팅 시스템의 DDM 구성요소에서 감지되는 오류에 대한 광범위한 정보를 제공하는 것이며 이 구성요소로부터 APAR(Authorized Program Analysis Report)를 작성할 수 있습니다.

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 상황을 표시할 수 있는 SQLCODE만 덤프된다는 점입니다.

첫실패 자료 캡처 덤프

시스템 감지 내부 실패는 덤프될 첫실패 자료 캡처(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 문제점이 감지되었습니다(여기서 Qxxxxxxx는 오퍼레이팅 시스템 모듈 ID임)라는 정보용 메시지가 QSYSOPR 메시지 대기행렬에 기록됩니다.

FFDC 덤프 조작의 출력을 보려면 WRKSPLF(스푼 파일에 대한 작업) 명령을 사용하여 QPSRVDMPL를 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 처음 1KB는 오류 기록부에 넣어집니다. 그러나 스푼 파일에 기록된 자료는 항상 완벽해야 하고 작업 하기 편해야 합니다. 다중 DDM 대화가 설정된 경우, 스푼 파일당 32개 항목만으로 제한되었으므로 덤프 출력은 두 개 이상의 스푼 파일에 넣어질 수 있습니다. 이 경우, QSYSOPR 메세지 대기행렬에는 별표(*) 접두 부가 표시된 여러 개의 소프트웨어 문제점 메세지가 들어가게 됩니다.

주: 해당 코드 예제를 사용하는 것은 286 페이지의 『코드 라이선스 및 면책사항 정보』의 조건에 동의한 것으로 간주합니다.

```

Work With Error Log                               02/27/91 13:33:05 Page . . . :      1
A
5738SS1 V2R1M1                                  AS/400 DUMP                                B
DUMP TAKEN FOR DETECTED ERROR                  090454/SRR/SRRS1                          02/27/91 15:12:52      PAGE 1

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       F       G
5738 MSGCPF3E86 F/QRWSQRY RC10000002

H
.SPACE-                                       01
000000   F0F17EC9 D5C4E740 F0F27EC6 C3E34E40   F0F37EC5 D4E2C740 F0F47ED7 D9D4E240   *01=INDX 02=FCT+ 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) *

L       M
.SPACE-                                       02
000000   200C1254 0102F5F8 F0F0F9
.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 *

O       P
.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 * * *

Q       R
.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 *

S       T
.SPACE-                                       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   *          QTDSQL400QSQ0201*
000060   F1002500 00000000 25000000 000010F0   F4F5F1F7 F461E2D9 D961C4E2 F3F7F840   *1          045174/SRR/DS378 *
000080   40404040 40404040 40404040 40404040   40404040 40404040 40404040 40404040   * * *

U
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 00000000 *
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 * RCHAS266          *
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 * CNSNRCVQL7971 *
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 *
DDMV02R01M*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 241BF000 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 *****

```

관련 참조

WRKSPLF(스플 파일에 대한 작업) 명령

FFDC 덤프 출력 설명

이 정보는 FFDC(First Failure Data Capture) 덤프 출력에서 사용할 수 있는 자료 영역 및 정보 유형에 대해 설명합니다.

268 페이지의 『첫실패 자료 캡처 덤프』의 그림과 같이 FFDC 덤프 출력에서 다음 자료 영역 및 정보 유형을 사용할 수 있습니다.

1. 각 FFDC 덤프 출력은 내용면에서는 서로 다르지만, 형식은 일반적으로 같습니다. 각 자료 섹션의 내용과 위치에 대한 이해를 돕기 위해 색인(I)을 제공합니다.
2. 각 자료 섹션은 SPACE와 번호로 식별됩니다. 예를 들어, SPACE- ... 01입니다. 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 섹션 이름입니다. SQL 진단 영역 자료를 찾으려면 색인(I)에서 SQCA를 찾으십시오. 덤프 색인 샘플에서 SQCA는 자료 섹션 10(10=SQCA)으로 표시됩니다. SQL 진단 영역 자료를 보려면 SPACE- 10 머리말로 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. 현재 덤프 출력에 자료 섹션이 실제로 존재하는지 여부에 관계없이 모든 자료 섹션에 대해 색인 항목이 제공됩니다. 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 들어, 샘플 덤프 출력에서 SPACE-17에서 SPACE- 21까지의 자료 섹션은 첫 번째 덤프된 자료 제어 블록(CCB 1)이고, SPACE-22에서 SPACE- 25까지의 자료 섹션은 두 번째 덤프된 자료 제어 블록(CCB 2)입니다.

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.

증상 스트링 내용:

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. 샘플 덤프 출력에서 숫자 "1"은 이 실패와 연관된 첫 번째 덤프 파일임(하나 뿐일 수 있음)을 나타냅니다.

오류 유형을 나타내는 리턴 코드의 제일 오른쪽 끝에는 4개의 숫자(0이 아님)가 있을 수 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 아래의 **I**에서 정의됩니다.

I 각 자료 섹션의 내용과 위치에 대한 이해를 돕기 위한 SPACE- 번호(**H**에 정의됨)의 색인 및 정의. 다른 자료 섹션의 순서는 다른 모듈의 덤프 출력에 따라 다를 수 있습니다. 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 않을 수 있습니다.
- 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 다음 조건에 따라 다릅니다.
 - 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 존재하지 않을 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 다음 조건에 따라 다릅니다.
 - 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 일부가 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. 각 대화 제어 블록은 다음 항목을 포함합니다.
 - 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.

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 색인에 있는 'ARDB' 스트링으로 식별됩니다. 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. LU 이름 다음에는 6바이트 LUW 인스턴스 번호 X'A7CCA7541372'가 옵니다.

관련 태스크

208 페이지의 『첫실패 자료 캡처 자료』

이 주제는 iSeries 서버에서 첫실패 자료 캡처(FFDC) 자료를 찾는 방법에 대한 추가 정보를 설명합니다. 이 정보는 어플리케이션 서버(AS)에서 FFDC 자료 출력을 일으키는 실패가 발생할 경우에 가장 유용합니다. 어플리케이션 리퀘스터(AR)에 대한 FFDC 자료는 보통 어플리케이션 프로그램을 실행하는 작업과 연관된 스펴 파일 중 하나에서 찾을 수 있습니다.

DDM 오류 코드

이들 오류 코드는 DDM 오류 조건을 식별하는 FFDC 덤프(샘플 덤프 출력의 L)에 포함됩니다. 이들 조건은 DDM 구조로 정의될 수도 또는 정의되지 않을 수도 있습니다.

명령 검사 코드:

FCT+ (SPACE- 02)에서 바이트 3과 4에 1254가 들어 있으면 바이트 6에서 다음 코드 중 하나를 찾아보십시오.

- 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 송신된 DDM 명령 또는 오브젝트가 DRDA2 구조에 대한 i5/OS 확장을 위반합니다.
- 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)에서 바이트 3과 4에 1245가 들어 있으면 바이트 6에서 다음 코드 중 하나를 찾아보십시오.

- 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
- 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 음.

FF 예상했던 SQLCARD가 빠짐.

DDM 구문 오류 코드 설명:

FCT+ (SPACE- 02)에서 바이트 3과 5에 124C가 들어 있으면 바이트 6에서 다음 코드 중 하나를 찾아보십시오.

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 오브젝트 길이가 4 미만입니다. 예를 들어, 명령 매개변수 길이가 2로 지정되고 명령 길이가 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 들어, X'8032'의 코드점이 발견되었지만 X'8'이 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'의 PRCCNVCD와 함께 PRCCNVRM을 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 DSS 연속이 2 이하. 예를 들어, DSS 연속에 대한 길이 바이트가 1 값인 경우.
- 17 필요한 순서가 아닌 오브젝트. 예를 들면, RECAL 오브젝트에 지정된 순서가 아닌 RECNR 오브젝트 앞에 오는 RECORD 오브젝트가 있는 경우.
- 18 DSS 체인화 비트가 바이너리 1이 아니지만 DSSFMT 비트 3이 바이너리 1로 설정됨. 그 다음 DSS는 동일 요구 상관자를 가짐.
- 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.








분산 데이터베이스 프로그래밍에 대한 관련 정보



다음은 분산 데이터베이스 프로그래밍 주제에 관련된 제품 매뉴얼과 IBM Redbook(PDF 형식), 웹 사이트 및 Information Center 주제입니다. 모든 PDF를 보거나 인쇄할 수 있습니다.

iSeries 서버 정보

이들 iSeries 책과 Information Center 주제에는 사용자에게 필요할 수 있는 정보가 들어 있습니다.

- ADTS/400: Data File Utility  는 어플리케이션 프로그래머, 프로그래머 또는 헬프 데스크 담당자에게 파일에 자료를 입력하고 갱신 및 조회하며 DFU 프로그램을 실행하는 프로그램을 작성하기 위한 어플리케이션 개발 툴 자료 파일 유틸리티(DFU)에 대한 정보를 제공합니다. 이 매뉴얼은 워크스테이션 오퍼레이터에게 DFU를 학습하기 위한 활동 및 자료도 제공합니다.
- 백업 및 회복 주제에서는 데이터베이스 파일에 작성된 변경사항을 기록하고 이 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.

- CL 프로그래밍 주제에서는 오브젝트와 라이브러리, 제어 언어(CL) 프로그래밍, 프로그램 간의 제어 흐름과 통신, CL 프로그램의 오브젝트에 대한 작업 및 CL 프로그램 작성을 포함하여 프로그래밍 주제의 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.
- 통신 관리  에는 통신 상태, 통신 관련 작업 관리 주제, 통신 오류, 성능, 회선 속도 및 서브시스템 기억 장치에 대한 정보가 들어 있습니다.
- iSeries용 DB2 Universal Database의 조회 관리 프로그래밍(QMF)  은 어플리케이션 프로그래머에게 보고서에 대해 조회할 데이터베이스 파일을 판별하고, SQL 조회 정의를 정의 및 조회 관리 명령을 사용하는 프로시저어를 사용하고 작성하는 데 대한 정보를 제공합니다. 이 매뉴얼은 또한 조회 글로벌 변수 지원을 사용하는 방법 및 오퍼레이팅 시스템의 조회 관리 기능과 iSeries용 IBM Query 간의 관계를 이해하는 데 대한 정보를 설명합니다.
- 분산 자료 관리 주제는 어플리케이션 프로그래머 또는 시스템 프로그래머에게 리모트 파일 처리에 대한 정보를 제공합니다. 이 주제는 i5/OS 분산 자료 관리(DD)에 대해 리모트 파일을 정의하는 방법과 DDM 파일 작성 방법, DDM을 통해 지원되는 파일 유틸리티 및 다른 시스템과 관련된 i5/OS DDM의 요구사항에 대해 설명합니다.
- DSNX Support  는 리모트 관리 지원(분산 호스트 명령 기능), 변경 관리 지원(분산 시스템 노드 실행) 및 문제점 관리 지원(경고)을 사용하기 위해 iSeries 서버를 구성하는 데 대한 정보를 제공합니다.
- Local Device Configuration  은 시스템 오퍼레이터 또는 시스템 관리자에게 초기 로컬 하드웨어 구성 방법 및 이 구성을 변경하는 방법에 대한 정보를 제공합니다. 또한 장치 구성에 관한 개념적 정보와 9406, 9404, 9402 시스템 장치의 장치 구성에 관한 계획 정보도 들어 있습니다.
- SNA Distribution Services  는 시스템 프로그래머 또는 네트워크 관리자에게 SNADS 및 VM/MVS 브릿지에 대해 통신 네트워크를 구성하는 데 대한 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. 여기에는 자료 서술 스펙(DDS) 키워드, 시스템 제어 형식, 리턴 코드, 파일 전송 지원 및 프로그래밍 예에 관한 정보도 포함됩니다.
- LAN, Frame-Relay and ATM Support  에는 토큰링 네트워크, 이더넷 네트워크 또는 브릿지 네트워크 환경에서 iSeries 서버 사용에 관한 정보가 들어 있습니다.

- Remote Work Station Support  는 표시장치 pass-through, 분산 호스트 명령 기능 및 3270 리모트 접속과 같은 리모트 워크스테이션 지원을 설정하고 사용하는 방법에 대한 정보를 제공합니다. 또한 어플리케이션 프로그래머 또는 시스템 프로그래머에게 구성 명령 및 회선, 제어기 및 장치 정의에 관한 정보를 제공합니다.
- 보안 주제는 시스템 프로그래머(또는 보안 담당자의 책임이 지정된 사용자)에게 시스템 보안 개념, 보안 계획 및 시스템에서 보안 설정에 관한 정보를 제공합니다.
- SQL 프로그래밍 주제는 어플리케이션 프로그래머, 프로그래머 또는 데이터베이스 관리자에게 SQL문을 설계하고 작성, 테스트 및 실행하는 방법의 개요를 제공합니다. 또한 대화식 SQL을 설명합니다.
- SQL 참조 주제는 어플리케이션 프로그래머, 프로그래머 또는 데이터베이스 관리자에게 SQL문 및 해당 매개변수에 관한 자세한 정보를 제공합니다.
- X.25 Network Support  에는 X.25 네트워크에서 iSeries 서버 사용에 관한 정보가 들어 있습니다.

분산 관계형 데이터베이스 라이브러리

이 주제에서는 IBM 분산 관계형 데이터베이스 구조(DRDA) 구현에 대한 백그라운드 및 일반 지원 정보를 제공합니다.

- *DRDA: Every Manager's Guide, GC26-3195*는 분산 관계형 데이터베이스 및 분산 파일에 대한 간결하고 상위 레벨의 교육 내용을 제공합니다. 이 설명서는 IBM이 분산 자료 시스템의 개발을 지원하는 방법을 설명하고 분산 자료에 대한 몇 가지 현재 IBM 제품 및 발표된 지원에 대해 논의합니다. 이 설명서의 정보는 경영자, 관리자 및 기술 직원이 분산 자료의 개념을 이해하는 데 도움을 주기 위한 것입니다.
- *DRDA: Planning for Distributed Relational Database, SC26-4650*은 분산 관계형 자료를 계획하는 데 도움을 줍니다. 이 설명서는 실행 단계, 의사 결정 및 의사 결정시 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, SC26-4783*은 분산 관계형 데이터베이스 구조를 지원하는 IBM 제품의 상호연결 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, SC26-4773*은 IBM의 관계형 데이터베이스 관리 시스템에 액세스하는 어플리케이션 프로그램을 설계하고 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, SC26-4782*는 분산 관계형 데이터베이스 환경에서의 문제점의 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

기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리

이 주제는 기타 IBM 분산 관계형 데이터베이스 플랫폼 라이브러리에 대해 설명합니다.

DB2 Connect 및 Universal Database


DB2 Connect 및 Universal Database에서 작업 중이며 자세한 정보를 원할 경우 Linux, UNIX 및 Windows용

DB2 Universal Database 웹 페이지  를 참조하십시오. 이 웹 페이지에서 다음 서적을 찾을 수 있습니다.

- *DB2 Connect Enterprise Edition Quick Beginning*
- *DB2 Connect Personal Edition Quick Beginning*
- *DB2 Connect User's Guide*
- *DB2 UDB Administration Guide*
- *DB2 UDB Command Reference*
- *DB2 UDB for OS/2 Quick Beginnings*
- *DB2 UDB for UNIX Quick Beginnings*
- *DB2 UDB for Windows NT[®] Quick Beginnings*
- *DB2 UDB Messages Reference*
- *DB2 UDB Personal Edition Quick Beginnings*
- *DB2 UDB SQL Getting Started*
- *DB2 UDB SQL Reference*
- *DB2 UDB Troubleshooting Guide*

z/OS 및 OS/390[®]용 DB2

z/OS 및 OS/390용 DB2에서 작업 중이며 자세한 정보를 원할 경우 z/OS 및 OS/390용 DB2 웹 페이지

 를 참조하십시오. 이 웹 페이지에서 다음 서적을 찾을 수 있습니다.

- *DB2 for z/OS and OS/390 Command Reference*
- *DB2 for z/OS and OS/390 Messages and Codes*
- *DB2 for z/OS and OS/390 Reference for Remote DRDA*
- *DB2 for z/OS and OS/390 SQL Reference*
- *DB2 for z/OS and OS/390 Utility Guide and Reference*

VSE & VM용 DB2 Server

VSE & VM용 DB2 Server에서 작업 중이며 자세한 정보를 원할 경우 VSE & VM용 DB2 Server 웹 페이지

를 참조하십시오. 이 웹 페이지에서 다음 서적을 찾을 수 있습니다.

- *DB2 and Data Tools for VSE and VM*
- *DB2 for VM Control Center Installation*
- *DB2 Server Data Spaces Support for VM/ESA[®]*
- *DB2 Server for VM Application Programming*
- *DB2 Server for VM Database Administration*
- *DB2 Server for VM Database Services Utilities*
- *DB2 Server for VM Diagnosis Guide*
- *DB2 Server for VM Interactive SQL Guide*
- *DB2 Server for VM Master Index and Glossary*
- *DB2 Server for VM Messages and Codes*
- *DB2 Server for VM Operation*
- *DB2 Server for VM System Administration*
- *DB2 Server for VM/VSE Training Brochure*
- *DB2 Server for VSE & VM Quick Reference*
- *DB2 Server for VSE & VM SQL Reference*
- *DB2 Server for VSE & VM LPS*
- *DB2 Server for VSE & VM Data Restore*
- *SBOF for DB2 Server for VM*

구조 서적

이 주제에서는 서로 다른 구조 서적에 대해 설명합니다.

- 문자 자료 표시 구조: 세부사항, 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 컬렉션과 IBM에서 사용되는 문자 명명 정보가 들어 있는 CD-ROM 이 있습니다. CD는 또한 제공된 자료로 사용될 보기 유틸리티를 포함합니다. 표시기는 OS/2, Windows 3.1 및 Windows 95에서 작동합니다.

- 문자 자료 표시 구조: 개요, GC09-2207
- *DRDA V3 Vol. 1: Distributed Relational Database Architecture*

이러한 기술 표준은 분산 관계형 데이터베이스 구조 스펙을 문서화한 세 권 중 한 권입니다. 이 책은 어플리케이션 프로그램이 분산 관계형 자료에 액세스할 수 있도록 하는 관계형 데이터베이스 관리자 간의 연결성에 대해 설명합니다. 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. 이 문서는 Open Group 웹 사이트(www.opengroup.org/dbiop/index.htm)에서 구할 수 있습니다.

- *DRDA V3 Vol. 2: Formatted Data Object Content Architecture*


이 문서는 Distributed Relational Database Architecture, 버전 3을 문서화하는 세 권의 기술 표준 중 한 권입니다. 이 책은 형식화된 자료 오브젝트 내용 구조(FD:OCA: Formatted Data Object Content Architecture)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. 이 문서는 www.opengroup.org/dbiop/index.htm에 있는 Open Group 웹 사이트에서 구할 수 있습니다.

- *DRDA V3 Vol. 3: Distributed Data Management Architecture*

이 문서는 (DRDA(Distributed Relational Database Architecture)를 문서화하는 세 권의 기술 표준 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. 이 문서는 Open Group 웹 사이트(www.opengroup.org/dbiop/index.htm)에서 구할 수 있습니다.

IBM Redbook

이 주제는 분산 관계형 데이터베이스에 사용할 수 있는 IBM Redbook에 대해 설명합니다.

- Distributed Relational Database: Using DDCS/6000 DRDA Support with DB2 and DB2/400, GG24-4155-00 

- DRDA Client/Server for VM and VSE Setup for System and Performance Management, GG24-4275-00



- Getting Started with DB2 Stored Procedures: Give Them a Call through the Network, SG24-4693-01



- WOW! DRDA Supports TCP/IP: DB2 Server for OS/390 and DB2 Universal Database, SG24-2212-00



코드 라이선스 및 면책사항 정보

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

- | 강행 법규에 규정된 보증 조항의 적용을 제외하고, IBM은 해당 프로그램 또는 기술 지원에 대한 상품성, 특정 목적에의 적합성 및 타인의 권리 침해에 대한 묵시적 보증을 포함한(단, 이에 한하지 않음) 일체의 묵시적 또는 명시적인 보증이나 주장도 제공하지 않습니다.
- | IBM, IBM 프로그램 개발자 또는 공급자는, 손해 발생의 가능성을 통지 받은 경우를 포함한 어떠한 경우에도 다음에 대하여 책임 지지 않습니다.
 - | 1. 데이터의 손실 또는 손상
 - | 2. 직접적인, 특별한, 우연에 의한 또는 간접적인 손상 또는 이에 따른 경제적 손실 또는
 - | 3. 기대했던 이익, 사업, 수익, 영업권 또는 비용 절감이 실현되지 못함으로 인해 발생하는 손해
- | 일부 관할권에서는 부수적 또는 결과적 손해의 제외사항이나 제한사항을 허용하지 않으므로, 상기 제외사항이나 제한사항이 귀하에게 적용되지 않을 수도 있습니다.

부록. 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 언급하는 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 운용에 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, 변경된 사항은 최신판에 통합됩니다. IBM은 이 책에서 설명한 제품 및/또는 프로그램을 사전 통지없이 언제든지 개선 및/또는 변경할 수 있습니다.

이 정보에서 언급되는 비IBM의 웹 사이트는 단지 편의상 제공된 것으로, 어떤 방식으로든 이들 웹 사이트를 옹호하고자 하는 것은 아닙니다. 해당 웹 사이트의 자료는 본 IBM 제품 자료의 일부가 아니므로 해당 웹 사이트 사용으로 인한 위험은 사용자 본인이 감수해야 합니다.

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

(i) 독립적으로 작성된 프로그램과 기타 프로그램(본 프로그램 포함) 간의 정보 교환 및 (ii) 교환된 정보의 상호 이용을 목적으로 본 프로그램에 관한 정보를 얻고자 하는 라이선스 사용자는 다음 주소로 문의하십시오.

135-270

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 사용 가능한 모든 라이선스가 있는 자료
- | 는 IBM이 IBM 기본 계약, IBM 프로그램 라이선스 계약(IPLA), 기계 코드에 대한 IBM 라이선스 계약 또
- | 는 이와 동등한 계약에 따라 제공한 것입니다.

본 문서에 포함된 모든 성능 데이터는 제한된 환경에서 산출된 것입니다. 따라서 다른 운영 환경에서 얻어진 결과는 상당히 다를 수 있습니다. 일부 성능은 개발 레벨 상태의 시스템에서 측정되었을 수 있으므로 이러한 측정치가 일반적으로 사용되고 있는 시스템에서도 동일하게 나타날 것이라고는 보증할 수 없습니다. 또한, 일부 성능은 추정을 통해 추측되었을 수도 있으므로 실제 결과는 다를 수 있습니다. 이 책의 사용자는 해당 데이터를 사용자의 특정 환경에서 검증해야 합니다.

비IBM 제품에 관한 정보는 해당 제품의 공급업체, 공개 자료 또는 다른 기타 범용 소스로부터 얻은 것입니다. IBM에서는 이러한 비IBM 제품을 테스트하지 않았으므로, 이들 제품과 관련된 성능의 정확성, 호환성 또는 기타 주장에 대해서는 확신할 수 없습니다. 비IBM 제품의 성능에 대한 의문사항은 해당 제품의 공급업체에 문의하십시오.

IBM이 제시하는 방향 또는 의도에 관한 모든 언급은 특별한 통지없이 변경될 수 있습니다.

표시된 모든 IBM 가격은 IBM이 제시하는 소매 가격이고 현재가이며 통지없이 변경될 수 있습니다. 실제 판매가는 다를 수 있습니다.

이 정보는 계획 수립 목적으로만 사용됩니다. 이 정보는 기술된 제품이 GA(General Availability)되기 전에 변경될 수 있습니다.

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.

저작권 라이선스:

이 정보에는 여러 가지 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 이러한 샘플 프로그램을 추가 비용없

이 어떤 형태로든 복사, 수정 및 배포할 수 있습니다. 이러한 예제는 모든 조건하에서 철저히 테스트된 것은 아닙니다. 따라서 IBM은 이러한 프로그램의 신뢰성, 서비스 가능성 또는 기능을 보증하거나 암시하지 않습니다.

이러한 샘플 프로그램 또는 파생 제품의 각 사본이나 그 일부에는 반드시 다음과 같은 저작권 표시가 포함되어야 합니다.

© (귀하의 회사명) (연도). 이 코드의 일부는 IBM Corp.의 샘플 프로그램에서 파생됩니다. © Copyright IBM Corp. _연도_. All rights reserved.

이 정보를 소프트웨어로 확인하는 경우에는 사진과 컬러 삽화가 제대로 나타나지 않을 수도 있습니다.

프로그래밍 인터페이스 정보

이 분산 데이터베이스 프로그래밍 서적은 고객이 프로그램을 작성하여 IBM i5/OS의 서비스를 제공받을 수 있도록 하는 계획된 프로그래밍 인터페이스를 문서화합니다.

상표

다음 용어는 미국 또는 기타 국가에서 사용되는 IBM Corporation의 상표입니다.

- |
- | Advanced Peer-to-Peer Networking
- | AIX
- | C/400
- | CICS
- | COBOL/400
- | DataJoiner
- | DataPropagator
- | DB2
- | DB2 Connect
- | DB2 Universal Database
- | Distributed Relational Database Architecture
- | DRDA
- | DXT
- | e(로고)server
- | Extended Services
- | i5/OS
- | IBM
- | IBM(로고)
- | IMS
- | Informix

- | Integrated Language Environment
- | iSeries
- | MVS
- | NetView
- | OS/2
- | OS/390
- | OS/400
- | PS/2
- | Redbooks
- | RPG/400
- | RS/6000
- | S/390
- | PS/2
- | Redbooks
- | RPG/400
- | RS/6000
- | S/390
- | SQL/DS
- | VM/ESA
- | z/OS

- | Intel, Intel Inside(로고), MMX 및 Pentium은 미국 또는 기타 국가에서 사용되는 Intel Corporation의 상표
| 입니다.

Microsoft, Windows, Windows NT 및 Windows 로고는 미국 또는 기타 국가에서 사용되는 Microsoft Corporation의 상표입니다.

Java 및 모든 Java 기반 상표는 미국 또는 기타 국가에서 사용되는 Sun Microsystems, Inc.의 상표입니다.

- | Linux는 미국 또는 기타 국가에서 사용되는 Linus Torvalds의 상표입니다.

UNIX는 미국 또는 기타 국가에서 The Open Group의 등록상표입니다.

기타 회사, 제품, 서비스 이름은 해당 회사의 상표 또는 서비스표입니다.

조건

다음 조건에 따라 본 발행물을 사용할 수 있습니다.

개인적 사용: 귀하는 모든 소유권 사항을 표시하는 것을 조건으로 본 발행물을 개인적, 비상업적 용도로 복제할 수 있습니다. 귀하는 IBM의 명시적 동의없이 본 발행물 또는 그 일부를 배포 또는 게시하거나 이에 대한 2차적 저작물을 만들 수 없습니다.

상업적 사용: 귀하는 모든 소유권 사항을 표시하는 것을 조건으로 본 발행물을 귀하 사업장 내에서만 복제, 배포 및 게시할 수 있습니다. 귀하의 사업장 외에서는 IBM의 명시적 동의없이 본 발행물의 2차적 저작물을 만들거나 본 발행물 또는 그 일부를 복제, 배포 또는 게시할 수 없습니다.

본 허가에서 명시적으로 부여된 경우를 제외하고, 본 발행물이나 본 발행물에 포함된 정보, 데이터, 소프트웨어 또는 기타 지적 재산권에 대해서는 어떠한 허가나 라이선스 또는 권리도 명시적 또는 묵시적으로 부여되지 않습니다.

IBM은 본 발행물의 사용이 IBM의 이익을 해친다고 판단하거나 위에서 언급된 지시사항이 준수되지 않는다고 판단하는 경우 언제든지 부여한 허가를 철회할 수 있습니다.

귀하는 미국 수출법 및 관련 규정을 포함하여 모든 적용 가능한 법률 및 규정을 철저히 준수하는 것을 조건으로 본 정보를 다운로드, 송신 또는 재송신할 수 있습니다.

IBM은 본 발행물의 내용에 대해 어떠한 보증도 하지 않습니다. IBM은 상품성 및 특정 목적에의 적합성에 대한 보증을 포함하여 명시적이든 묵시적이든 일체의 보증없이 "현상태대로" 본 발행물을 제공합니다.

IBM